A Direct Algorithm for 1D Total Variation Denoising
Laurent Condat

To cite this version:

Laurent Condat. A Direct Algorithm for 1D Total Variation Denoising. IEEE Signal Processing Letters, Institute of Electrical and Electronics Engineers, 2013, 20 (11), pp.1054-1057. <10.1109/LSP.2013.2278339>. <hal-00675043v4>

HAL Id: hal-00675043
https://hal.archives-ouvertes.fr/hal-00675043v4
Submitted on 11 Aug 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
A Direct Algorithm for 1D Total Variation Denoising

Laurent Condat

Abstract—A very fast noniterative algorithm is proposed for denoising or smoothing one-dimensional discrete signals, by solving the total variation regularized least-squares problem or the related fused lasso problem. A C code implementation is available on the web page of the author.

Index Terms—Total variation, denoising, nonlinear smoothing, fused lasso, regularized least-squares, nonparametric regression, convex nonsmooth optimization, taut string

I. INTRODUCTION

The problem of smoothing a signal, to remove or at least attenuate the noise it contains, has numerous applications in communications, control, machine learning, and many other fields of engineering and science [1]. In this paper, we focus on the numerical implementation of total variation (TV) denoising for one-dimensional (1D) discrete signals; that is, we are given a (noisy) signal $y = (y[1],\ldots,y[N]) \in \mathbb{R}^N$ of size $N \geq 1$, and we want to efficiently compute the denoised signal $x^* \in \mathbb{R}^N$, defined implicitly as the solution to the minimization problem

$$\minimize_{x \in \mathbb{R}^N} \frac{1}{2} \sum_{k=1}^{N} |y[k] - x[k]|^2 + \lambda \sum_{k=1}^{N-1} |x[k+1] - x[k]|, \quad (1)$$

for some regularization parameter $\lambda \geq 0$ (whose choice is a difficult problem by itself [2]). We recall that, as the functional to minimize is strongly convex, the solution $x^*$ to the problem exists and is unique, whatever the data $y$. The TV denoising problem has received large attention in the communities of signal and image processing, inverse problems, sparse sampling, statistical regression analysis, optimization theory, among others. It is not the purpose of this paper to review the properties of the nonlinear TV denoising filter, since numerous papers can be found on this vast topic; see, e.g., [3]–[6] for various insights.

A more general problem, which encompasses TV denoising as a particular case, is the fused lasso signal approximator, introduced in [7], which yields a solution that has sparsity in both the coefficients and their successive differences. It consists in solving the problem

$$\minimize_{z \in \mathbb{R}^N} \frac{1}{2} \sum_{k=1}^{N} |z[k] - y[k]|^2 + \lambda \sum_{k=1}^{N-1} |z[k+1] - z[k]| + \mu \sum_{k=1}^{N} |z[k]|, \quad (2)$$

for some $\lambda \geq 0$ and $\mu \geq 0$. The fused lasso has many applications, e.g. in bioinformatics [8]–[10]. As shown in [9], the complexity of the fused lasso is the same as TV denoising, since the solution $z^*$ can be obtained by simple soft-thresholding from the solution $x^*$ of (1):

$$z^*[k] = \begin{cases} x^*[k] - \mu \text{sign}(x^*[k]) & \text{if } |x^*[k]| > \mu \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

It is straightforward to add soft-thresholding steps to the proposed algorithm, for essentially the same computation time. So, for simplicity of the exposition, we focus on the TV denoising problem (1) in the sequel.

To solve the convex nonsmooth optimization problem (1), we mostly find in the literature iterative fixed-point methods [11]. Until not so long ago, such methods applied to TV regularization had rather high computational complexity [12]–[16], but the growing interest for related $\ell_1$-norm problems in compressed sensing or sparse recovery [17], [18] has yielded advances in the field. Recent iterative methods based on operator splitting, which exploit both the primal and dual formulations of the problems and use variable stepsize strategies or Nesterov-style accelerations, are quite efficient when applied to TV-based problems [19]–[21]. Graph cuts methods can also be used to solve (1) or its extension on graphs [22]; they actually solve a quantized version of (1); the minimizer $x^*$ is not searched in $\mathbb{R}^N$ but in $\mathbb{E}Z^N$, for some $\mathbb{E} > 0$, with complexity $O(\log_2(1/\mathbb{E})N)$. If $\mathbb{E}$ is small enough, the exact solution in $\mathbb{R}^N$ can be obtained from the quantized one, as shown by Hochbaum [23], [24]. In this paper, we present a novel and very fast algorithm to compute the denoised signal $x^*$ solution to (1), exactly, in a
1D TV Denoising Algorithm

Input: integer size $N \geq 1$, real sequence $(y[1], \ldots, y[N])$, real parameter $\lambda > 0$. Output: real sequence $(x^*[1], \ldots, x^*[N])$ solution to (1).

1. Set $k = k_0 = k_\ast \leftarrow 1$, $v_{\min} \leftarrow y[1] - \lambda$, $v_{\max} \leftarrow y[1] + \lambda$, $u_{\min} \leftarrow -\lambda$, $u_{\max} \leftarrow -\lambda$.
2. If $k = N$, set $x^*[N] \leftarrow v_{\min} + u_{\min}$ and terminate.
3. If $y[k+1] + u_{\min} < v_{\min} - \lambda$, set $x^*[k] = \cdots = x^*[k_-] \leftarrow v_{\min}$, $k = k_0 = k_- \leftarrow k_- + 1$, $v_{\min} \leftarrow y[k]$, $v_{\max} \leftarrow y[k] + 2\lambda$, $u_{\min} \leftarrow -\lambda$, $u_{\max} \leftarrow -\lambda$.
4. Else, if $y[k+1] + u_{\max} > v_{\max} + \lambda$, set $x^*[k] = \cdots = x^*[k_+ - 1] \leftarrow v_{\min}$, $k = k_0 = k_+ \leftarrow k_+ + 1$, $v_{\min} \leftarrow y[k] - 2\lambda$, $v_{\max} \leftarrow y[k]$, $u_{\min} \leftarrow -\lambda$, $u_{\max} \leftarrow -\lambda$.
5. Else, set $k \leftarrow k + 1$, $v_{\min} \leftarrow u_{\min} + y[k] - v_{\min}$ and $u_{\max} \leftarrow u_{\max} + y[k] - v_{\max}$.
6. If $u_{\max} \leq -\lambda$, set $u_{\max} \leftarrow u_{\max} + (u_{\max} - \lambda)/(k - k_0 + 1)$, $u_{\min} \leftarrow -\lambda$, $k \leftarrow k - 1$.
7. If $k < N$, go to 3.
8. If $v_{\min} < 0$, set $x^*[k_0] = \cdots = x^*[k_-] \leftarrow v_{\min}$, $k = k_0 = k_- \leftarrow k_- + 1$, $v_{\min} \leftarrow y[k]$, $u_{\min} \leftarrow -\lambda$, $u_{\max} \leftarrow y[k] + \lambda - v_{\max}$.
9. Then go to 2.
10. Else, set $x^*[k_0] = \cdots = x^*[N] \leftarrow v_{\min} + u_{\min}/(k - k_0 + 1)$ and terminate.

Then, the problems (1) and (4) are equivalent, in the sense that their respective solutions $x^*$ and $s^*$ are related by $x^*[k] = s^*[k] - s^*[k-1]$, for $1 \leq k \leq N$ [26]. Thus, the formulation (4) allows to express the TV solution $x^*$ as the discrete derivative of a string threaded through a tube around the discrete primitive of the data, and pulled taut such that its length is minimized. This principle is illustrated in Fig. 1. The taut string algorithm [26] is directly based on this formulation; it consists in alternating between the computation of the greatest convex minorant and least concave majorant of the upper and lower strings $r + \lambda$ and $r - \lambda$. By contrast, the proposed algorithm does not manipulate any running sum, does not require any auxiliary memory buffer, and only performs forward scans. We describe it and discuss its performances in the next section.

II. PROPOSED METHOD

We first introduce the (Fenchel-Moreau-Rockafellar) dual problem to the primal problem (1) [11]:

\[
\min_{u \in \mathbb{R}^{N+1}} \sum_{k=1}^{N} \|y[k] - u[k] + u[k-1]\|^2 \quad \text{s.t.} \quad |u[k]| \leq \lambda, \quad \forall k = 1, \ldots, N - 1, \quad \text{and} \quad u[0] = u[N] = 0. \tag{5}
\]

Once the solution $u^*$ to the dual problem is found, one recovers the primal solution $x^*$ by

\[
x^*[k] = y[k] - u^*[k] + u^*[k-1], \quad \forall k = 1, \ldots, N. \tag{6}
\]


\[
u^*[0] = 0, \quad u^*[N] = 0 \quad \text{and} \quad \forall k = 1, \ldots, N - 1, \quad \begin{cases} u^*[k] \in [-\lambda, \lambda] & \text{if } x^*[k] = x^*[k+1], \\ u^*[k] = -\lambda & \text{if } x^*[k] < x^*[k+1], \end{cases} \tag{7}
\]

Hence, the proposed algorithm consists in running forwardly through the samples $y[k]$; at location $k$, it tries to prolongate the current segment of $x^*$ by $x^*[k+1] = x^*[k]$. If this is not possible without violating (6) and (7), it goes back to the last location where a jump can be introduced in $x^*$, validates the current segment until this location, starts a new segment, and continues. In more details, the proposed algorithm, given at the top of the page, works as follows. The variables are initialized at Step 1. At Step 2., we are at some location $k$ and we are building a segment starting at $k_0$, with value $v = x^*[k_0] = \cdots = x^*[k]$. $v$ is unknown but we know the values $v_{\min}$ and $v_{\max}$ such that $v \in [v_{\min}, v_{\max}]$. The auxiliary values $v_{\min}$ and $v_{\max}$ are the values of $u^*[k]$ in the hypothetic cases $v = v_{\min}$ and $v = v_{\max}$, respectively. Now, we are trying to prolongate the
The worst case complexity of the algorithm is $O(N + (N - 1) + \cdots + 1) = O(N^2)$. Indeed, every added segment has size at least one, but the algorithm may have to scan all the remaining samples to validate it in one of the steps 3., 4., 8., 9. However, this worst case scenario is encountered only when $x^*$ is a ramp with very small slope of order $N^{-2}$, except at the boundaries; for instance, consider that $\lambda = 1$ and $y[1] = -2$, $y[k] = \alpha(k - 2)$ for $2 \leq k \leq N - 1$, $y[N] = \alpha(N - 3) + 2$, where $\alpha = 4/(N - 2)$. The solution $x^*$ is such that $x^*[1] = y[1] + 1$, $x^*[k] = y[k]$ for $2 \leq k \leq N - 1$, $x^*[N] = y[N] - 1$. Actually, such a pathological case, for which there is no interest in applying TV denoising, is only a curiosity; the complexity is $O(N)$ in all practical situations, because the segments of $x^*$ are validated with a delay which does not depend on $N$.

The algorithm was implemented in C and run on an Apple laptop with a 2.3 GHz Intel Core i7 processor. The computation time was around 25ms with $N = 10^6$, for various test signals and noise levels. Importantly, the computation time is insensitive to the value of $\lambda$. The taut string algorithm was implemented in C as well, by adapting Matlab code written by Lutz Dümbgen and available online. In the same conditions, the computation time was around 55ms.

Thus, the taut string algorithm is efficient, but the proposed algorithm outperforms it by a constant factor consistently. We also implemented the popular iterative method FISTA [19] to solve the dual problem (5). The C code was quite optimized, with only one loop of size $N$ per iteration. As a result, on the same machine, the computation time was around $10^{-6}N$ seconds per iteration. Thus, we can consider that the proposed algorithm takes roughly the same time as three iterations of FISTA. We should keep in mind that an iterative method like FISTA may need several thousands of iterations to converge within reasonable precision, especially for large values of $N$ and $\lambda$.

For illustration purpose, we consider the example of a noisy Lévy process [6]: $y[k] = x_0[k] + e[k]$ for $1 \leq k \leq N = 1000$, where $e \sim \mathcal{N}(0, I_N)$ and the ground truth $x_0$ has a fixed value $x_0[1]$ and i.i.d. random increments $d[k] = x_0[k] - x_0[k - 1]$ for $2 \leq k \leq N$. We chose a sparse Bernoulli-Gaussian law for the increments, since TV denoising is close to optimality for such signals [5], [6]; that is, the probability density function of $d[k]$ is $p(\delta(t) + \frac{1 - p}{\sigma \sqrt{2\pi}} \exp(-\frac{t^2}{2\sigma^2}))$, $\forall t \in \mathbb{R}$, where $p = 0.95$, $\sigma = 4$ and $\delta(t)$ is the Dirac distribution. We found empirically that the mean squared error $\|x_0 - x^*\|^2/N$ is minimized for $\lambda = 2$. The computation time of $x^*$, averaged over several runs and realizations, was 25 microseconds. One realization of the experiment is depicted in Fig. 3. For this example, FISTA needs 10,000 iterations to converge within machine precision.

III. Conclusion

In this article, we proposed a direct and very fast algorithm for denoising 1D signals by total variation (TV) minimization or fused lasso approximation. Since the algorithm computes the proximity operator [11] of the 1D TV seminorm, it can be used as a basic unit within iterative splitting methods, to solve inverse problems in signal processing and imaging. This approach will be developed in a forthcoming paper.

It would be worth investigating the possibility of extending the algorithm to complex-valued or multi-valued signals [10] and to data of higher dimensions, like 2D images or graphs [29]. Besides, path-following, a.k.a. homotopy, algorithms have been proposed for $\ell_1$-penalized problems; they can find the smallest value of $\lambda$ and the associated $x^*$ in (1), such that $x^*$ has at most $m$ segments, with complexity $O(mN)$ [9], [18], [27], [35]–[37]. Their relationship to the approach in [38] and to the proposed algorithm should be studied. This is left for future work.

References


Fig. 2. Taut string interpretation of the progression of the proposed algorithm, for the construction of the segment from $x^*_{[11]}$ to $x^*_{[14]}$ in the example of Fig. 1. The values $v_{\min}$ and $v_{\max}$ are the respective slopes of the affine minorant and majorant (in blue) of the sought string segment (in red in (h)). Each subfigure from (a) to (h) shows the progression during one pass (Steps 3 to 6) of the algorithm. The index $k_*$ keeps track of the last position where the majorant touches the upper string. Note that $v_{\max}$ and $k_*$ are updated (Step 6) in (a)–(d) but not after. The jump is detected in (h) because an update of $v_{\min}$ would violate $v_{\min} \leq v_{\max}$. Consequently, the segment ends at $k_*=14$.
Fig. 3. In this example, $y$ (in red) is a piecewise constant process of size $N = 1000$ (unknown ground truth, in green) corrupted by additive Gaussian noise of unit variance. The TV-denoised signal $x^*$, with $\lambda = 2$, is in blue.