STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. 9, June 2021, pp 435–452.
Published online in International Academic Press (www.IAPress.org)

Nonconvex Energy Minimization with Unsupervised Line Process Classifier for Efficient Piecewise Constant Signals Reconstruction

Anass Belcaid [1], Mohammed Douimi [2,*]

[1] Department of Mathematics, School of Artificial Intelligence, Euromed-Fes
[2] Department of Mathematics, National School of Arts and Craft-Meknes

Abstract
In this paper, we focus on the problem of signal smoothing and step-detection for piecewise constant signals. This problem is central to several applications such as human activity analysis, speech or image analysis and anomaly detection in genetics. We present a two-stage approach to approximate the well-known line process energy which arises from the probabilistic representation of the signal and its segmentation. In the first stage, we minimize a total variation (TV) least square problem to detect the majority of the continuous edges. In the second stage, we apply a combinatorial algorithm to filter all false jumps introduced by the TV solution. The performances of the proposed method were tested on several synthetic examples. In comparison to recent step-preserving denoising algorithms, the acceleration presents a superior speed and competitive step-detection quality.

Keywords signal denoisng, edge preserving, DPS algorithm, total variation, energy minimization, unsupervised segmentation.

AMS 2010 subject classifications 94A08, 68U10

DOI:10.19139/soic-2310-5070-994

1. Introduction

The problem of removing noise from piecewise constant (PWC) signals occurs naturally in several fields of applied sciences like genomic [1, 2], nanotechnology [3], image analysis [4] and finance [5, 6]. The main task is to recover the true signal from a noisy measurement without losing the important information of the abrupt jumps. Being a PWC implies that the denoising task could be interpreted in two different ways. First, the values of each plateau constitute a reconstruction of the true signal. Second, the position of jumps gives a direct segmentation of the true signal. Therefore, the solution is a joint restoration and segmentation, which leads to better performance compared to executing each task separately [7].

We focus on the numerical implementation of recovering one-dimensional discrete PWC signal from noisy measurements. The observed signal is defined as $y = u + \varepsilon$, where $y = (y_0, \dots, y_n)^T \in \mathbb{R}^{n+1}$, $u = (u_0, \dots, u_n) \in \mathbb{R}^{n+1}$ is a PWC signal, and $\varepsilon = (\varepsilon_0, \dots, \varepsilon_n) \in \mathbb{R}^{n+1}$ is i.i.d. Gaussian noise with variance σ^2. The goal is to restore efficiently the optimal PWC signal from the observed measurement y.

The relation between u and y alone is not sufficient to compute a solution [8]. It is mandatory to take advantage of additional prior knowledge about the initial signal in order to fix a set of acceptable

[*]Correspondence to: Anass Belcaid (Email: a.belcaid@ueuromed.org). Department of Mathematics, Euromed Fes.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

solutions. The prior knowledge is incorporated either by the means of regularization [9, 10] or by the Markov Random Fields (MRF) [11, 12] framework. Both of which lead to the minimization of an energy function in the form:

$$E(u) = \|u - y\|_2^2 + \lambda\phi(u). \tag{1}$$

The first term in the energy E represents the fidelity term, where the choice of the L_2 norm corresponds to the assumption that ε is a white Gaussian noise. The second term encodes the prior knowledge and forces the solution u to exhibit a set of given features. Finally, the parameter $\lambda > 0$ is set to control the trade-off between the fidelity term and the prior knowledge [13]. The choice of λ is a difficult problem in itself [14]. Therefore, the energy in (1) needs to be minimized for different values of λ in order to find the best restoration.

In the MRF framework, the prior is generally expressed as a sum of Interaction Penalties (IP) over pairwise cliques (higher order cliques are also useful [15]). For PWC signals, the IP specify a smoothness assumption on the homogeneous plateaus, while this constraint must be switched off for the abrupt jumps so as to avoid over smoothing. Since the seminal work of Geman and Geman [16], several non-convex IP have been considered [17, 13, 18] for PWC denoising. We emphasize on the truncated quadratic interaction [12] $V(u_p, u_q) = \min\{\alpha, (u_p - u_q)^2\}$ which is equivalent to the Line Process (LP) introduced in [16]. With this choice, the regularizer tem $\phi(u)$ is given as sum ov IP $V(u_i, u_{i+1})$ over binary cliques in the form of $(i, i+1)$:

$$\phi(u) = \sum_{i=0}^{n-1} V(u_{i+1}, u_i) = \sum_{i=0}^{n-1} \min\left\{\alpha, \lambda(u_{i+1} - u_i)^2\right\}. \tag{2}$$

The LP model adds a boolean variable $l \in \{0,1\}^n$ that explicitly indicates the existence $(l_i = 1)$ or absence $(l_i = 0)$ of a discontinuity. The state of variable l_i is defined by a threshold h called the sensitivity as follows:

$$l_i = \begin{cases} l_i = 0 & \text{if} \qquad\qquad |u_{i+1} - u_i| > h \\ l_i = 1 & \text{otherwise.} \end{cases} \tag{3}$$

The energy is then given by:

$$E(u, l) = \|u - y\|_2^2 + \lambda \sum_{i=0}^{n-1} \left((u_{i+1} - u_i)^2(1 - l_i) + \alpha l_i\right), \tag{4}$$

where $\alpha = \dfrac{\lambda h^2}{2}$ is the penalty for introducing a discontinuity [19].

The minimization of the energy in (4) is NP-hard [20], as it is non-convex, non-smooth at the minimizer and involves mixed continuous and binary variables. Several algorithms have been proposed to compute an approximation of the solution. We mention some convex relaxation techniques [21, 22], Simulated Annealing approach [23], sequential tree-reweighted message passing [24] and graph cut based algorithms [12, 25, 26]. We refer the reader to [27, 28] for a thorough investigation of the literature.

The main contributions of this paper are as follows:

- Minimize a total variation (TV) regularized least square problem with the L_1 penalty in order to find an approximation to (4).
- Make use of the fast Condat denoiser [29] to compute a solution $\hat{u}$ to the TV problem in a linear time $\mathcal{O}(n)$.

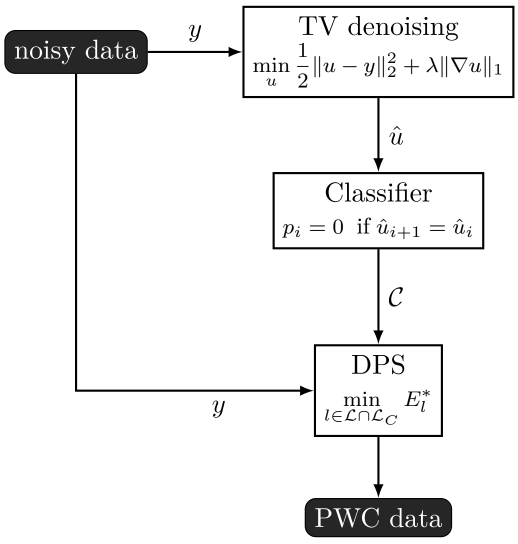
noisy data
y
TV denoising
$\min_u \frac{1}{2}\|u-y\|_2^2 + \lambda\|\nabla u\|_1$
$\hat{u}$
Classifier
$p_i = 0$ if $\hat{u}_{i+1} = \hat{u}_i$
$\mathcal{C}$
DPS
$\min_{l\in\mathcal{L}\cap\mathcal{L}_C} E_l^*$
y
PWC data

Figure 1. Flowchart of the proposed algorithm.

- Incorporate the finding in $\hat{u}$, to propose an efficient pruning scheme for Discontinuity Position Sweep (DPS) combinatorial search.

The main scheme of our method is depicted in (Fig. 1). We demonstrate the effectiveness of this scheme by a series of simulation on synthetic signals proposed in [30, 31, 32]. The results show an efficient increase in time and gain in robustness in the case of extremely corrupted signals.

2. Optimization algorithm

2.1. Total variation denoising for line process classification

TV denoising is widely used in noisy signal processing [33]. The solution is implicitly defined as the global minimum of a convex energy function similar to (4) involving a data fidelity term and a regularizer. For PWC signals, the most used choice of the regularizer is the gradient L_1 norm and the resulting convex optimization problem is:

$$\min_{u\in\mathbb{R}^{n+1}} \frac{1}{2}\sum_{i=0}^{n}(u_i - y_i)^2 + \lambda\sum_{i=0}^{n-1}|u_{i+1}-u_i|. \tag{5}$$

In the MRF framework, the TV energy functional in (5) corresponds to an IP function given by $V(u_p,u_q) = |u_p - u_q|$. Compared to the truncated quadratic function $\min\{\alpha,(u_p-u_q)^2\}$, this interaction potential is simpler as it is strongly convex and leads to a convex energy functional. This energy function admits a unique minimizer u^*, whatever the data y.

For the choice of regularization parameter λ, it turned out that is a difficult problem by itself [3]. Several works have been proposed to study the properties and characteristics of this nonlinear TV denoiser. We refer the reader to [34, 35, 36] for various insights on the topic.

The energy in (5) is generally tackled by fixed point methods [37] with optimal theoretical complexity. Contrastingly, we choose the fast non-iterative TV denoising algorithm [29] with a higher theoretical complexity compared to fixed point methods. Experimentally, it achieves a competitive or even faster linear complexity and can handle large bandwidth signals in a few milliseconds.

The proposed algorithm solves the (Frenchel-Moreau-Rockafellar) dual problem [38] to (5):

$$\begin{array}{ll} \min_{v\in\mathbb{R}^{n+2}} & \sum_{k=0}^{n+1} |y_k - v_{k+1} + v_k| \\ \text{s.t.} & |v_k| \leq \lambda \ \ \forall k = 1,\ldots,N \ \text{ and } u_0 = u_n = 0. \end{array} \tag{6}$$

Once the solution v^* is computed, we recover the primal solution u^* by:

$$u_k^* = y_k - v_{k+1}^* + v_k^* \ \ \forall k = 0,\ldots,n. \tag{7}$$

The algorithm solves the dual problem (6) by defining the Karush-Kuhn-Tucker conditions [38, 29]:

$$\begin{cases} v_0^* = v_n^* = 0 & \text{and} & \forall k = 1,\ldots,n \\ v_k^* \in [-\lambda,\lambda] & \text{if} & u_k^* = u_{k+1}^* \\ v_k^* = -\lambda & \text{if} & u_k^* < u_{k+1}^* \\ v_k^* = \lambda & \text{if} & u_k^* > u_{k+1}^*. \end{cases} \tag{8}$$

The algorithm solves (6) by a non-iterative approach where it tries to construct the constant parts of the signal while respecting the constraints in (8). In more details, the algorithm starts by defining an index k_0 representing the start of the current plateau (for the first plateau $k_0 = 0$). Then for each point $k = k_0 + 1,\ldots,n$, the algorithm tries to extend the current plateau by adding k to it. There are three possible cases, corresponding to the comparison between u_{k-1}^* and u_k^*. In case if the equality $u_{k-1}^* = u_k^*$ respects the constraints in (8), it adds k and goes to the next index $k+1$. In contrast, if the equality can't respect (8), it declares $[k_0, k-1]$ as its own plateau, marks $k_0 = k$ as the start of a new plateau and continue from there. The sign and the height of the jump between u_{k-1}^* and u_k^* is computed from an estimated value of v_k^*. We refer the reader to [29] for a detailed description of the algorithm.

The solution u^* to the problem in (5) is generally over fitted as it suffers from the well known stair case effect which produces multiples jumps around the true discontinuity. To illustrate this effect, we used the algorithm to restore a signal depicted in (Fig. 2). The signal has a length of 1000 and 10 plateaux. The minimal jump between these plateaux is defined as $H = \min_k |u_{k+1} - u_k|$ and it equals to 4. We considered two noisy versions with a white Gaussian noise with a known standard deviation σ. In the first example (first row), the noise has a standard deviation $\sigma = 4$, and for the second example (second row) we increased the value of σ to 8. For a PWC signal the Signal to Noise Ratio (SNR) is defined as $\frac{H}{\sigma}$ which gives a SNR of 1 in the first case and 0.5 in the second. In (Fig. 2), we present the results of with a known standard deviation σ the two restorations in the first column. We could see that the solution has multiple false jumps marked by a red plus in the second column of the figure. We also remark that the number of this false jumps greatly increased with smaller values of the SNR.

The solution $\hat{u}$ to (5) is generally over fitted as it contains additional false jumps. This is due to the use of the convex L_1 regularizer which produces a stair-casing effect.

In order to filter all the false jumps produced by the TV solution u^*, we will store its segmentation results and use our algorithm called Discontinuity Position Sweep (DPS) to refine these jumps. More precisely, we will define a binary vector $p = (p_0,\ldots,p_n) \in \{0,1\}^n$ such as

$$p_i = \begin{cases} 0 & \text{if} \quad \hat{u}_{i+1} = \hat{u}_i \\ 1 & \text{otherwise.} \end{cases} \tag{9}$$

Any edge (u_i, u_{i+1}) such as $p_i = 0$ will be considered as continuous and its state will never change. In contrast, discontinuous edges (u_i, u_{i+1}), with $p_i = 1$, are questionable and need further processing as

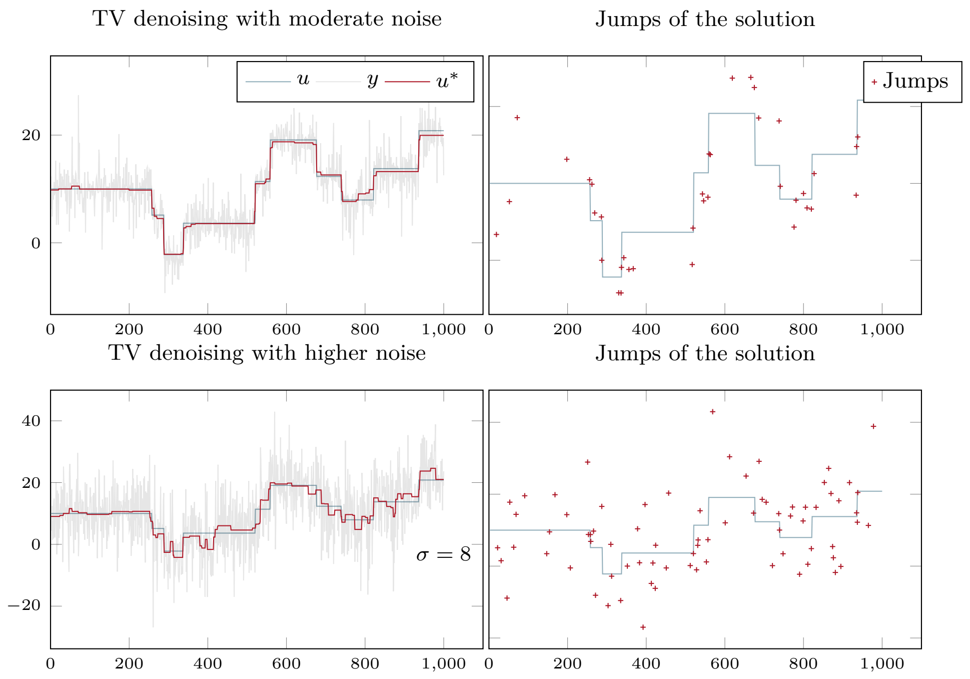

Figure 2. Total variation denoising with the L_1 prior. The signal contains 1000 data points, with a minimum jump $H = 4$. In the first example (first row), where the SNR=1, the denoiser catches all the abrupt jumps but produces additional false ones. All the produced jumps are marked by a red plus in the second column. For a higher noise (second row), the noise level corresponds to a SNR $= 0.5$, we can see that the number of false jumps increased considerably.

they could correspond to a false jump. Hence, when we apply our combinatorial algorithm DPS, we will never question the state of a continuous edge but only try to refine the results on the discontinuous ones.

This method is efficient as the number of jumps is too small compared to the signal size. Therefore, the number of positions considered by DPS will be greatly reduced. In more details, if we define K as the number of reported jumps by the TV solution u^*, then K is quite small compared to n and the ratio the of continuous edges

$$r = \frac{(n-K)}{n} \tag{10}$$

is closer to 1. For example, for the two restorations in (Fig. 2), r is 97% for the first row and 93% for the second.

2.2. Combinatorial line process search

Once we obtained the results of the TV denoiser segmentation p, will apply our algorithm DPS to filter the false jumps. We transform (4) into a purely combinatorial optimization problem over the LP l. But we will incorporate the results of the previous classifier as additional constraints to reduce the search domain $\mathcal{L}$.

The main idea of DPS is to eliminate the continuous variable u in the mixed problem (4). We observe that for a fixed line process l, the energy becomes

$$E_l(u) = \|u-y\|_2^2 + \lambda \sum_{i=0}^{n-1}(1-l_i)(u_{i+1}-u_i)^2 + \alpha \sum_{i=0}^{n-1} l_i. \tag{11}$$

This energy is quadratic and strictly convex. Furthermore, finding its unique solution comes at a low price, since it only involves solving a tridiagonal system. Given this fact, we rewrite (4) into a purely combinatorial problem over l and we propose an efficient search strategy (Fig. 3) to seek the optimum configuration.

The global unique minimum of (11) is characterized by the stationarity condition $\nabla E_l(u^*) = 0$ and is given as the solution of the following linear system:

$$(I + \lambda Q^T L_l Q)u^* = y, \tag{12}$$

where Q is a n by $n+1$ circulant matrix representing the difference operator, and L_l is the n diagonal matrix holding the coefficients of $l^{'} = 1 - l$ such as

$$Q = \begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \quad L_l = \begin{pmatrix} l_0^{'} & & & \\ & l_1^{'} & & \\ & & \ddots & \\ & & & l_{n-1}^{'} \end{pmatrix}. \tag{13}$$

Proof
Let's compute the partial derivative of $E_l(u)$ with respect to u_i.

$$\frac{\partial E_l}{\partial u_i} = 2\,(u_i - y_i) + 2\lambda\Big[-(1-l_i)(u_{i+1}-u_i) + (1-l_{i-1})(u_i - u_{i-1})\Big].$$

Hence each component of the optimal solution u_i^* is characterized by the following equation:

$$u_i^* + \lambda\Big[-(1-l_i)u_{i+1}^* + \big\{(1-l_i) + (1-l_{i-1})\big\}u_i^* - (1-l_{i-1})u_{i-1}^*\Big] = y_i. \tag{14}$$

We show that this equation could be obtained in a compact matrix form as stated in (12). Hence, let's compute the coefficients of the matrix $R_l = Q^T L_l Q$. We start by computing the coefficients of $Q^T L_l$:

$$\big(Q^T L_l\big)_{ij} = \sum_{k=0}^{n-1} Q_{ik}^T (L_l)_{kj} = Q_{ij}^T(1-l_j). \tag{15}$$

For the last equality, we used the fact that (L_l) is diagonal and the only nonzero element is $(L_l)_{jj} = (1-l_j)$. Now, we use the fact that only nonzero elements of Q^T are $Q_{i,i}^T = 1$ and $Q_{i+1,i}^T = -1$.

$$\big(Q^T L_l\big)_{ij} = \begin{cases} (1-l_i) & j = i \\ -(1-l_{i-1}) & j = i-1 \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Finally, we could compute the coefficients of our matrix $Q^T L_l Q$:

$$\begin{aligned} \big(Q^T L_l Q\big)_{ij} &= \sum_{k=0}^{n-1} \big(Q^T L_l\big)_{ik} Q_{kj} \quad (17)\\ &= (1-l_i)Q_{ij} - (1-l_{i-1})Q_{i-1,j}. \quad (18) \end{aligned}$$

Those coefficients are null except for the three cases $(i, i-1)$, (i, i) and $(i, i+1)$ with the following values:

$$\left(Q^T L_l Q\right)_{ij} = \begin{cases} -(1-l_{i-1}) & i = j+1 \\ (1-l_i) + (1-l_{i-1}) & i = j \\ -(1-l_i) & i = j-1 \\ 0 & \text{otherwise.} \end{cases} \tag{19}$$

Hence our proof, as we could write the set of equations in (14) as matrix multiplication:

$$\left(I + Q^T L_l Q\right) u^* = y$$

$\square$ $\square$

We denote E_l^* the energy associated to u^*. The mixed problem (4) is equivalent to finding the LP $l \in \mathcal{L} = \{0,1\}^n$ that minimizes E_l^*.

$$\min_{l \in \mathcal{L}} E_l^*. \tag{20}$$

An exhaustive search over all configurations of $\mathcal{L}$ has exponential complexity. Therefore, we structure the domain $\mathcal{L}$ into an n-hypercube. We recall that an n-hypercube is a regular graph (each node has exactly n edges) containing 2^n nodes and $2^{n-1}n$ edges [39]. The nodes of n-hypercube are encoded as n-bits using Gray code which ensures that two neighboring LP differ only in one bit.

The first pruning strategy eliminates all the incompatible line processes with the denoising results. That is, we only keep the configurations that are consistent with the previous segmentation p. First, we define the set of continuous edges (u_i, u_{i+1}) positions:

$$\mathcal{C} = \{i \mid p_i = 0\}. \tag{21}$$

According to the results of the TV segmentation, those edges are not candidates for a discontinuity. Hence any LP l that has a one in an index in $\mathcal{C}$ is not consistent with the TV findings and will be discarded. As a result, we denote the subset of the n-hypercube

$$\mathcal{L}_{\mathcal{C}} = \{l \mid l_j = 0 \ , \ \forall j \in \mathcal{C}\} \tag{22}$$

that regroups the set of line processes that are null in all the indices in $\mathcal{C}$. And we reduce the search space to this subset:

$$(P): \quad \min_{l \in \mathcal{L}_{\mathcal{C}}} E_l^*. \tag{23}$$

Additionally, DPS performs a breadth first search strategy where the depth of the search tree is defined by the levels of the hypercubes. For this purpose, we decompose the n-hypercube into disjoints sets $\mathcal{L}_k$, called levels. Each level $\mathcal{L}_k$ contains the line processes that have exactly k nonzero bits. Ultimately, we rewrite the problem (23) as a sequence of subproblems $\left((P_k)\right)_{k \in \{0,\dots,n\}}$ such as:

$$(P_k): \quad \min_{l \in \left(\mathcal{L}_k \cap \mathcal{L}_{\mathcal{C}}\right)} E_l^*. \tag{24}$$

The second pruning strategy, depicted in (Fig. 3), uses the information provided by the previous level to limit the search into the neighbors of its optimal solution. Suppose we get the optimal configuration $l^{k,*}$ of the problem P_k, the solution $l^{k,*}$ already gives us the positions of k discontinuities. So, in the search for the solution $l^{k+1,*}$ we keep those discontinuities, and we only seek to add a new one. This means that we add the constraint that $l^{k+1,*}$ must be a neighbor for $l^{k,*}$. We denote $\mathcal{V}(l^{k,*})$ the neighbors of $l^{k,*}$, and we transform the sequence $\big((P_k)\big)_{k\in\{0,\dots,n\}}$ defined in (24) into a recursive sequence $\big((Q_k)\big)_{k\in\{0,\dots,n\}}$ as follows:

$$\begin{cases} (Q_0)\ \min_{l\in\mathcal{L}_0} E_l^* \\ (Q_k)\ \min_{l\in\left(\mathcal{L}_k\cap\mathcal{L}_{\mathcal{C}}\cap\mathcal{V}(l^{k-1,*})\right)} E_l^* \quad 0\le k\le n \end{cases} \tag{25}$$

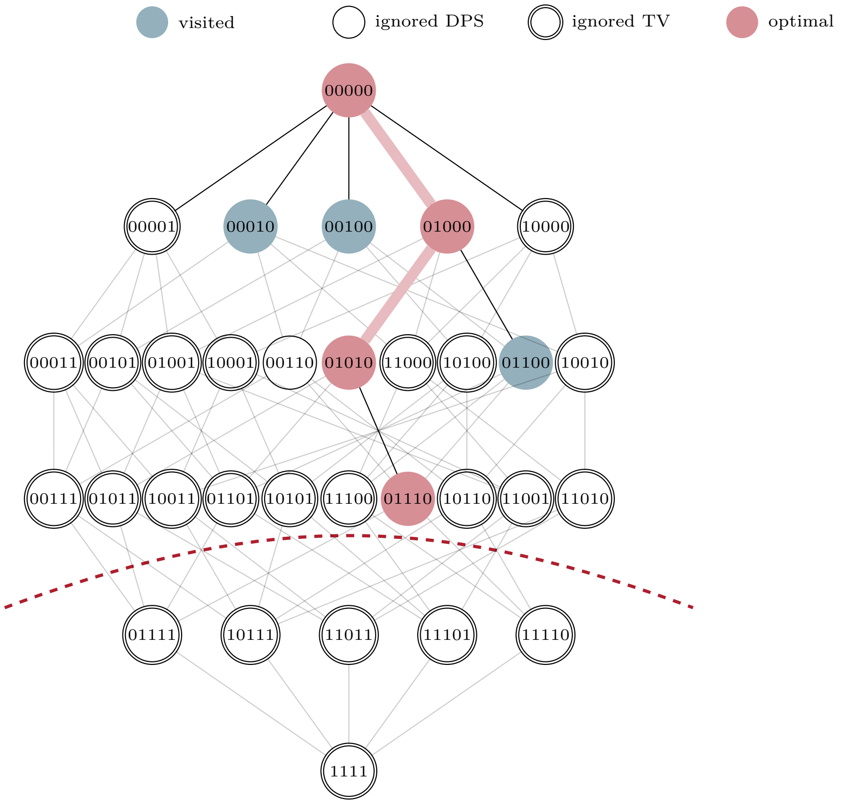

Figure 3. Pruning process of DPS: the example illustrates the search on 5-hypercube with the observation $\mathcal{C}=\{0,4\}$, i.e., any line process with a 1 on the first or fifth position will be discarded. The set of LP that is inconsistent with this observation is ignored (double drawn emtpy nodes). Single empty nodes are ignored by the DPS pruning process and blue nodes are visited. The red ones represent optimal configurations on their levels. The global optimum contains 2 steps 01010, but we must check the third level to verify if the energy could be decreased. The remaining levels are ignored (after red dashed lines) since the energy increased from $\mathcal{L}_2$ to $\mathcal{L}_3$.

The resulting algorithm (Alg.1) follows a top-down paradigm which starts with no discontinuities and at each level introduces a new step if the energy decreases. Otherwise, it stops the search and reports the actual solution as optimal (Alg.1).

2.3. Complexity analysis

The complexity of the algorithm is $\mathcal{O}(emn^2)$; n is the size of the signal; m is the number of jumps and $e=1-r$ where r is the ratio of continuous edges in the TV denoiser (10). Indeed, to restore such a signal, we call the TV denoiser, to attenuate the noise, that has a quadratic cost $\mathcal{O}(n^2)$ in the worst case [29]. After that, DPS checks en nodes in the first level $\mathcal{L}_0$, $en-1$ in the second $\mathcal{L}_1$, and so on so forth until the level $\mathcal{L}_{m+1}$ is reached. The energy of each node involves solving a tridiagonal symmetric system (12)

Algorithm 1 DPS algorithm with classified line process

```
Input: $\lambda$, $h$, $y$
  Find $\hat{u}$ in (5) using the proposed TV denoiser.
  Compute the set $\mathcal{C}$ of the classified continuous positions.
  $l^* = 0$        #initial line process with zeros: $l^*_k = 0\ \forall k \in \{0, \dots, n-1\}$
  $\alpha = \frac{\lambda h^2}{2}$ , $k = 0$        #initialization
  $u^*_0$ solution of $Q_0$        #Continuous solution
  $E^*_0 = E(u^*_0, l)$        # initial energy.
  repeat
    $k = k + 1$        #next level
    $(u^*_k, l^*) = \text{argmin}_{l \in \left(\mathcal{L}_k \cap \mathcal{L}_\mathcal{C} \cap \mathcal{V}(l^*)\right)} \|u - y\|_2^2 + \sum_{i=0}^{n-1}(1 - l_i)(u_{i+1} - u_i)^2 + \alpha l_i$        #optimum
    $E^*_k = E(u^*_k, l^*)$        #compute the optimal energy
  until $\left(E^*_k > E^*_{k-1}\right)$
Output: $u^*_{k-1}$        #the restored signal
```

by applying an adapted solver that costs $\mathcal{O}(n)$. The complexity of the two stages is:

$$\mathcal{O}(n^2) + \sum_{i=0}^{m+1} \mathcal{O}\big((en - i)n\big) = \mathcal{O}(emn^2). \tag{26}$$

As a result, the worst-case complexity of the algorithm is $\mathcal{O}(n^2)$ (the TV denoiser fails to classify any continuous edges). In practical situations, the elimination rate is very significant (Fig. 4), since the number of jumps is small compared to the number of observations. Practically, we could achieve a linear complexity $\mathcal{O}(mn)$.

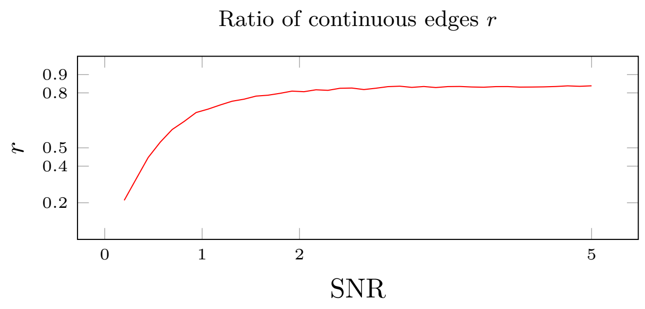

Figure 4. Ratio of continuous edges as a function of the SNR. The signal contains 200 data points with 10 equally distributed jumps.

3. Numerical experiments

In this section, we test the performances of our algorithm against a set of state of the art algorithms for jump detection and PWC signals reconstruction.

3.1. Example 1

In the first example, we apply our algorithm for a synthetic example depicted in (Fig. 5). The example is taken from [30] to exemplify the stair-casing effect that emerges from using the L_1 norm in the prior term. The signal $y \in \mathbb{R}^n$ contains $n = 1024$ observations with an additive white Gaussian noise with a standard deviation ($\sigma = 0.5$). The PWC original signal is produced by the function MakeSignal from the WaveLab framework using the 'blocks' option (Fig. 5a). For the two stages, we set $\lambda = 3$ and for DPS we choose a sensitivity $h = 1$.

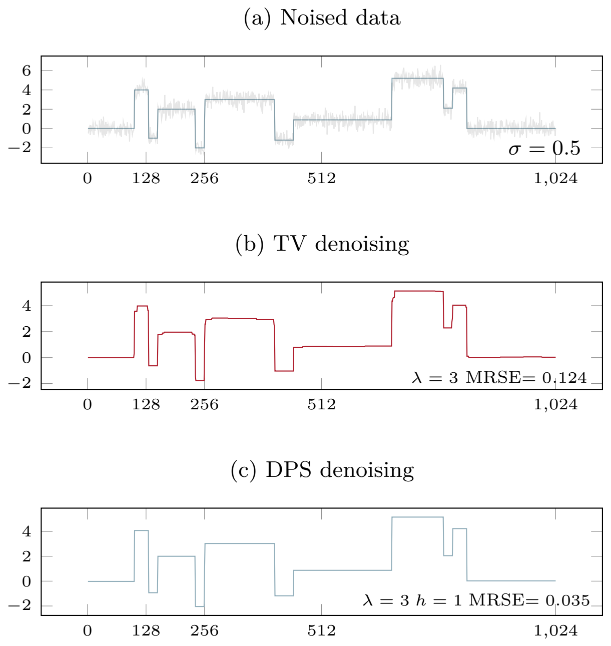

Figure 5. DPS denoising for the example in [30].

Figure 5c displays the improvement by the DPS algorithm, as all the false jumps produced by the TV denoiser (Fig. 5b) are eliminated. This is reflected by the Root Mean Square Error (RMSE) as DPS get a lower RMSE compared to TV denoiser. Furthermore, the DPS error is less than the one reported in [30] with nonconvex penalties. Table 1 shows the RMSE values of each algorithm.

	DPS	Condat TV denoiser	TV denoiser in [30]
RMSE	0.035	0.124	0.245

Table 1. RMSE comparison between DPS and a TV denoiser with nonconvex regularization ([30])

3.2. Example 2

In the second example, we consider the signal in [31] where the authors use a class of nonconvex penalties that tightly penalizes the sparsity of the signal than L_1 norm. The signal is synthetically generated to illustrate the stair-casing problem with two monotonous positive jumps with height a, $y \in \mathbb{R}^{200}$ and $y_{[1:50]} = a$, $y_{[51:100]} = 2a$ and $y_{[101:200]} = 3a$ (Fig. 6a). The noise is drawn from independently white Gaussian noise distribution with variance ($\sigma = 1$). DPS successfully filters all the false jumps generated by the TV denoiser (Fig. 6b) and produces the correct value of each plateau. In addition to that, we successfully repeat the experience for higher values of noise variance up to a standard deviation ($\sigma = 12$) (Fig. 6c).

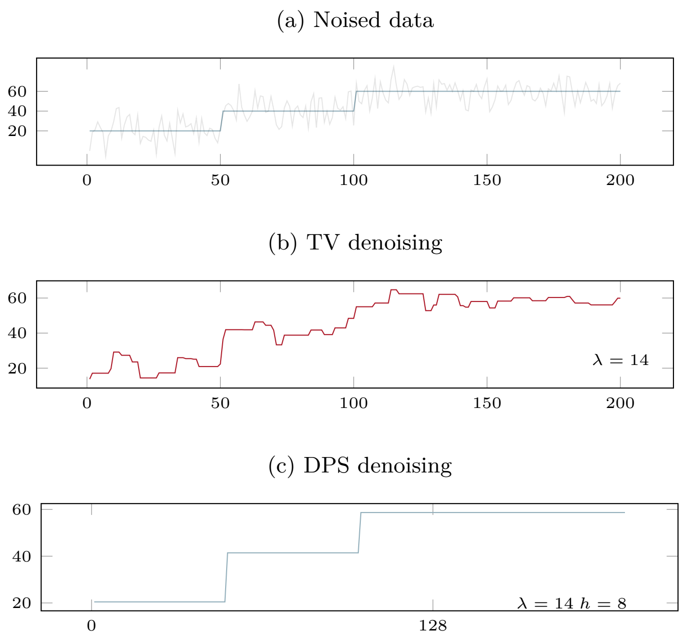

Figure 6. DPS denoising for the example in [31].

3.3. Example 3

In the third simulation, we apply the DPS algorithm to detect stepping motion in the trajectory of molecular motors in biology [40]. The detection of these steps allows the analysis of physical and chemical properties of single DNA based molecular machines such as polymerases [32]. The problem consists of finding motion positions from measures of the length of a molecular machine in base pair (bp) [41]. A bp is a unit consisting of two nucleobases bound to each other (e.g., A–T or G–T) and a motion is characterized by an abrupt change in the length of the molecular motor. We will use the example studied in [32], it contains experimental data of RNA polymerase II. The initial signal, depicted in (Fig. 7), contains $n = 5\ 10^4$ noisy measurements of the length of the polymerase in bp within a time interval $t \in [0, 2]$ in seconds.

We applied DPS to reconstruct and detect steps in this signal. We chose $\lambda = 10$ for the TV denoiser, $\lambda = 20$ and $h = 0.55$ for DPS. The reconstruction took $45ms$ and is depicted in (Fig. 7). In the same figure, we present the results with a state of the art algorithm called Energy Based Scheme (EBS) used in the same paper [32]. As shown in (Fig. 7), our algorithm has higher step detection quality as it only misses one hard jump in the middle while EBS missed several of them. Also in term of signal restoration, we compared the Mean Square Error (MSE) for both methods. The results are presented in Table 2 and shows that we obtained a lower MSE.

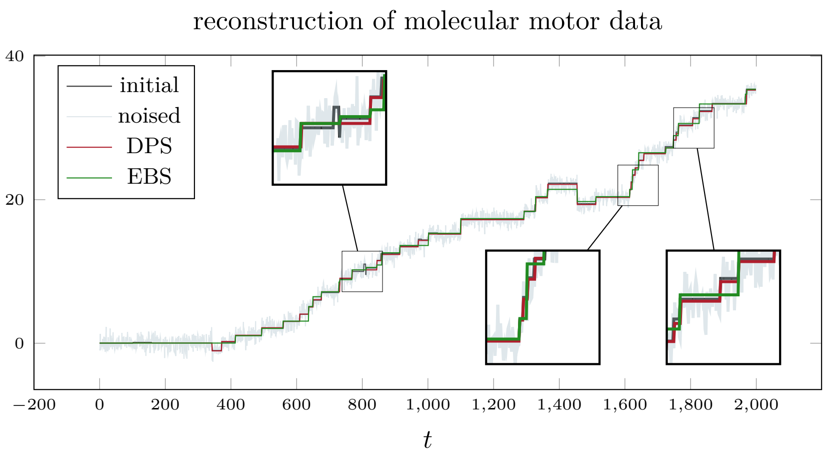

Figure 7. Reconstruction of molecular motor data by the DPS algorithm combined with the line process classifier.

In order to assess the reconstruction and segmentation quality of our algorithm, we compared the three classification metrics: precision, recall and f_1-score with a set of state of art algorithms. To define these metrics, we denote TP the number of correct jumps detected by the algorithm, FP the number of false jumps, FN the number of missed jumps and finally TN the number of true continuous position. The three metrics are expressed as follows:

$$\text{precision} = \frac{TP}{TP + FP}, \quad \text{recall} = \frac{TP}{TP + FN}, \quad f_1\text{-score} = \frac{2 \text{ precision recall}}{(\text{precision} + \text{recall})}. \tag{27}$$

Precision represents the ratio of correct reported jumps, a perfect value of 1 means that all the reported jumps are true. Recall measures the capacity of the algorithm in finding all the jumps. A perfect value of 1 in recall means that the algorithm detected all the jumps. Finally, the f_1-score leverages precision and recall by taking the harmonic mean.

	MSE	Precision	Recall	f_1 score
EBS	$13.57\ 10^{-1}$	0.55	0.87	0.67
DPS	23.8410^{-2}	0.71	0.97	**0.82**

Table 2. Reconstruction and Detection metrics for the recovery of the molecular motors mouvements

The compared algorithms are: PELT [42] which is a combinatorial algorithm with a pruning strategy that gives the optimal solution with the L_2 cost function. A Binary Segmentation scheme [43, 44] that offers a recursive algorithm to detect the jumps one by one. A Window sliding algorithm [45] that considers a local cost function over a window and slide it along the signal to detect the jumps one by one. Finally, we consider a Bottom-Up algorithm [46] that takes a dual approach to binary segmentation and start with a signal that has all the jumps and successively deletes the false ones. The results are depicted in (Fig. 8) for a sequence of noise standard deviation σ varying from 10^{-2} to 1.8. The figure shows that our algorithm offers the highest precision while keeping a stable recall. This superiority is reflected in the combined f_1 score where our algorithm kept a higher score $f_1 \approx 0.8$ for a high noise level $\sigma = 1.8$. In contrast, the f_1 score for the rest of the algorithms dropped significantly.

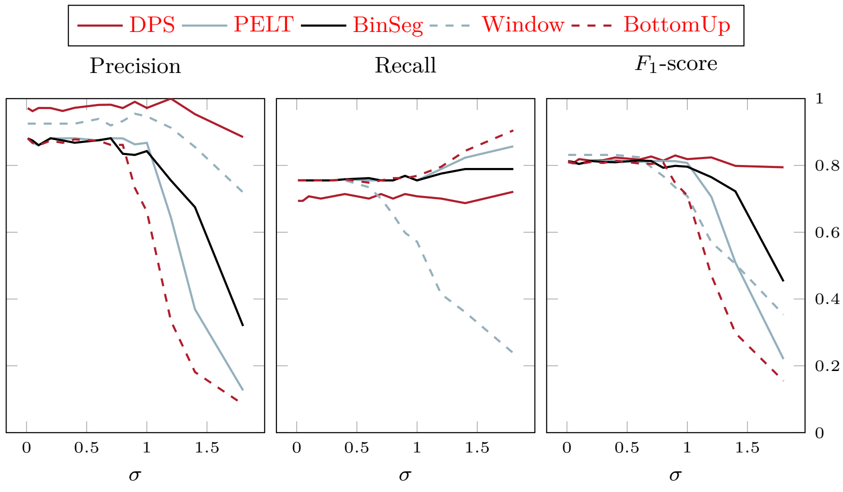

Figure 8. Detection metrics for the compared algorithms. The x-axis represents the noise standard deviation σ. The first figure (left) plots the precision. We could see from the figure that DPS has the highest precision. In the second figure (center), we plot the recall, our algorithm has lower recall but keeps it consistent even with strong noise. In the third figure (right), we combine the two metrics as the f_1-score, the figure shows that DPS offers the best detection quality.

3.4. Example 4

In the fourth example, we compare the classification results of the TV denoiser and the classical implementation of DPS to illustrate the differences between them. We consider the mean shift in the ruptures package [47]. Examples of these synthetic signals are depicted in (Fig 9). Each example is of size $N = 500$ and has 10 jumps. The goal is to consider several noise levels and measure the classification metrics. Hence, we generated a sequence of σ_i between $[0.1, 2]$ with a step 0.1. For each case, we measure the classification metrics for 10 corrupted signals. The results of those metrics are reported in (Fig. 10).

We could see that both algorithms have the same recall. This is the essence of our combination method as a missed point by the TV denoiser will most likely be missed by DPS. For the precision, the algorithms are different as DPS offers a better precision in low and moderate noise presence, $\sigma \in [0.1, 1.4]$. This is not the case for higher noise presence $\sigma > 1.4$ as the TV denoiser offer a better precision.

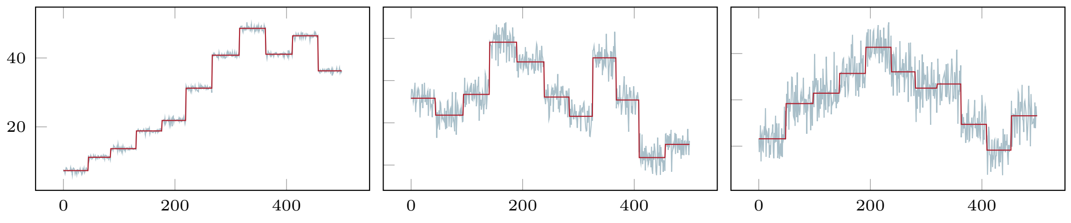

Figure 9. Examples of the mean shift signal from [47]. The signals has a size $n = 500$ and contain 10 jumps. Each one is corrupted with different noise standard deviation σ.

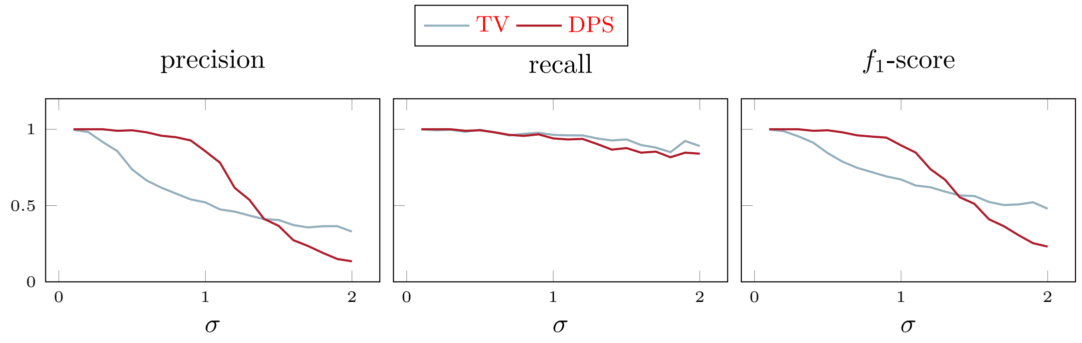

Figure 10. Classification metrics comparison between DPS and the TV denoiser.

3.5. Example 5

In this example, we compare the performances and running time of DPS, with and without the LP classifier.

In the first experience, we study the effect of the noise standard deviation σ on the f_1-score for both implementations. We considered a regular sequence of values $\sigma_i \in [0.5, 5]$ with a step 0.2. For each values σ_i, we generate 10 instances for the mean shift synthetic signal and we measure the f_1-score by both algorithms. The results of this simulation are depicted in (Fig.11) (left). We could see that for lower noise presence, both algorithms have the same score. Nevertheless, as the noise power becomes more important, the combined version is more robust with a higher score. This is due the TV denoiser that eliminates some false jumps that will never be considered by DPS.

In the second experience, we fix the noise standard deviation $\sigma = 1.4$ for a moderate signal presence. We also the fixed the sensitivity $h = 0.8$ to its best value and we study the effect of the regularization coefficient λ on the f_1-score. We thereby consider a sequence of $\lambda_i \in [5, 90]$ with a step 20. Same a previous experience, for each case of λ_i, we generate 10 examples of the mean shift synthetic signal and we measure the performances of each algorithm. We present the result of this experience in (Fig.11)(middle). We remark that for higher values of λ there is no difference between the two implementations. But for lower values, The combined version with the TV denoiser is more precise as the it helps to prune some false position from the LP hypercube. For the classical implementation of DPS, a lower λ would encourage smaller plateaux [19] and therefore will increase the number of false jumps.

In the last experience, we compared the f_1-score according to the sensitivity parameter h. We therefore fixed the value of $\lambda = 40$ and $\sigma = 1.4$ and we considered a sequence $h_i \in [0.4, 6]$ with a step 0.2. The results of this experience are depicted in (Fig.11)(right). Firstly, we see that both metrics follow the classical shape (increasing and decreasing). The first regime (with lower sensitivity) corresponds to an over fitted solution where DPS produces solutions with false jumps (lower precision). The second regime (higher sensitivity), DPS become more conservative and only accepts jumps with higher gradient and could therefore miss a true jump (lower recall). Secondly, we remark the same benefit of using TV denoiser for lower sensitivity value as its helps eliminates those false jumps. This is reflected in the f_1-score which is higher in the first regime while they are the same in the second one.

In the second part of this example, we compare the running time of DPS, with and without the LP classifier, as a function of the signal size, the number of jumps as well as the noise power. First, We generate 20 realizations of a PWC signal with sizes n ranging from 100 to 10^3. Each realization has 10 randomly distributed jumps with a minimal height $h = 10$ and corrupted with white Gaussian noise with a

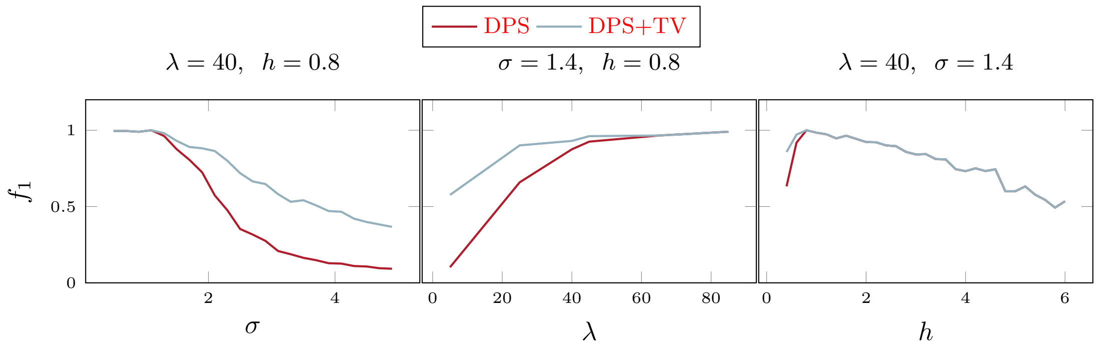

Figure 11. F_1-score comparison between DPS and our combined version on different hyper parameters and noise power. For the first scenario (left), we fixed the best choice for the hyper parameters λ and h and measured the f_1-score for an ascending noise standard deviation σ. We could remark that the combined version is more robust to noise as the TV denoiser helps to eliminate some false jumps. In the second simulation (middle), we tested the effect of λ on the f_1-score while fixing the sensitivity h and the noise power σ. Also in the situation, the combined version is more robust especially for lower values of λ where DPS tends to accepts false jumps. The main reason for this behavior is that λ also represents the length of the plateaux. Hence, a lower λ increases the number of plateaux and jumps. Finally, in the last simulation (right), we study the robustness of both algorithms according to the sensitivity parameter h. Here we observe a classical behavior, where the f_1-score increases until we get the best value $h \approx 0.8$ and then starts to decreases as larger values of h would get a conservative algorithm that only accepts larger jumps.

standard deviation $\sigma = 8$. (Fig 12a) plots the running time of each algorithm as a function of the signal size in a logarithmic scale. For the classical DPS, the experimental slope is $\alpha_1 = 2.002$ and, by incorporating the LP classifier, the slope goes down to $\alpha_2 = 1.5702$. Second, we fix the signal size to $n = 10^3$ and we consider 20 examples wherein we change the number of jumps from 10 to 50. The jumps are uniformly distributed over the signal and we keep the same noise power $\sigma = 8$. The outcome of the experiment is displayed in (Fig 12b) where the left axis (red line) represents the running time of the classical DPS with a experimental slope $\alpha_1 = 0.9329$ while the right axis (blue line) shows the running time using the LP classifier, the slope is reduced to $\alpha_2 = 0.6895$. Finally, (Fig. (12c) illustrates the running time as a function of the signal power. The figure confirms (in left axis) that the running time of the classical DPS is independent of the SNR [26]. Conversely, this property is lost for the new implementation (axis in the right), as the TV classifier is more sensible to noise and the rate of the predicted null processes decreases with lower SNR (Fig. 4).

4. Conclusion

In this paper, we proposed a two-stage efficient algorithm to restore noisy PWC signals while preserving their edges. The method uses TV denoising in the first stage to reduce the noise and classify the continuous edges. In the second stage, we apply a combinatorial algorithm to refine the TV findings by filtering false jumps. Compared to existing denoising schemes, our algorithm offers a blazingly fast time while keeping a superior restoration and step-detection quality especially for high-noise data.

The findings might also be extended to higher dimension data like 2-D images. The MRF graph is this case is the classical 2-D lattice with four neighbors system [17]. The generalization, called th weak membrane, for the LP model to theses models was already studied in [19]. The model suffers from a great computational complexity as the number of lines processes is exactly the number of edges the graph and is quadratic in the image size n. In order to reduce this complexity, the TV denoising could be applied to each row and column of the image to label the continuous edges. After that, we solve the week membrane

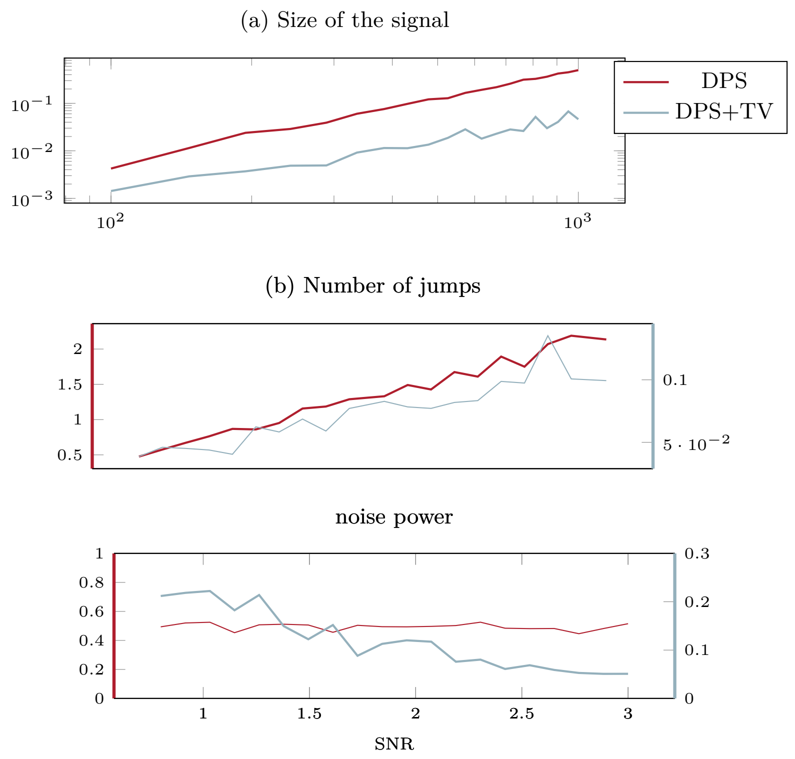

Figure 12. DPS running time as a function of the signal size (first row), number of jumps (second row) and the noise power (third row)

.

problem, but we never touch the labeled edges by the TV denoising. We are currently investigating this solution.

REFERENCES

1. J. V. Braun, R. K. Braun, H. G. Muller, Multiple changepoint fitting via quasilikelihood, with application to dna sequence segmentation, Biometrika 87 (2) (2000) 301–314.
2. A. B. Olshen, E. S. Venkatraman, R. Lucito, M. Wigler, Circular binary segmentation for the analysis of array-based dna copy number data, Biostatistics 5 (4) (2004) 557–572.
3. M. A. Little, N. S. Jones, Sparse Bayesian step-filtering for high-throughput analysis of molecular machine dynamics, in: 2010 IEEE International Conference on Acoustics Speech and Signal Processing, IEEE, 2010.
4. Y. Zhang, M. Brady, S. Smith, Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm, IEEE Transactions on Medical Imaging 20 (1) (2001) 45–57.
5. E. Andreou, E. Ghysels, Detecting multiple breaks in financial market volatility dynamics, Journal of Applied Econometrics 17 (5) (2002) 579–600.
6. L. Meligkotsidou, E. Tzavalis, I. Vrontos, On Bayesian analysis and unit root testing for autoregressive models in the presence of multiple structural breaks, Econometrics and Statistics (May 2017).
7. R. Ramlau, W. Ring, A mumford–shah level-set approach for the inversion and segmentation of x-ray tomography data, Journal of Computational Physics 221 (2) (2007) 539–557.
8. S. I. Kabanikhin, Inverse and Ill-posed Problems, Theory and Applications, DE GRUYTER, Berlin, Boston, 2011.
9. K. Hohm, M. Storath, A. Weinmann, An algorithmic framework for Mumford–Shah regularization of inverse problems in imaging, Inverse Problems 31 (11) (2015) 115011.
10. M. Storath, A. Weinmann, J. Frikel, M. Unser, Joint image reconstruction and segmentation using the Potts model, Inverse Problems 31 (2) (2015) 025003.
11. R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for markov random fields, in: Computer Vision – ECCV 2006, Springer Berlin Heidelberg, 2006, pp. 16–29.

12. O. Veksler, Graph cut based optimization for MRFs with truncated convex priors, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007.
13. M. Nikolova, Energy minimization methods, in: Handbook of Mathematical Methods in Imaging, Springer New York, 2011, pp. 139–185.
14. C. Deledalle, S. Vaiter, G. Peyré, J. Fadili, C. Dossal, Unbiased risk estimation for sparse analysis regularization, in: 2012 19th IEEE International Conference on Image Processing, 2012, pp. 3053–3056.
15. V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV, IEEE Comput. Soc, 2001.
16. S. Geman, D. Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6 (6) (1984) 721–741.
17. A. Blake, P. Kohli, C. Rother, Markov Random Fields for Vision and Image Processing, The MIT Press, 2011.
18. M. Nikolova, M. K. Ng, C.-P. Tam, Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction, IEEE Transactions on Image Processing 19 (12) (2010) 3073–3088.
19. A. Blake, A. Zisserman, Visual Reconstruction, MIT Press, Cambridge, MA, USA, 1987.
20. Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1222–1239.
21. M. Nikolova, Markovian reconstruction using a GNC approach, IEEE Transactions on Image Processing 8 (9) (1999) 1204–1220.
22. T. Pock, A. Chambolle, D. Cremers, H. Bischof, A convex relaxation approach for computing minimal partitions, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009.
23. M. Hurn, C. Jennison, An extension of Geman and Reynolds' approach to constrained restoration and the recovery of discontinuities, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (6) (1996) 657–662.
24. V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1568–1583.
25. Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, in: Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 359–374.
26. M. Douimi, H. Cherifi, Cutting enumerative algorithm for the minimizing of energy function., GRETSI, Trait. Signal 15 (1) (1998) 67–78.
27. M. A. Little, N. S. Jones, Generalized methods and solvers for noise removal from piecewise constant signals. i. background theory, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 467 (2135) (2011) 3088–3114.
28. M. A. Little, N. S. Jones, Generalized methods and solvers for noise removal from piecewise constant signals. II. new methods, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 467 (2135) (2011) 3115–3140.
29. L. Condat, A direct algorithm for 1-d total variation denoising, IEEE Signal Processing Letters 20 (11) (2013) 1054–1057.
30. I. W. Selesnick, A. Parekh, I. Bayram, Convex 1-d total variation denoising with non-convex regularization, IEEE Signal Processing Letters 22 (2) (2015) 141–144.
31. M. Malek-Mohammadi, C. R. Rojas, B. Wahlberg, A class of nonconvex penalties preserving overall convexity in optimization-based mean filtering, IEEE Transactions on Signal Processing 64 (24) (2016) 6650–6664.
32. J. Rosskopf, K. Paul-Yuan, M. B. Plenio, J. Michaelis, Energy-based scheme for reconstruction of piecewise constant signals observed in the movement of molecular machines, Physical Review E 94 (2) (aug 2016).
33. C. Couprie, L. Grady, L. Najman, J.-C. Pesquet, H. Talbot, Dual constrained TV-based regularization on graphs, SIAM Journal on Imaging Sciences 6 (3) (2013) 1246–1273.
34. L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D: nonlinear phenomena 60 (1-4) (1992) 259–268.
35. A. Chambolle, V. Caselles, D. Cremers, M. Novaga, T. Pock, An introduction to total variation for image analysis, Theoretical foundations and numerical methods for sparse recovery 9 (263-340) (2010) 227.
36. R. Gribonval, Should penalized least squares regression be interpreted as maximum a posteriori estimation?, IEEE Transactions on Signal Processing 59 (5) (2011) 2405–2410.
37. C. R. Vogel, M. E. Oman, Iterative methods for total variation denoising, SIAM Journal on Scientific Computing 17 (1) (1996) 227–238.
38. H. H. Bauschke, P. L. Combettes, et al., Convex analysis and monotone operator theory in Hilbert spaces, Vol. 408, Springer, 2011.
39. K. Thulasiraman, S. Arumugam, T. Nishizeki, A. Brandstädt, et al., Handbook of Graph Theory, Combinatorial Optimization, and Algorithms., Taylor & Francis, 2016.
40. Y. R. Chemla, K. Aathavan, J. Michaelis, S. Grimes, P. J. Jardine, D. L. Anderson, C. Bustamante, Mechanism of force generation of a viral dna packaging motor, Cell 122 (5) (2005) 683–692.
41. J. D. Watson, Molecular biology of the gene, Vol. 1, Pearson Education India, 2004.
42. R. Killick, P. Fearnhead, I. A. Eckley, Optimal detection of changepoints with a linear computational cost, Journal of the American Statistical Association 107 (500) (2012) 1590–1598.
43. J. Bai, Estimating multiple breaks one at a time, Econometric theory 13 (3) (1997) 315–352.
44. P. Fryzlewicz, et al., Wild binary segmentation for multiple change-point detection, The Annals of Statistics 42 (6) (2014) 2243–2281.
45. F. Desobry, M. Davy, C. Doncarli, An online kernel change detection algorithm, IEEE Trans. Signal Processing 53 (8-2) (2005) 2961–2974.

46. P. Fryzlewicz, Unbalanced haar technique for nonparametric function estimation, Journal of the American Statistical Association 102 (480) (2007) 1318–1327.
47. C. Truong, L. Oudre, N. Vayatis, Selective review of offline change point detection methods, Signal Processing 167 (2020) 107299.