Piecewise-polynomial Signal Segmentation Using Reweighted Convex Optimization

Michaela Novosadová and Pavel Rajmic
Signal Processing Laboratory (SPLab)
Dept. of Telecommunications, Faculty of Electrical Engineering and Communication
Brno University of Technology, Technická 12, 616 00 Brno, Czech Republic
Email: novosadova@phd.feec.vutbr.cz, rajmic@feec.vutbr.cz

***Abstract*—We present a method for segmenting a one-dimensional piecewise polynomial signal corrupted by an additive noise. The method's principal part is based on sparse modeling, and its formulation as a reweighted convex optimization problem is solved numerically by proximal splitting. The method solves a sequence of weighted ℓ_{21}-minimization problems, where the weights used for the next iteration are computed from the current solution. We perform experiments on simulated and real data and discuss the results.**

***Keywords*—proximal splitting algorithm; reweighted convex optimization; signal segmentation; signal smoothing; sparsity**

I. Introduction

Segmentation of signals is one of the most important application in digital signal processing. In this article we restrict to segmenting one-dimensional signals which are assumed to consists of several polynomial segments. The number of the segments is considerably lower than the total number of the observed signal samples. This fact suggests utilizing sparse signal processing techniques.

Segmentation is most often used in image processing. Our approach can be smoothly generalized to a higher dimension.

The recent publication [1], coping with segmentation of images, motivated this work. The authors of [1] use greedy [2] approach enabling the state-of-the-art results, and our goal was to explore how the convex relaxations of sparsity measures [3] will compare to it.

When the signal is assumed to lack jumps (i.e. the segments' borderpoints have to coincide), efficient methods have been presented, based on sparse modeling, see [4] or [5] in a more general setup.

To the best of our knowledge, there is no other work apart from [1] which would use *overparametrization* for segmenting the signal, and apart from [6] which, however, constructs nonconvex optimization problems.

In our previous articles [7], [8] and [9] we formulated various recovery problems aiming at segmentation. This article naturally follows on [9] and newly provides a formulation of a constrained convex problem and imitates nonconvexity.

The authors want to thank Nathanaël Perraudin, Michal Šorel and Marie Daňková for valuable discussion and comments. Research described in this paper was financed by the National Sustainability Program under grant LO1401 and by the Czech Science Foundation under grant no. GA16-13830S. For the research, infrastructure of the SIX Center was used.

Nonconvexity is imitated via a series of convex programs. This approach is not new in general, it has been both theoretically and practically justified by [10], for example. A series of convex problems is formulated, where parameters of the problem currently being solved depend on the latest solution. In our case such parameters are called weights and they are adaptively changed during iterations. In [9] we argued that such adaptation causes disbalance between the data fidelity term and the penalty, and this was the reason why a constrained problem was formulated in this paper.

Our segmentation algorithm consists of three main steps:

- Reweighting Condat Algorithm
- Detection of segments borders (breakpoints) in achieved parametrization coefficients
- Smoothing of detected segments.

II. Problem formulation

We have one-dimensional piecewise polynomial signal $\mathbf{y} = [y_1, \dots, y_N]^\top \in \mathbb{R}^N$. We suppose the signal consists of S segments. Each segment $s \in \{1, \dots, S\}$ can be described by the $K+1$ parametrization coefficients $x_{s,0}, x_{s,1}, \dots, x_{s,K}$ such that the i-th element of $\mathbf{y}$ is

$$y_i = x_{s,0} + x_{s,1} \cdot i + x_{s,2} \cdot i^2 + \cdots + x_{s,K} \cdot i^K + e_i, \quad (1)$$

where e_i represents an iid Gaussian noise corruptions. In vector notation, we can model the signal as

$$\mathbf{y} = \begin{bmatrix} \mathbf{I}\ \mathbf{D}^1 \dots \mathbf{D}^K \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \vdots \\ \mathbf{x}_K \end{bmatrix} + \mathbf{e} = \mathbf{A}\mathbf{x} + \mathbf{e} \quad (2)$$

where $\mathbf{x}_k = [x_{1,k}, x_{2,k}, \dots, x_{N,k}]^\top \in \mathbb{R}^N$ are $K+1$ parametrization vectors, $\mathbf{I} = \mathbf{I}_N$ is the identity matrix, $\mathbf{D} = \mathrm{diag}(1/N, 2/N, \dots, 1)$ is a diagonal matrix with linearly growing entries, $\mathbf{D}^j$ is its power (and therefore $\mathbf{D}^0 = \mathbf{I}$), and $\mathbf{e}$ represents the noise vector.

It is obvious that signal description by $\mathbf{A}\mathbf{x}$ is overparametrized. Each segment is built from polynomials up to degree K and the parameterization vectors $\mathbf{x}_i$ are piecewise constant within each segment. The breakpoints in all parametrization vectors lie on the same positions. Thanks to this assumption we formulate the optimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}\mathbf{x}\|_{21} \text{ s.t. } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \delta, \tag{3}$$

where $\mathbf{L} = \mathbf{W}\nabla$ and matrix $\mathbf{W}$ is defined as

$$\mathbf{W} = \begin{bmatrix} \operatorname{diag}(\mathbf{w}_0) & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \operatorname{diag}(\mathbf{w}_K) \end{bmatrix} \tag{4}$$

where $\mathbf{w}_k = \tau_k \mathbf{w}$ are $K+1$ vectors of weights. Vector $\mathbf{w} \in \mathbb{R}^{N-1}$ is a common vector for $\mathbf{w}_k$, which will be adaptively changed later. $\tau_0, \cdots, \tau_k$ are $K+1$ regularization coefficients, corresponding to individual polynomial degrees. They are set (and stay fixed) according to the noise level and prior experience.

The first term of (3) is the penalty. Piecewise constant vectors $\mathbf{x}_i$ suggest that these vectors are sparse under the difference operation ∇. We use the ℓ_{21}-norm [11] that acts on a matrix $\mathbf{Z}$ of size $p \times q$ and is formally defined by

$$\begin{aligned} \|\mathbf{Z}\|_{21} &= \Big\| \left[\|\mathbf{Z}_{1,:}\|_2, \|\mathbf{Z}_{2,:}\|_2, \ldots, \|\mathbf{Z}_{p,:}\|_2 \right] \Big\|_1 \\ &= \|\mathbf{Z}_{1,:}\|_2 + \ldots + \|\mathbf{Z}_{p,:}\|_2 \,, \end{aligned} \tag{5}$$

i.e. the ℓ_2-norm is applied to the particular rows of $\mathbf{Z}$ and the resulting vector is measured by the ℓ_1-norm. The ℓ_1-norm is usually taken as a convex substitute of the true sparsity measure [10], [12]. The second term is the data fidelity term. The Euclidean ℓ_2-norm reflects the fact that gaussianity of the noise is assumed and should be lower than the noise level δ. Finally, vector $\hat{\mathbf{x}}$ contains the achieved optimizers.

III. Algorithms

A. Proximal splitting methods

Proximal splitting methodology is a tool for iterative solution of convex minimization problems. Proximal algorithms are suitable for finding minimum of a sum of convex functions. Proximal algorithms (PA) perform iterations involving simple computational tasks such as evaluation of gradient or/and proximal operators related to each of the functions. It is proven that under mild conditions, PA provide convergence. The speed of convergence is influenced by character of functions and by the parameters used in the algorithms.

In this paper, we compare algorithms able to

$$\text{minimize } g(\mathbf{x}) + h_1(\mathbf{L}_1\mathbf{x}) \tag{6}$$

and

$$\text{minimize } h_1(\mathbf{L}_1\mathbf{x}) + h_2(\mathbf{L}_2\mathbf{x}), \tag{7}$$

respectively. Functions g, h_1 and h_2 are convex and $\mathbf{L}_1$, $\mathbf{L}_2$ are linear operators. The Condat algorithm [13] is used for solving both problems. For the problem (6) it is possible to use the Chambolle-Pock algorithm [14], which can be considered a special case of the Condat algorithm [13].

B. Proximal operators

A proximal operator (PO) of a function h maps $\mathbf{z} \in \mathbb{R}^N$ to another vector in $\mathbb{R}^N$. In this article proximal operators of several particular functions are used. The PO of the ℓ_{21}-norm is

$$\operatorname*{prox}_{\tau\|[\cdot,\ldots,\cdot]\|_{21}}(\mathbf{Z}) = \operatorname{soft}_\tau^{\text{row}}(\mathbf{Z}), \tag{8}$$

mapping matrix $\mathbf{Z} = [z_{ij}]$ to another [15]. It can be shown that it is a soft thresholding over groups consisting of matrix rows. The PO of an indicator function of a convex set C, denoted ι_C, is the orthogonal projection onto that set: $\operatorname{prox}_{\iota_C} = \operatorname{proj}_C$. Consequently, the PO of $\iota_{\{\mathbf{z}:\, \|\mathbf{y}-\mathbf{z}\|_2 \leq \delta\}} = \iota_{B_2(\mathbf{y},\delta)}$ is the projection on the ℓ_2-ball

$$\operatorname*{prox}_{\iota_{B_2(\mathbf{y},\delta)}}(\mathbf{z}) = \operatorname*{proj}_{B_2(\mathbf{y},\delta)}(\mathbf{z}) = \frac{\delta(\mathbf{z}-\mathbf{y})}{\max(\|\mathbf{z}-\mathbf{y}\|, \delta)}. \tag{9}$$

The PO of $\iota_{\{\mathbf{z}:\, \|\mathbf{y}-\mathbf{A}\mathbf{z}\|_2 \leq \delta\}}$ is projection onto an ℓ_2-ellipsoid and will be computed/approximated iteratively via Unlocbox [16] later in this paper.

Given a convex function f, the PO of its conjugate f^* can be computed at virtually the same cost as prox_f thanks to the Moreau identity [17], [13].

C. Condat algorithm

For solving our optimization problem (3) we can use the Condat algorithm (CA) able to solve problems of a very general form

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}) + \sum_{m=1}^{M} h_m(\mathbf{L}_m\mathbf{x}) \tag{10}$$

where f is a differentiable function, g and h_m are convex functions. The operators $\mathbf{L}_m$ are linear (and bounded). Simpler kinds of optimization problems are achieved by identifying, for example, some of the functions in (10) with zero, or letting $\mathbf{L}_m$ be the identity. The CA is a primal-dual iterative algorithm. Steps of the CA solving (10) can be found in [13].

In the following subsections we define two variants of the CA, which are reflecting the variation in CA for solution of two types of minimization problems (6), (7).

1) Variant 1 — CA_{gh}: We consider solving problem of type

$$\text{minimize } g(\mathbf{x}) + h_1(\mathbf{L}_1\mathbf{x})$$

where both functions g and h_1 are convex. With respect to the general formulation (10) we assign $f(\mathbf{x}) = 0$, $g(\mathbf{x}) = \iota_{\{\mathbf{z}:\, \|\mathbf{y}-\mathbf{A}\mathbf{z}\|_2 \leq \delta\}}(\mathbf{x})$ and $h_1(\mathbf{L}_1\mathbf{x}) = \|\mathbf{L}_1\mathbf{x}\|_{21} = \|\mathbf{W}\nabla\mathbf{x}\|_{21}$. Then our recovery problem is formulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_1\mathbf{x}\|_{21} + \iota_{\{\mathbf{z}:\, \|\mathbf{y}-\mathbf{A}\mathbf{z}\|_2 \leq \delta\}}(\mathbf{x}). \tag{11}$$

The algorithm solving problem (6) is given in Alg. 1.

Algorithm requires knowledge of proximal operators associated to g and h_1. We assign: prox_g is the projection onto $\{\mathbf{z} : \|\mathbf{y} - \mathbf{A}\mathbf{z}\|_2 \leq \delta\}$, and $\operatorname{prox}_{h_1}$ is the group soft

thresholding, as described in Sec. III-A. For the linear operator we have

$$\mathbf{L}_1(\mathbf{x}) = \mathbf{L}_1(\mathbf{x}_0, \dots, \mathbf{x}_K) = \begin{bmatrix} \operatorname{diag}(\mathbf{w}_0)\nabla\mathbf{x}_0 \\ \vdots \\ \operatorname{diag}(\mathbf{w}_K)\nabla\mathbf{x}_K \end{bmatrix} \tag{12}$$

with ∇ of size $(N-1)\times N$, and for its adjoint

$$\begin{aligned}\mathbf{L}_1^\top(\mathbf{u}) &= \mathbf{L}_1^\top(\mathbf{u}_0, \dots, \mathbf{u}_K) \\ &= \begin{bmatrix} \operatorname{diag}(\mathbf{w}_0)\nabla^\top & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \operatorname{diag}(\mathbf{w}_K)\nabla^\top \end{bmatrix} \mathbf{u}.\end{aligned} \tag{13}$$

Convergence of the CA$_{\text{gh}}$ is guaranteed when $\xi\sigma\|\mathbf{L}_1\|^2 \le 1$, therefore we need the upper bound on the operator norm $\|\mathbf{L}_1\|$:

$$\begin{aligned}\|\mathbf{L}_1\|^2 &= \max_{\|\mathbf{x}\|_2=1} \|\mathbf{L}_1\mathbf{x}\|_2^2 = \max_{\|\mathbf{x}\|_2=1} \left\| \begin{bmatrix} \tau_0 \operatorname{diag}(\mathbf{w})\nabla\mathbf{x}_0 \\ \vdots \\ \tau_K \operatorname{diag}(\mathbf{w})\nabla\mathbf{x}_K \end{bmatrix} \right\|_2^2 \\ &= \max_{\|\mathbf{x}\|_2=1} \left(\sum_{k=0}^{K} \|\tau_k \operatorname{diag}(\mathbf{w})\nabla\mathbf{x}_k\|_2^2 \right) \\ &\le \sum_{k=0}^{K} \left(\max_{\|\mathbf{x}\|_2=1} \|\tau_k \operatorname{diag}(\mathbf{w})\nabla\mathbf{x}_k\|_2^2 \right) \\ &\le \sum_{k=0}^{K} (\tau_i \max(\mathbf{w}))^2 \|\nabla\|^2 \le 4(\max(\mathbf{w}))^2 \sum_{k=0}^{K} \tau_k^2\end{aligned}$$

and thus $\|\mathbf{L}_1\| \le 2\max(\mathbf{w})\|\boldsymbol{\tau}\|_2$.

2) Variant 2 — CA$_{hh}$: Now we consider the variant to

$$\text{minimize } h_2(\mathbf{L}_2\mathbf{x}) + h_1(\mathbf{L}_1\mathbf{x})$$

where both h_1 and h_2 are convex. With respect to the CA, we assign $f(\mathbf{x}) = 0$, $g(\mathbf{x}) = 0$, $h_1(\mathbf{L}_1\mathbf{x}) = \|\mathbf{L}_1\mathbf{x}\|_{21} = \|\mathbf{W}\nabla\mathbf{x}\|_{21}$ and $h_2(\mathbf{L}_2\mathbf{x}) = \iota_{\{\mathbf{z}:\,\|\mathbf{y}-\mathbf{z}\|_2\le\delta\}}(\mathbf{L}_2\mathbf{x})$. Then our recovery problem is formulated as

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}_1\mathbf{x}\|_{21} + \iota_{\{\mathbf{z}:\,\|\mathbf{y}-\mathbf{z}\|_2\le\delta\}}(\mathbf{L}_2\mathbf{x}). \tag{14}$$

Algorithm 1: The Condat Algorithm solving (6) — CA$_{\text{gh}}$

Input: Functions g, h_1, linear operator $\mathbf{L}_1$ of size $(N-1)(K+1)\times N(K+1)$, $\mathbf{L}_1^\top$
Output: $\hat{\mathbf{x}} = \mathbf{x}^{(i+1)}$
Set parameters $\xi, \sigma > 0$ and $\rho \in [0, 2]$
Set initial primal variables $\mathbf{x}^{(0)} \in \mathbb{R}^{N(K+1)}$ and dual variables $\mathbf{u}_1^{(0)} \in \mathbb{R}^{(N-1)(K+1)}$
for $i = 0, 1, \dots$ **do**
$\bar{\mathbf{x}}^{(i+1)} = \operatorname{prox}_{\xi g}(\mathbf{x}^{(i)} - \xi\mathbf{L}_1^\top\mathbf{u}_1^{(i)})$
$\mathbf{x}^{(i+1)} = \rho\bar{\mathbf{x}}^{(i+1)} + (1-\rho)\mathbf{x}^{(i)}$
$\bar{\mathbf{u}}_1^{(i+1)} = \operatorname{prox}_{\sigma h_1^*}(\mathbf{u}_1^{(i)} + \sigma\mathbf{L}_1(2\bar{\mathbf{x}}^{(i+1)} - \mathbf{x}^{(i)}))$
$\mathbf{u}_1^{(i+1)} = \rho\bar{\mathbf{u}}_1^{(i+1)} + (1-\rho)\mathbf{u}_1^{(i)}$
return $\mathbf{x}^{(i+1)}$

It remains to define the linear operators: $\mathbf{L}_2 = \mathbf{A}$ (and $\mathbf{L}_2^T = \mathbf{A}^T$) and thus $h_2(\mathbf{L}_2\mathbf{x}) = h_2(\mathbf{A}\mathbf{x}) = \iota_{\{\mathbf{z}:\,\|\mathbf{y}-\mathbf{A}\mathbf{z}\|_2\le\delta\}}(\mathbf{x})$. $\mathbf{L}_1$ is as in variant 1, see Eq. (12). Algorithm solving (7) is described in Alg. 2.

The algorithm utilizes proximal operators of g, h_1 and h_2, for which we have: $\operatorname{prox}_g = Id$, $\operatorname{prox}_{h_1}$ is the projection onto $B_2(\mathbf{y}, \delta)$ and $\operatorname{prox}_{h_1}$ is the group soft thresholding.

Convergence of the CA$_{\text{hh}}$ is guaranteed when it holds $\xi\sigma\|\mathbf{L}_1^\top\mathbf{L}_1 + \mathbf{L}_2^\top\mathbf{L}_2\| \le 1$. We use the inequality $\|\mathbf{L}_1^\top\mathbf{L}_1 + \mathbf{L}_2^\top\mathbf{L}_2\| \le \|\mathbf{L}_1\|^2 + \|\mathbf{L}_2\|^2$, therefore we need the upper bound of $\|\mathbf{L}_1\|$ and $\|\mathbf{L}_2\|$. Obviously $\|\mathbf{L}_2\|^2 = \|\mathbf{A}\|^2$. Operator norm of $\mathbf{A}$ satisfies $\|\mathbf{A}\|^2 = \|\mathbf{A}\mathbf{A}^\top\|$ and thus it suffices to find the maximum eigenvalue of $\mathbf{A}\mathbf{A}^\top$. Since rows of $\mathbf{A}$ are mutually orthogonal, it can be shown that $\|\mathbf{A}\|^2 = K+1$. Operator $\mathbf{L}_1$ is as in variant 1, therefore $\xi\sigma(K+1+4(\max(\mathbf{w}))^2\|\boldsymbol{\tau}\|_2^2) \le 1$.

D. Reweighting

The weights used for the differences in the penalty are adaptively changed. The points which turn out to probably be the breakpoints would be gradually assigned lower weight, thus penalized less. In this way we extend our problem (3) to

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{L}^{(j)}\mathbf{x}\|_{21} \text{ s.t. } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \le \delta \tag{15}$$

where the penalty can change via $\mathbf{L}_1^{(j)}$, where j represents the counter for problem repetitions. The linear operator is changed after a suitable number of the CA iterations by recomputing the weights:

$$\left(\mathbf{w}^{(j+1)}\right)_s = \frac{1}{\left\| \left[\left(\nabla\hat{\mathbf{x}}_0^{(j)}\right)_s, \cdots, \left(\nabla\hat{\mathbf{x}}_K^{(j)}\right)_s \right] \right\|_p + \epsilon} \tag{16}$$

where $(\cdot)_s$ denotes the sth component of a vector, parameter $\epsilon > 0$ provides stability and ensures that a zero-valued component in $\nabla\mathbf{x}_k$ does not strictly prohibit a nonzero estimate at the next step. It is shown in [10] that ϵ should be set slightly smaller than the expected nonzero magnitudes of $\nabla\mathbf{x}$. The

Algorithm 2: The Condat Algorithm solving (7) — CA$_{\text{hh}}$

Input: Functions h_1, h_2, linear operators $\mathbf{L}_1$ of size $(N-1)(K+1)\times N(K+1)$, $\mathbf{L}_1^\top$, $\mathbf{L}_2$ of size $N(K+1)\times N(K+1)$, $\mathbf{L}_2^\top$
Output: $\hat{\mathbf{x}} = \mathbf{x}^{(i+1)}$
Set parameters $\xi, \sigma > 0$ and $\rho \in [0, 2]$
Set initial primal variables $\mathbf{x}^{(0)} \in \mathbb{R}^{N(K+1)}$ and dual variables $\mathbf{u}_1^{(0)}, \mathbf{u}_2^{(0)} \in \mathbb{R}^{(N-1)(K+1)}$
for $i = 0, 1, \dots$ **do**
$\bar{\mathbf{x}}^{(i+1)} = (\mathbf{x}^{(i)} - \xi(\mathbf{L}_1^\top\mathbf{u}_1^{(i)} + \mathbf{L}_2^\top\mathbf{u}_2^{(i)}))$
$\mathbf{x}^{(i+1)} = \rho\bar{\mathbf{x}}^{(i+1)} + (1-\rho)\mathbf{x}^{(i)}$
$\bar{\mathbf{u}}_1^{(i+1)} = \operatorname{prox}_{\sigma h_1^*}(\mathbf{u}_1^{(i)} + \sigma\mathbf{L}_1(2\bar{\mathbf{x}}^{(i+1)} - \mathbf{x}^{(i)}))$
$\mathbf{u}_1^{(i+1)} = \rho\bar{\mathbf{u}}_1^{(i+1)} + (1-\rho)\mathbf{u}_1^{(i)}$
$\bar{\mathbf{u}}_2^{(i+1)} = \operatorname{prox}_{\sigma h_2^*}(\mathbf{u}_2^{(i)} + \sigma\mathbf{L}_2(2\bar{\mathbf{x}}^{(i+1)} - \mathbf{x}^{(i)}))$
$\mathbf{u}_2^{(i+1)} = \rho\bar{\mathbf{u}}_2^{(i+1)} + (1-\rho)\mathbf{u}_2^{(i)}$
return $\mathbf{x}^{(i+1)}$

weights are based on the latest solution of the CA and since the weights get changed, it is necessary to recompute operators $\mathbf{L}_1^{(j)}, \mathbf{L}_1^{\top(j)}$ as well:

$$\mathbf{L}_1^{(j)}(\mathbf{x}_0, \cdots, \mathbf{x}_K) = \begin{bmatrix} \tau_0 \operatorname{diag}(\mathbf{w}^{(j)})\nabla\mathbf{x}_0 \\ \vdots \\ \tau_K \operatorname{diag}(\mathbf{w}^{(j)})\nabla\mathbf{x}_K \end{bmatrix} \tag{17}$$

The norm of the operator should be recomputed. The reweighting Condat algorithms rCA_{gh}/rCA_{hh} are presented in Alg. 3.

Algorithm 3: Reweighting Condat Algorithm — rCA_{gh}, rCA_{hh}

Input: Functions g, h_1, linear operators $\mathbf{L}_1$, $\mathbf{L}_1^\top$ (CA_{gh}); h_1, h_2, linear operators $\mathbf{L}_1$, $\mathbf{L}_1^\top$, $\mathbf{L}_2, \mathbf{L}_2^\top$ (CA_{hh}); number of iteration Y of CA, number of reweighting iteration X, parameter ϵ , p for ℓ_p-norm, τ_ks (CA_{gh}, CA_{hh})

Output: $\hat{\mathbf{x}} = \hat{\mathbf{x}}^{(j)}$

Set: initial vector of weights $\mathbf{w}^{(0)}$

Compute initial linear operators: $\mathbf{L}_1^{(0)}, \mathbf{L}_1^{\top(0)}, \|\mathbf{L}_1^{(0)}\|^2$

for $j = 0, 1, \ldots, X$ **do**

- Compute:
  Condat algorithm CA_{gh} or CA_{hh}
  - Input (CA_{gh}): $\mathbf{L}_1^{(j)}, \mathbf{L}_1^{\top(j)}, Y, g, h_1$
  - Input (CA_{hh}): $\mathbf{L}_1^{(j)}, \mathbf{L}_1^{\top(j)}, Y, h_2, h_1$
  - return: $\hat{\mathbf{x}}^{(j)}$
- Compute new weights: $\mathbf{w}^{(j+1)}$
- Compute: $\mathbf{L}_1^{(j+1)}, \mathbf{L}_1^{\top(j+1)}, \|\mathbf{L}_1^{(j+1)}\|^2$

return $\hat{\mathbf{x}}^{(j)}$

E. Signal segmentation/denoising

Achieved optimizers $\hat{\mathbf{x}}$ of problem (15) allow simple estimation of the underlying signal $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$.

The nonzero values in $\nabla\hat{\mathbf{x}}_0, \ldots, \nabla\hat{\mathbf{x}}_K$ indicate segment borders. In practice, it is almost impossible to tune regularization coefficients τ_ks to achieve truly piecewise constant optimizers [7] and therefore vectors $\nabla\hat{\mathbf{x}}_k$ are full of small values, besides larger values indicating possible segment borders. We apply a two-part procedure to obtain the segmented and denoised signal: first, the breakpoints are detected and then each detected segment is denoised individually.

The ℓ_{21}-norm cost causes significant values in $\nabla\hat{\mathbf{x}}_i$s situated at the same positions. We make a single vector of all achieved $\nabla\hat{\mathbf{x}}_i$s by computing the ℓ_p-norm according to the formula:

$$\mathbf{d} = \sqrt[p]{(\alpha_0\nabla\hat{\mathbf{x}}_0)^p + \cdots + (\alpha_K\nabla\hat{\mathbf{x}}_K)^p}, \tag{18}$$

where $\alpha_0, \ldots, \alpha_K$ are positive factors computed as $\alpha_k = 1/\max(|\nabla\hat{\mathbf{x}}_k|)$ serving to normalize the range of values in the parametrization vectors differences.

Then, a moving median filter is applied to $\mathbf{d}$ and the obtained filtered signal is subtracted from $\mathbf{d}$, yielding $\hat{\mathbf{d}}$. By this approach we keep the significant breakpoints and push small values more to zero in $\hat{\mathbf{d}}$. Values in $\hat{\mathbf{d}}$ satisfying $\left|\hat{\mathbf{d}}\right| > \lambda$ constitute the detected breakpoints. The threshold λ has to be set properly.

Occasionally, several breakpoints could be detected that are adjacent to each other. In such cases we pick the breakpoint having the largest absolute value.

The final step is denoising, which is applied on each detected segment separately. Denoising is done simply by the least squares method on data $\mathbf{y}$ restricted to the segment range. By doing this, we obtain a new parametrization coefficient set $\dot{\mathbf{x}}$, which is now constant on the segment-by-segment basis. By reconstruction according to $\dot{\mathbf{y}} = \mathbf{A}\dot{\mathbf{x}}$ we arrive at the denoised/segmented signal $\mathbf{y}$.

IV. Experiments

The experiments have been done in Matlab, and for some proximal algorithm components we benefited from using the flexible UnLocBox toolbox [16].

a) Goals: The main goal of our experiment is compare two variants of reweighted Condat algorithm rCA_{gh} and rCA_{hh} and find out which variant of reweighted Condat algorithm produces better results. Next question is if reweighted variants of CA (rCA_{gh}, rCA_{hh}) give better results than their nonweighted variants (CA_{gh}, CA_{hh}).

b) Quadratic signal with high noise: We present an experiment performed on a randomly generated piecewise polynomial signal of degree 2 and of total length $N = 300$. Parametrization vectors $\mathbf{x}_0, \mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathbb{R}^N$ used in the simulation are piecewise constant, according to the assumption. In our case, the signal contained six independent polynomial segments. Signal $\mathbf{y}$ is created using $[\mathbf{I}\ \mathbf{D}\ \mathbf{D}^2][\mathbf{x}_0^\top\ \mathbf{x}_1^\top\ \mathbf{x}_2^\top]^\top$ and the iid Gaussian noise is added on top of it such that the SNR is 19.88 dB.

The regularization parameters τ_0, τ_1, τ_2 have been tuned to obtain the parametrization coefficients as close to piecewise constant as possible. For the reweighted variants of CA have been the weights recomputed after each one iteration of the CA consisting of 300 iteration in total, the nonweighted variants of CA have been stopped after 300 iterations.

The results of presented algorithms are depicted in Fig. 1. We observe perfect detection of the changepoints and a good fit of the polynomial segments. Actually, the fit of the segments depends on the character of the noise in the segments. In the upper figure, the clean signal, its noisy version $\mathbf{y}$ and the denoised signal are shown. In all variants of CA were the breakpoints detected on the same positions and therefore the denoised signal for all variants is the same.

In the figure below, the respective parametrization coefficients are presented (i.e. the vectors $\hat{\mathbf{x}}_k$ subsequently used for the final breakpoint detection with median filter window of length 5 and with $\lambda = .1$). Convergence profiles of all used CA are depicted in Fig. 2.

The achieved signals of reweighted variants of CA (rCA_{gh}, rCA_{hh}) have higher SNR than their nonweighted versions (CA_{gh}, CA_{hh}). And the achieved signals of rCA_{hh} have higher SNR than of rCA_{gh}. The SNR is computed before signal segmentation/denoising.

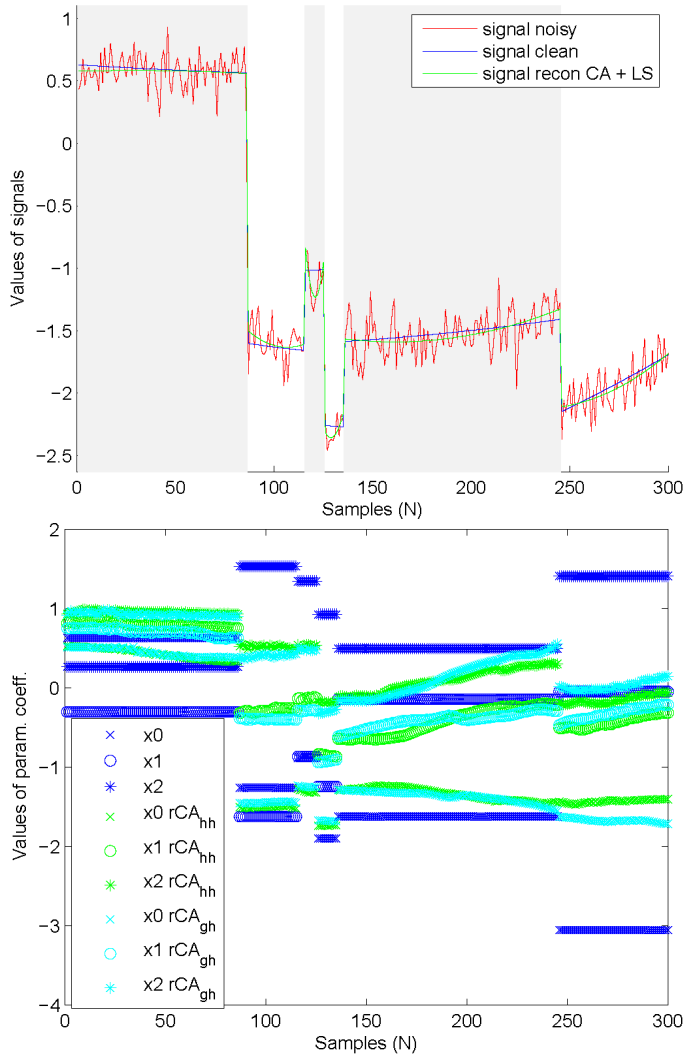

Fig. 1. Approximation of a random piecewise polynomial signal achieved with $\tau_0 = .72$, $\tau_1 = .51$, $\tau_2 = .36$, $\epsilon = 1$, $p = 2$. The SNR of the input signal was 19.88 dB, the SNR of the signal $\mathbf{A}\hat{\mathbf{x}}_{(\mathrm{rCA_{gh}})}$ was 25.62 dB, $\mathbf{A}\hat{\mathbf{x}}_{(\mathrm{CA_{gh}})}$ was 24.52 dB, $\mathbf{A}\hat{\mathbf{x}}_{(\mathrm{rCA_{hh}})}$ was 26.12 dB, $\mathbf{A}\hat{\mathbf{x}}_{(\mathrm{CA_{hh}})}$ was 24.89 dB and the SNR of the final estimate after the least squares method ($\mathbf{A}\dot{\mathbf{x}}$, shown in the upper figure in green) was 30.59 dB.

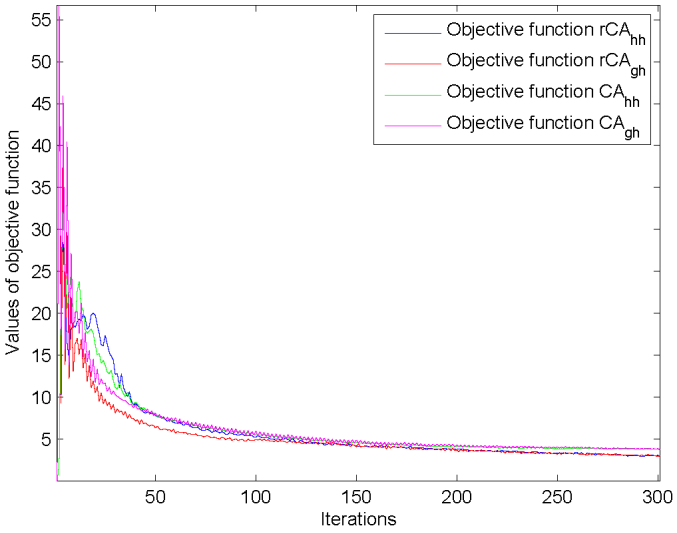

Fig. 2. Typical convergence profiles of four variants of CA solving our problem. The value of the objective is plot against the number of iterations. Variant $\mathrm{CA_{gh}}$ converges faster than variant $\mathrm{CA_{hh}}$, but computation time for one iteration of $\mathrm{CA_{gh}}$ is 3 times slower than of $\mathrm{CA_{hh}}$. Variant $\mathrm{CA_{hh}}$ is easier to implement.

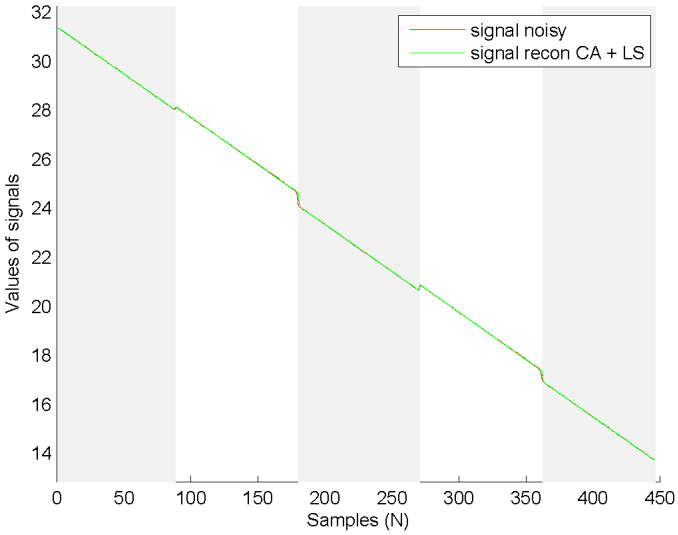

Fig. 3. Approximation of OTDR signal is achieved with $\tau_0 = 1.8$, $\tau_1 = 1.4$. After 1600 iterations of the CA algorithms (for reweighting version of CA have been weights recomputed after each 200 iteration) and subsequent postprocessing (with median filter length 25 and $\lambda = 0.09$).

c) Real data: We also present the experiment on real signal acquired by an OTDR (Optical Time Domain Reflectometer) instrument. OTDR [18] is a basic method used for evaluating the quality of optical routes in optical networks. The signal is 446 samples long, which corresponds to ca 55 kilometers long cable and was measured at wavelength 1310 nm. The signal consists of five segments. We got perfect detection of breakpoints by $\mathrm{rCA_{hh}}$, $\mathrm{rCA_{gh}}$ and $\mathrm{CA_{gh}}$ algorithms. The algorithm $\mathrm{CA_{hh}}$ detects four times more breakpoints than $\mathrm{rCA_{hh}}$. Therefore we can state that reweighting is promising way how to improve classical algorithms.

V. Conclusion

The paper shows that the presented methodology of signal segmentation/denoising is promising. However, careful tuning of the parameters must be carried out to obtain proper segmentation. It is shown that signal reconstructions achieved by the reweighted variants of the algorithms have higher SNR.

One could hope for better results via using more robust recovery programs, developed by introducing non-convex regularizers. For example using ℓ_p-"norms" with $p < 1$ [19] would be a natural step further. Our next step is the generalization of algorithms to higher dimension for image segmentation. We also want to explore the effect of using orthogonal polynomials instead of the basic ones in the model matrix $\mathbf{D}$.

REFERENCES

[1] R. Giryes, M. Elad, and A. Bruckstein, "Sparsity based methods for overparameterized variational problems," *SIAM journal on imaging sciences*, vol. 8, no. 3, pp. 2133–2159, 2015.

[2] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, pp. 2231–2242, 2004.

[3] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer, 2010.

[4] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 trend filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009. [Online]. Available: http://dx.doi.org/10.1137/070690274

[5] R. J. Tibshirani, "Adaptive piecewise polynomial estimation via trend filtering," *Annals of Statistics*, vol. 42, no. 1, pp. 285–323, 2014.

[6] S. Shem-Tov, G. Rosman, G. Adiv, R. Kimmel, and A. M. Bruckstein, *Innovations for Shape Analysis*, ser. Mathematics and Visualization. Springer, 2012, ch. On Globally Optimal Local Modeling: From Moving Least Squares to Over-parametrization, pp. 379–405.

[7] P. Rajmic and M. Novosadová, "On the limitation of convex optimization for sparse signal segmentation," in *Proceedings of the 9th International Conference on Telecommunications and Signal Processing*. Brno University of Technology, 2016, pp. 550–554.

[8] M. Novosadová and P. Rajmic, "Piecewise-polynomial curve fitting using group sparsity," in *Proceedings of the 8th International Congress on Ultra Modern Telecommunications and Control Systems*, Lisabon, 2016, pp. 317–322.

[9] P. Rajmic, M. Novosadová, and M. Daňková, "Piecewise-polynomial signal segmentation using convex optimization," *Submitted to Kybernetika*, 2017.

[10] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 12 2008.

[11] M. Kowalski and B. Torrésani, "Structured Sparsity: from Mixed Norms to Structured Shrinkage," in *SPARS'09 – Signal Processing with Adaptive Sparse Structured Representations*, R. Gribonval, Ed. Inria Rennes – Bretagne Atlantique, 2009, pp. 1–6. [Online]. Available: http://hal.inria.fr/inria-00369577/en/

[12] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proceedings of The National Academy of Sciences*, vol. 100, no. 5, pp. 2197–2202, 2003.

[13] L. Condat, "A generic proximal algorithm for convex optimization—application to total variation minimization," *Signal Processing Letters, IEEE*, vol. 21, no. 8, pp. 985–989, Aug 2014.

[14] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[15] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 303–324, 2009.

[16] N. Perraudin, D. I. Shuman, G. Puy, and P. Vandergheynst, "Unlocbox A Matlab convex optimization toolbox using proximal splitting methods," 2014. [Online]. Available: http://arxiv.org/abs/1402.0779

[17] N. Komodakis and J. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 31–54, Nov 2015.

[18] GN Nettest, *Understanding OTDR*. GN Nettest, 2 2000, rev. A. [Online]. Available: http://www.celemetrix.com.au/_literature_4374/Understanding_OTDRs

[19] R. Chartrand, "Shrinkage mappings and their induced penalty functions," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2014. [Online]. Available: http://math.lanl.gov/Research/Publications/Docs/chartrand-2014-shrinkage.pdf