

Efficient Iteratively Reweighted LASSO Algorithm for Cross-Products Penalized Sparse Solutions

1st David Luengo
Dep. of Audiovisual and Comm. Eng.
Universidad Politécnica de Madrid
Madrid, Spain
david.luengo@upm.es

2nd Javier Vía
Dep. of Communications Eng.
Universidad de Cantabria
Santander, Spain
javier.via@unican.es

3rd Tom Trigano
Dep. of Electrical and Electronics Eng.
Shamoon College of Engineering
Ashdod, Israel
thomast@sce.ac.il

Abstract—In this paper, we describe an efficient iterative algorithm for finding sparse solutions to a linear system. Apart from the well-known L_1 norm regularization, we introduce an additional cost term promoting solutions without too-close activations. This additional term, which is expressed as a sum of cross-products of absolute values, makes the problem non-convex and difficult to solve. However, the application of the successive convex approximations approach allows us to obtain an efficient algorithm consisting in the solution of a sequence of iteratively reweighted LASSO problems. Numerical simulations on randomly generated waveforms and ECG signals show the good performance of the proposed method.

Index Terms—sparsity-aware learning, LASSO, sparse coding, non-convex optimization

I. INTRODUCTION

The reconstruction of an unknown signal of interest from noisy observations is a common problem in signal processing. Often, this signal of interest corresponds to a sparse sequence of events that can be modelled as the output of a linear system whose input is a spike train. This happens, for instance, in electrocardiography, where the observed ECG is composed of a regular sequence of waveforms generated by the propagation of electrical impulses inside the heart. In this case, the spike train corresponds to the occurrence of the heartbeats and the linear system models the path from the heart to the sensor located on the body's surface [1], [2]. As a second example, we may consider the field of Gamma spectroscopy, whose objective is to identify radioactive sources and their activity. In this case, the spike train corresponds to the arrival times of the detected particles and the linear system models the measurement process [3], [4]. These two applications (as well as many others) have another important common characteristic: the negative co-occurrence of the events modelled by the spike train. Indeed, in the case of electrocardiograms (ECGs), after a cardiac cell activation there exists a so called *refractory period* where the cell cannot be excited [5]. Similarly, in the analysis of spectrometric data obtained through Type I counters, the detector (e.g., a Geiger counter) has an associated *dead time* during which it is unable to record another particle interaction [6]. These physical constraints impose inactivity periods that must be taken into account.

This work has been partly funded by the Spanish government through the KERMES excellence network (ref. TEC2016-81900-REDT).

On the one hand, co-occurrence has been extensively exploited in the field of computer vision [7], [8], but it has received much less attention in other signal processing and machine learning domains. On the other hand, many sparsity-aware signal processing approaches have been proposed since the introduction of the LASSO by Tibshirani in 1996 [9]: the *fused LASSO* [10], which imposes both sparsity and flatness of the obtained coefficients profile, the *elastic net*, which favors sparsity obtained by correlated variables [11], model-based compressive sensing [12], etc.

Unfortunately, very few approaches have been proposed so far to simultaneously encourage both sparsity and negative co-occurrence, i.e., a minimum distance among activations in the latent spike train. In [1]–[4] an initial spike train was obtained by means of the LASSO and a post-processing stage was then applied to keep only the strongest activations and remove those that did not respect the negative co-occurrence period. Then, in [13] a novel penalty term, based on the cross-products of the reconstruction coefficients, was added to the LASSO cost function in order to enforce negative co-occurrence. The resulting cross-products LASSO (CP-LASSO) approach resulted in a non-convex optimization problem that was solved using the successive convex approximations (SCA) technique. However, although the proposed algorithm showed a good performance, its computational cost prevented its application to large datasets.

In this paper, we introduce an efficient iterative algorithm to obtain an approximate solution of the CP-LASSO problem, through a sequence of iteratively reweighted LASSO problems, that allows us to deal with thousands of samples. The paper is organized as follows. Section II shows the problem formulation. The CP-LASSO cost function and its efficient minimization using the iteratively reweighted LASSO are described in detail in Section III. Then, Section IV shows the numerical results on randomly generated waveforms and ECG signals. Finally, Section V concludes the paper.

II. PROBLEM FORMULATION

Let us consider an observed discrete-time signal, $y[n]$, which corresponds to the output of an LTI system contaminated by noise, i.e.,

$$y[n] = s[n] * h[n] + w[n], \quad (1)$$

where $*$ denotes the standard linear convolution, $h[n]$ is the length L_h impulse response of the LTI system, $w[n] \sim \mathcal{N}(0, \sigma_w^2)$ is additive white Gaussian noise (AWGN) with zero mean and variance σ_w^2 , and $s[n]$ is a sparse latent sequence:

$$s[n] = \sum_{k=0}^{K-1} S_k \delta[n - m_k], \quad (2)$$

with K indicating the total number of non-zero elements (i.e., spikes, activations or events), S_k their amplitudes (which are strictly positive in some applications), and $\delta[n]$ denoting Kronecker's delta. Substituting (2) into (1), we obtain

$$y[n] = \sum_{k=0}^{K-1} S_k h[n - m_k] + w[n]. \quad (3)$$

Alternatively, (3) can be expressed more compactly as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{w}, \quad (4)$$

where $\mathbf{y} = [y[0], \dots, y[N-1]]^\top$ is the observation vector, $\mathbf{s} = [s[0], \dots, s[N-1]]^\top$ is the K -sparse latent activations vector ($K \ll N$) with

$$s[n] = \begin{cases} S_k, & \text{if } n = m_k; \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

\mathbf{H} is the $N \times N$ Toeplitz channel matrix,

$$\mathbf{H} = \begin{bmatrix} h[0] & 0 & \cdots & \cdots & 0 \\ h[1] & h[0] & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ h[L_h - 1] & \ddots & \ddots & h[0] & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & h[L_h - 1] & \cdots & h[0] \end{bmatrix}, \quad (6)$$

and $\mathbf{w} = [w[0], \dots, w[N-1]]^\top$ is the noise vector.

III. PROBLEM SOLUTION

A. Minimization Problem: Cross-Products Penalized Cost Function

Given the observation model in (4), our goal is finding a latent vector $\hat{\mathbf{s}}$ that fulfills the following conditions:

- $\hat{\mathbf{s}}$ is P -sparse with $P \ll N$ and $P \approx K$ (note that the exact value of K is often unknown in practice) in order to ensure a good localization ability of the latent activations.
- $\mathbf{H}\hat{\mathbf{s}}$ is a good approximation of the unknown noiseless signal $\mathbf{x} = \mathbf{H}\mathbf{s}$.
- Activations are not too close in order to respect the restrictions imposed by the addressed physical problem.

Taking into account these constraints, we can express our optimization problem as [13]

$$\begin{aligned} \underset{\mathbf{s}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} |s[k]| + \beta \sum_{k=0}^{N-1} \left(y[k] - \sum_{\ell=0}^{L_h-1} h[\ell] s[k-\ell] \right)^2 \\ & + \sum_{k=0}^{N-1} \sum_{\ell=0}^{N-1} g[k-\ell] |s[k] s[\ell]|. \end{aligned} \quad (7)$$

Here, with a slight abuse of notation, the first term represents the conventional sparsity promoting L_1 -norm penalization term. The second term measures the residual error, with the parameter β allowing us to establish the desired tradeoff between the residual error and the sparsity level of the recovered activation signals. Finally, the last term aims to promote solutions with not too close activations. Thus, the third term in (7) penalizes the cross-products between samples, where $g[k]$ can be considered as a penalization window, which in practice will be symmetric ($g[k] = g[-k]$) and of a rather small length ($g[k] = 0, \forall |k| > L_g$).

The reformulation of the optimization problem (7) in matrix/vector form yields

$$\underset{\mathbf{s}}{\text{minimize}} \quad \|\mathbf{s}\|_1 + \beta \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2 + |\mathbf{s}|^T \mathbf{G} |\mathbf{s}|, \quad (8)$$

where \mathbf{H} is given by (6), \mathbf{G} is a symmetric Toeplitz matrix with $g[k]$ in its $\pm k$ -th diagonal, and $|\mathbf{s}|$ denotes the element-wise absolute value of the vector \mathbf{s}

B. Non-Convex Optimization Algorithm: Iteratively Reweighted LASSO

Unfortunately, the optimization problem in (8) is not convex, and we have to resort to some approximations in order to find a local solution. However, the application of the successive convex approximations (SCA) approach provides a simple method which can be easily interpreted as an iteratively reweighted LASSO algorithm. Thus, considering a candidate solution $\hat{\mathbf{s}}$, we can approximate the last term in (8) as

$$|\mathbf{s}|^T \mathbf{G} |\mathbf{s}| \simeq 2 |\hat{\mathbf{s}}|^T \mathbf{G} |\mathbf{s}| - |\hat{\mathbf{s}}|^T \mathbf{G} |\hat{\mathbf{s}}|, \quad (9)$$

which allows us to rewrite the optimization problem in (8) as

$$\underset{\mathbf{s}}{\text{minimize}} \quad \gamma^T |\mathbf{s}| + \beta \|\mathbf{y} + \mathbf{H}\mathbf{s}\|_2^2, \quad (10)$$

with $\gamma = \mathbf{1} + 2\mathbf{G}|\hat{\mathbf{s}}|$.

Interestingly, the above problem reduces to a weighted LASSO problem, where the weights in the vector γ summarize the penalization factors due to the sparsity promoting L_1 term, and the cross-product penalizations based on the candidate solution $\hat{\mathbf{s}}$. With this in mind, and after the introduction of an adaptive step-size α , the Iteratively Reweighted LASSO algorithm is summarized in Algorithm 1.

From the description in Algorithm 1, it is easy to see that the cost function is decreasing with the iterations, which suffices to guarantee the convergence to a local minimum. Note also that the starting point for our algorithm corresponds exactly to the LASSO solution. Consequently, Algorithm 1 is always going to attain a lower value of the cost function than the LASSO algorithm and with the added benefit of enforcing a minimum distance among the detected events. Finally, let us point out that all the matrix-vector products in the weighted LASSO for solving (10), as well as in Algorithm 1, involve Toeplitz matrices, and therefore can be very efficiently implemented as linear filterings.

Algorithm 1 Iterative Reweighted LASSO Algorithm for solving the Cross-Product Penalized Problem in (8).

Inputs: Observations \mathbf{y} , channels \mathbf{H} , and penalization parameters β and \mathbf{G} .

Output: Estimated Activations \mathbf{s} .

Initialize $\mathbf{s} = \mathbf{0}$, $\gamma = \mathbf{1}$, $\alpha = 1$, Cost = ∞ .

repeat

Solve the problem in (10) with the LASSO algorithm.

Modify the solution $\tilde{\mathbf{s}}$ of (10) as $\tilde{\mathbf{s}} \leftarrow (1 - \alpha)\mathbf{s} + \alpha\tilde{\mathbf{s}}$.

Evaluate the cost function (8) in $\tilde{\mathbf{s}}$.

if The cost function has decreased **then**

Update $\mathbf{s} = \tilde{\mathbf{s}}$, $\alpha \leftarrow (1 + \alpha)/2$.

else

Update $\alpha \leftarrow \alpha/2$

end if

Update the weights $\gamma \leftarrow \mathbf{1} + 2\mathbf{G}|\mathbf{s}|$.

until Convergence

IV. NUMERICAL RESULTS

A. Case Study 1: Random Shapes

In this first case study we consider randomly generated shapes, $h[n] \sim \mathcal{N}(0, 1)$ for $0 \leq n \leq L_h - 1$ and $L_h = 12$. We generate $N = 10^4$ observed samples, $y[n]$ for $0 \leq n \leq N - 1$, with regularly occurring activations every $M = 100$ samples plus a random delay $\Delta M_k \sim \mathcal{U}(\{1, 2, \dots, 10\})$ for $k = 0, 1, \dots, K - 1$ with $K = N/M = 1000$. Therefore, the activation times are $m_k \sim \mathcal{U}(\{100k+1, 100k+2, \dots, 100k+10\})$ for $0 \leq k \leq K - 1$ and their amplitudes are also normally distributed: $S_k \sim \mathcal{N}(0, 1)$. For the restriction period, we set $L_g = 30$, whereas the parameter $\beta^2 = C/\sigma_w^2$ for several tested values of C . As performance measures, we consider both the reconstruction SNR,

$$\widehat{\text{SNR}}(\text{dB}) = 10 \cdot \log_{10} \frac{\mathbf{x}^\top \mathbf{x}}{(\mathbf{x} - \mathbf{H}\hat{\mathbf{s}})^\top (\mathbf{x} - \mathbf{H}\hat{\mathbf{s}})} \quad (11)$$

and the recovered sparsity, measured as the number of non-zero values in $\hat{\mathbf{s}}$. We test a range of SNR values from 0 to 50 dB, with the constant C ranging from 0.001 to 0.05, performing 100 simulations for each case.

TABLE I

RECOVERED SPARSITY AND RECONSTRUCTION SNR FOR THE OPTIMUM VALUE OF β THAT ATTAINS A SPARSITY SIMILAR TO THE TRUE ONE.

SNR(dB)	10	20	30	40	50
β^2	0.1581	0.1000	0.1581	0.1000	0.1581
sparsity	0.0046	0.0062	0.0098	0.0096	0.0101
$\widehat{\text{SNR}}(\text{dB})$	2.89	7.17	19.93	25.90	39.52

Figure 1 shows the $\widehat{\text{SNR}}(\text{dB})$ as a function of the input SNR for different values of C , whereas Figure 2 shows the sparsity of $\hat{\mathbf{s}}$. From Fig. 1 we notice that the optimum value of β depends on the SNR of the observations, with smaller values of β providing better results as the SNR increases. However, from Fig. 2 we can also see that the improved reconstruction SNR may come at the cost of a lack of sparsity in the recovered

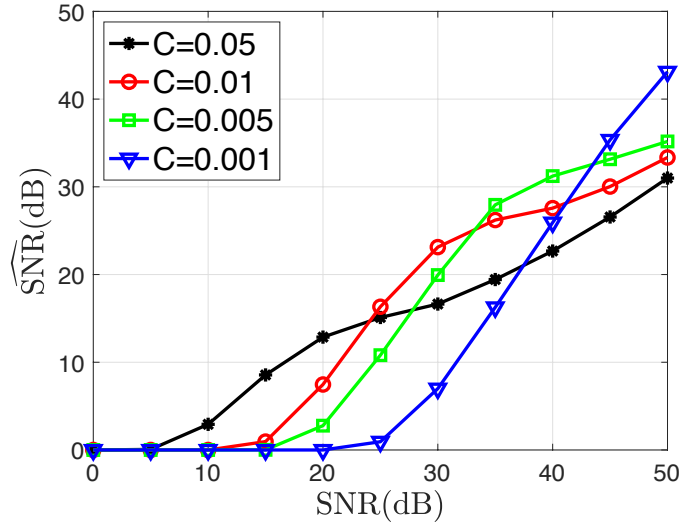


Fig. 1. Recovered SNR, $\widehat{\text{SNR}}(\text{dB})$ as defined in (11), for different SNR values in the observed sequence ($\sigma_w^2 = 10^{-\text{SNR}(\text{dB})/10}$) and $\beta^2 = C/\sigma_w^2$.

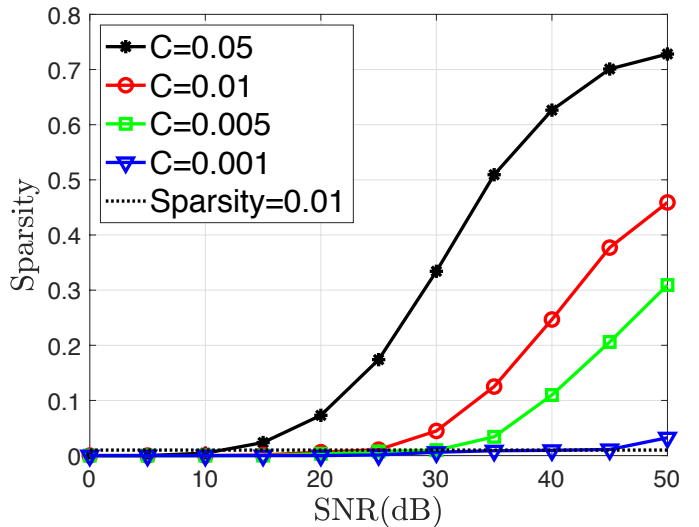


Fig. 2. Sparsity of the recovered $\hat{\mathbf{s}}$ for different SNR values in the observed sequence ($\sigma_w^2 = 10^{-\text{SNR}(\text{dB})/10}$) and $\beta^2 = C/\sigma_w^2$.

$\hat{\mathbf{s}}$. Note that the largest the value of β the more emphasis we place on a good reconstruction and the less importance we give to the sparsity of $\hat{\mathbf{s}}$. Therefore, if we do not want to lose the spike localization ability provided by a sparse recovered latent sequence $\hat{\mathbf{s}}$, we need to select the proper value of β that allows us to achieve the desired sparsity. For instance, for an SNR = 10 dB in the observations the optimum choice (out of the tested values) is $\beta^2 = 0.1581$ (i.e., $C = 0.05$), as this approach provides a 0.0046 average sparsity in $\hat{\mathbf{s}}$ and the best reconstruction SNR, $\text{SNR}(\text{dB}) = 2.89$. Interestingly, the optimum value of $\beta^2 = C/\sigma_w^2$ in order to achieve a good reconstruction with a sparse latent signal seems to remain quite stable as the SNR in the observations increases. Table I shows the reconstruction SNR for the optimum value of β (out of the tested ones) that attains a sparsity level similar to

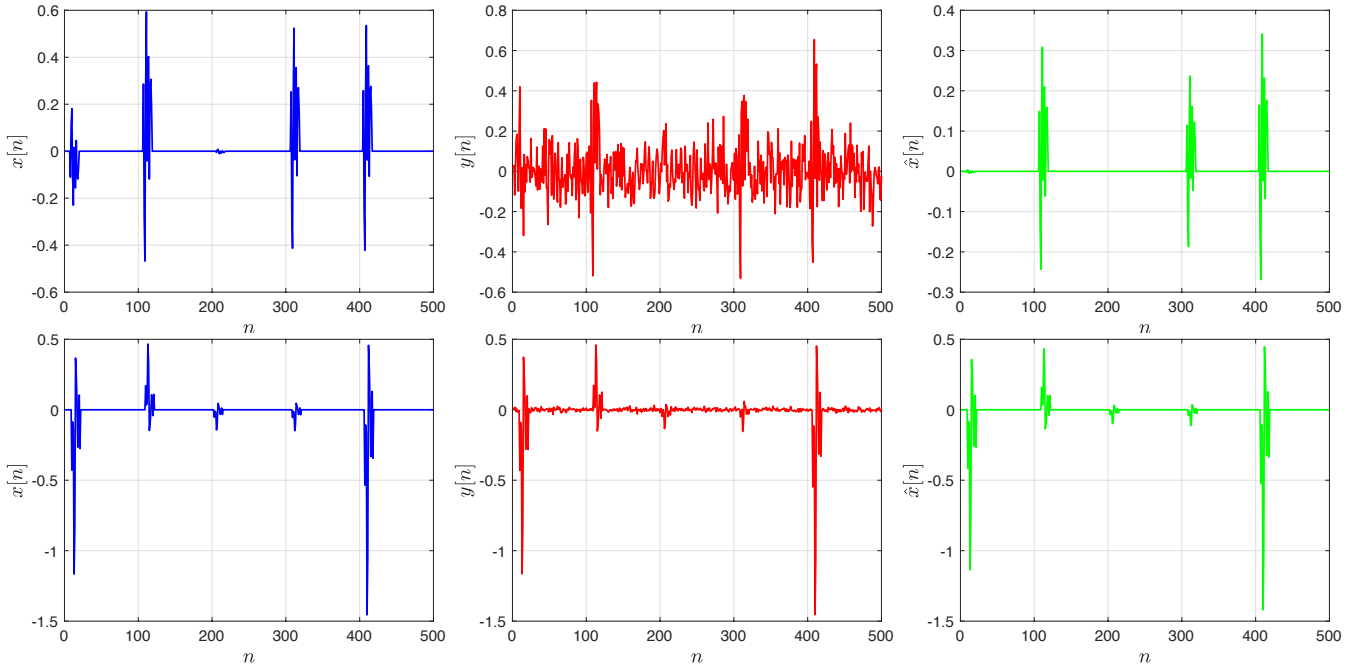


Fig. 3. **Example 1:** Unobserved clean signal $x[n]$ (left), observed noisy signal $y[n]$ (middle), and recovered signal $\hat{x}[n]$ (right) for two noise levels: SNR = 20 dB (upper row) and SNR = 40 dB (lower row).

the sparsity level of the original signal (0.01). Note that, even though the optimum value of C decreases as the observations SNR increases, as we set $\beta^2 = C/\sigma_w^2$, the optimum value of β is always either $\beta^2 = 0.1$ or $\beta^2 = 0.1581$. Finally, Figure 3 shows an example of the unobserved clean signal $x[n]$, the observed noisy signal $y[n]$, and the recovered signal $\hat{x}[n]$ for a low SNR situation (SNR = 20 dB) and a high SNR case (SNR = 40 dB). Note that all the spikes are properly detected in the high SNR case, whereas the small ones are masked by the noise (and thus cannot be recovered) in the low SNR case.

B. Case Study 2: Electrocardiographic (ECG) Signals

As a second case study, we consider an electrocardiographic (ECG) signal. ECG signals are composed of regularly occurring waveforms (the P and T waveforms and the QRS complex) and no activity intervals (isoelectric periods), contaminated by noise and interference. Therefore, they are intrinsically sparse signals with a latent sequence of activations corresponding to the appearance of the different waveforms. Furthermore, due to physiological restrictions a *refractory period* of 200–300 ms, during which no activations can occur, exists after each activation. Consequently, ECG signals fit perfectly within our model.

In this example, we focus on the detection of the R peak in the QRS complex, since this is the largest peak in the ECG and its detection is commonly the first stage in any ECG signal processing algorithm [14], [15]. Hence, for the dictionary we consider the typical shape of a QRS complex as extracted from the first heartbeat generated by the ECGSYN synthetic ECG generator [16]. Figure 4 shows this QRS complex: the $L_h = 70$ samples from the QRS onset to QRS offset are

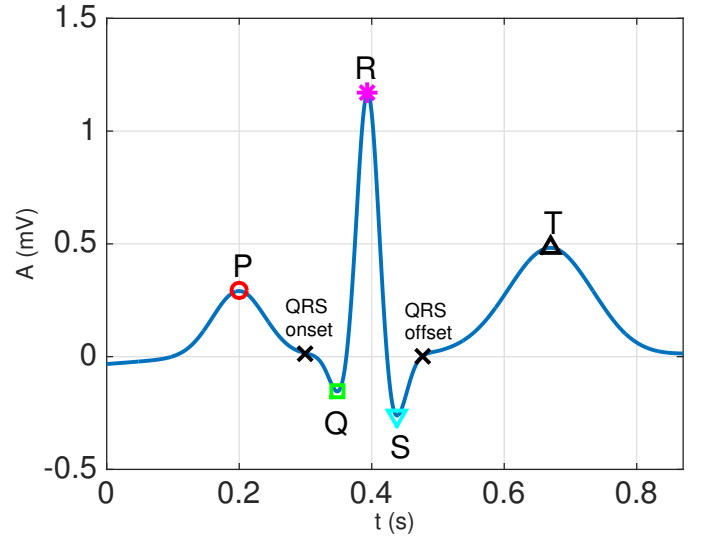


Fig. 4. Extracted P-QRS-T waveforms. The normalized QRS complex (from QRS onset to QRS offset) is used as $h[n]$ to build matrix \mathbf{H} .

used (after normalization) as $h[n]$ to build matrix \mathbf{H} . Then, Figure 5 shows an example of the unobserved clean signal $x[n]$, generated from another run of ECGSYN with $f_s = 360$ Hz and average heart rate (HR) of 60 bpm (with 6 bpm of standard HR deviation), the observed noisy signal $y[n]$, for an SNR = 20 dB, and the recovered signal $\hat{x}[n]$. Note that all the QRS complexes seem to have been properly detected. In order to verify this, we apply the well-known Pan-Tompkins algorithm [17] to the recovered signal $\hat{x}[n]$, and compare the result of the R peak detection w.r.t. the clean signal $x[n]$.

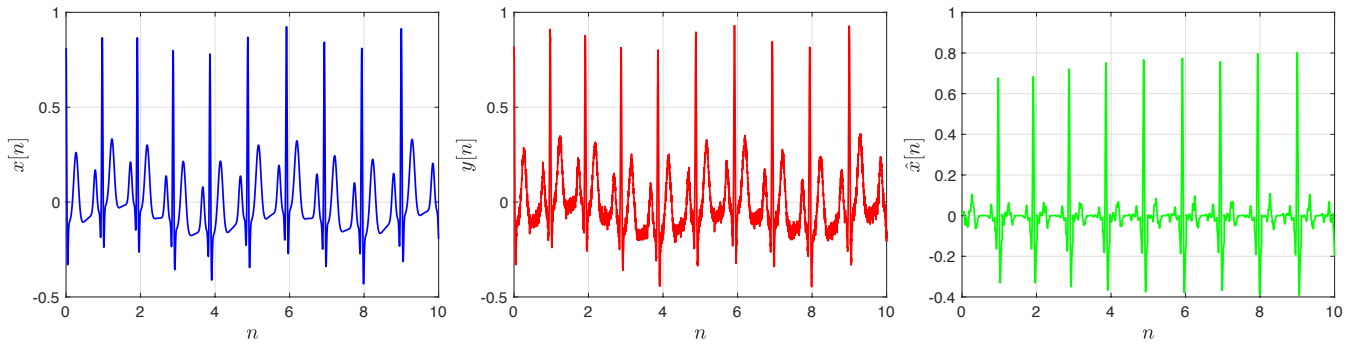


Fig. 5. **Example 2:** Unobserved clean signal $x[n]$ (left), observed noisy signal $y[n]$ (middle), and recovered signal $\hat{x}[n]$ (right) for an SNR = 20 dB.

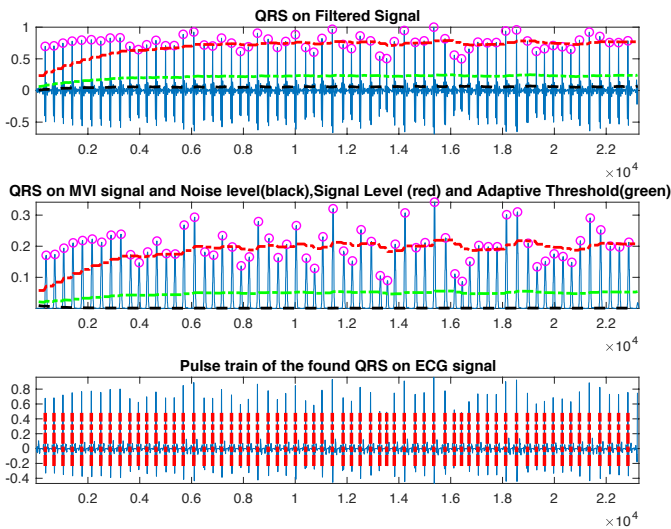


Fig. 6. Output of the Pan-Tompkins algorithm using the recovered signal $\hat{x}[n]$ as the input.

Figure 6 shows the output of the Pan-Tompkins algorithm when applied to the recovered signal $\hat{x}[n]$: all the 64 R peaks in the signal are properly detected. Indeed, these peaks are always within 1 sample (i.e., 2.8 ms) of the true location and their average distance is 0.3281 samples (i.e., 0.9 ms). Therefore, we conclude that the proposed approach is doing a good job in extracting the relevant information (i.e., locating all the R peaks) from the noisy signal.

V. CONCLUSIONS

In this paper, we have described an efficient algorithm to recover a sparse latent signal that requires the enforcement of a minimum distance among the activations. A cost function that includes both the sparsity and the minimum distance restrictions has been introduced and we have shown that this function can be minimized through an iterative reweighted LASSO approach. The proposed method has been tested on a first example with randomly generated samples and a second example with ECG signals with excellent results. Future lines include developing a method to obtain the optimum value of the sparsity parameter, performing extensive tests on real-

world signals (e.g., ECG and gamma spectrometry signals), and developing a multi-channel extension.

REFERENCES

- [1] S. Monzón, T. Trigano, D. Luengo, and A. Artés-Rodríguez, "Sparse spectral analysis of atrial fibrillation electrograms," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*. IEEE, 2012, pp. 1–6.
- [2] D. Luengo, S. Monzón, T. Trigano, J. Vía, and A. Artés-Rodríguez, "Blind analysis of atrial fibrillation electrograms: a sparsity-aware formulation," *Integrated Computer-Aided Engineering*, vol. 22, no. 1, pp. 71–85, 2015.
- [3] T. Trigano, Y. Sepulcre, M. Roitman, and U. Aferiat, "On nonhomogeneous activity estimation in gamma spectrometry using sparse signal representation," in *2011 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2011, pp. 649–652.
- [4] Y. Sepulcre, T. Trigano, and Y. Ritov, "Sparse regression algorithm for activity estimation in γ spectrometry," *IEEE Transactions on Signal Processing*, vol. 61, no. 17, pp. 4347–4359, 2013.
- [5] J. Jalife, M. Delmar, J. Anumonwo, O. Berenfeld, and J. Kalifa, *Basic Cardiac Electrophysiology for the Clinician*, 2nd ed. Wiley, 2011.
- [6] G. F. Knoll, *Radiation Detection and Measurement*. Wiley, 2000.
- [7] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-3, no. 6, pp. 610–621, nov. 1973.
- [8] M. J. Choi, J. Lim, A. Torralba, and A. Willsky, "Exploiting hierarchical context on a large database of object categories," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 129–136.
- [9] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267–288, 1996.
- [10] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused LASSO," *Journal of the Royal Statistical Society Series B*, vol. 67, no. 1, pp. 91–108, 2005.
- [11] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B*, vol. 67, no. 2, pp. 301–320, Jan. 2005.
- [12] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, Apr. 2010.
- [13] D. Luengo, J. Vía, S. Monzón, T. Trigano, and A. Artés-Rodríguez, "Cross-products LASSO," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6118–6122.
- [14] L. Sörnmo and P. Laguna, *Bioelectrical signal processing in cardiac and neurological applications*. Academic Press, 2005, vol. 8.
- [15] G. D. Clifford, F. Azuaje, and P. McSharry, *Advanced Methods and Tools for ECG Data Analysis*. Artech House, 2006.
- [16] P. E. McSharry, G. D. Clifford, L. Tarassenko, and L. A. Smith, "A dynamical model for generating synthetic electrocardiogram signals," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 3, pp. 289–294, 2003.
- [17] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, no. 3, pp. 230–236, 1985.