ENHANCING SMOOTHNESS IN AMPLITUDE MODULATED SPARSE SIGNALS

Stefan Ingi Adalbjörnsson, Johan Swärd, and Andreas Jakobsson

Centre for Mathematical Sciences, Lund University, Sweden.
email: {sia, js, aj}@maths.lth.se

ABSTRACT

In many applications, one desires to track the evolution of a sparse signal over time, space, or another appropriate dimension. Such signals may often be assumed to be locally stationary, such that the signal statistics does not vary significantly over shorter intervals, allowing the signal to be segmented over such intervals, which may then be modeled using a sparse reconstruction framework. The limited intervals necessarily implies lower resolution, higher variance, and performance loss in transition intervals, where the signal content varies. In this work, we introduce a sparse reconstruction framework restricting the smoothness and sparsity of the activated dictionary elements in the evolution along the processed (stationary) intervals. Numerical simulations illustrate the achieved performance gain.

***Index Terms*—** Sparse modeling, Non-stationary signals, Amplitude modulated frequencies, Convex modeling, ADMM.

1. INTRODUCTION

During the recent decade, there has been an explosive increase in the use and interest of sparse models in areas such as signal processing, statistics, machine learning, and economics (see, e.g., [1, 2]). In many of these areas, sparse modeling has been shown to be applicable to problems known to be notoriously difficult due to a difficult combination of an high dimensional multi-modal cost functions combined with a difficult model choice problem, with typical examples including spectral analysis, time-frequency analysis, and source separation [3–7]. Often, one can in these form of problems assume that the signal is stationary over some short interval, with typical examples being the modeling of audio signals or the tracking of slowly moving objects. In such application, the length of the data-window used for processing is typically determined by the desired temporal or spatial resolution, and by the variability of the signal. As an example, one may consider the short-time Fourier transform (STFT), where the signal is decomposed into a number of possibly overlapping windows, and the frequency estimate at each time-point is given as the discrete-time Fourier (DFT) transform of the window centered at that particular time-point, with the resolution thus being roughly inversely proportional to the number of samples in the window [8]. In this case, the resulting spectral estimates at each time-point are only related to each other through the amount of overlap between the used data windows. Imposing sparsity, many works consider the processing of only the considered data window, without imposing requirements of smoothness of the spectral or spatial content as the signal evolves over the considered dimension (see, e.g., [3, 4, 7, 9–11]). Recent efforts to estimate such signal have focused on creative recursive, or online, estimators that can create smooth estimate by using overlapping and/or short time window, e.g., such as [12, 13].

In this work, we attempt an alternative approach, by explicitly enforcing a smoothness in the estimated parameters in all the considered windows simultaneously, such that we strive to model a larger part of the signal, such as, for instance, a single utterance if modeling a speech signal. A similar approach was suggested in [14].

This work was supported in part by the Swedish Research Council, Carl Trygger's foundation, and the Royal Physiographic Society in Lund.

2. SPARSE MODELING WITH NON-STATIONARY DICTIONARY AMPLITUDES

To illustrate the proposed sparsely evolving reconstruction framework (SERF), we assume a signal $\mathbf{z} \in \mathbb{C}^N$ created from a sparse dictionary $\mathbf{A} \in \mathbb{C}^{N \times P}$, often constructed such that $P \gg N$, with slowly varying amplitudes evolving over the considered dimension, hereafter, for simplicity, assumed to be time, i.e., such that at time point $k \in \{1, \dots N\}$, the signal may be well modeled as $z_k = \mathbf{A}_{k:}\breve{\mathbf{X}}_{:k} + e_k$, where $z_k \in \mathbb{C}$ is the kth element of the vector $\mathbf{z}$, $\breve{\mathbf{X}}_{:k} \in \mathbb{C}^{P \times 1}$ a (column) vector of sparse amplitudes assumed to be slowly varying over time, $\mathbf{A}_{k:} \in \mathbb{C}^{1 \times P}$ the kth row of the dictionary matrix $\mathbf{A}$, and $e_k \in \mathbb{C}$ an additive noise term. The dictionary $\mathbf{A}$ can here, for example, be a set of sinusoids for a range of considered frequencies, as assumed in, e.g., [7], with $\breve{\mathbf{X}}_{:k}$ representing the complex amplitudes for those frequencies corresponding to sinusoidal components. Other dictionary alternatives include, for example, representations of blocks of frequencies [10], chirps [15], or directions of arrivals [3].

If no further assumptions are made on the amplitude vector, the number of unknown parameters in the model is PN. However, for many signals, it is reasonable to make an as-

	True positives	False positives
Spectrogram	0.90	996.43
Lasso	0.33	131.31
SERF	0.79	137.14

Table 1. Measured number of true and false positives.

sumption of smoothness of the dictionary amplitudes over time. To model this, we form a set of N_{win} data windows, $\mathbf{y}_n$, for $n = 1, \ldots, N_{\text{win}}$, consisting of shorter subsets of the samples in $\mathbf{z}$, such that the amplitude evolution over each data window may be assumed to be reasonably stationary. For notational simplicity, we will here assume that all such windows have the same length, M, although noting that this is no restriction of the presented algorithm. Thus,

$$\mathbf{y}_n = \mathbf{A}_n \mathbf{X}_{:n} + \mathbf{e}_n \tag{1}$$

with $\mathbf{A}_n \in \mathbb{C}^{M \times P}$ denoting the subset of rows of $\mathbf{A}$ used to form the data window, and $\mathbf{X}_{:n} \in \mathbb{C}^{P \times 1}$ the amplitude vectors for the considered data window[1]. This implies that the sought solution to the estimation problem in question may be expressed as

$$\begin{aligned} \min_{\mathbf{X}} \quad & \sum_{n=1}^{N_{\text{win}}} \|\mathbf{y}_n - \mathbf{A}_n \mathbf{X}_{:n}\|_2^2 \\ \text{subject to} \quad & \begin{cases} \mathcal{G}(\mathbf{X}_{:n}) \leq C & \text{for } n = 1, \ldots, N_{\text{win}} \\ \mathcal{H}(\mathbf{X}_{p:}) \leq L & \text{for } p = 1, \ldots, P \end{cases} \end{aligned} \tag{2}$$

where

$$\mathbf{X} = \left[\begin{array}{ccc} \mathbf{X}_1 & \ldots & \mathbf{X}_{N_{\text{win}}} \end{array} \right] \tag{3}$$

and $\mathbf{X}_{p:}$ denotes the pth row of the matrix $\mathbf{X}$. Furthermore, $\mathcal{G}(\cdot)$, denoting a function that measures the sparsity of a vector, is added to enforce sparsity at each time-point, and $\mathcal{H}(\cdot)$ is an appropriate operator that measure the smoothness along time. For tractability, we consider a convex modeling approach, using the standard ℓ_1 norm to enforce spareness, and a finite differences approximation of the second derivative to enforce smoothness in the time-domain, implying that the minimization may be expressed as

$$\begin{aligned} \min_{\mathbf{X}} \quad & \sum_{n=1}^{N_{\text{win}}} \|\mathbf{y}_n - \mathbf{A}_n \mathbf{X}_{:n}\|_2^2 \\ \text{subject to} \quad & \begin{cases} \|\mathbf{X}_{:n}\|_1 \leq C & \text{for } n = 1, \ldots, N_{\text{win}} \\ \|\mathbf{H}\mathbf{X}_{p:}\|_1 \leq L & \text{for } p = 1, \ldots, P \end{cases} \end{aligned} \tag{4}$$

with the matrix $\mathbf{H}$ selected to enforce smoothness, e.g., by using a finite difference approximation of the first or second derivative (see also the discussion below).

[1] The vectors $\mathbf{X}_{:n}$, for $n = 1, \ldots, N_{\text{win}}$, may thus be seen as (approximately being) a subset of the vectors $\tilde{\mathbf{X}}_{:k}$, for $k = 1, \ldots, N$.

Algorithm 1 The SERF algorithm

1: Let $i = 1$ and $\mathbf{Z}^{(0)} = \mathbf{X}^{(0)} = \mathbf{D}^{(0)} = \mathbf{0}$.
2: **repeat**
3: **for** each window, $n = 1, \ldots, N_{\text{win}}$ **do**
4: Solve (11) yielding a sparse $\mathbf{X}_{:n}$
5: **end for**
6: **for** each dictionary element, $p = 1, \ldots, P$ **do**
7: Solve (14) yielding a smooth $\mathbf{Z}_{p:}$
8: **end for**
9: Update $\mathbf{D}$ as (15) and set $i = i + 1$
10: **until** Convergence or $i = I_{\max}$

The resulting minimization enforces smoothness between the parameters for each the considered data-windows, while simultaneously imposing sparsity in each window, with the user parameters C and L controlling the sparsity and smoothness, respectively. Generally, as it might be non-intuitive to select the parameters C and L appropriately, we here reformulate the minimization in (4) as

$$\min_{\mathbf{X}} \quad f(\mathbf{X}) + g(\mathbf{X}) \tag{5}$$

where

$$f(\mathbf{X}) = \sum_{n=1}^{N_{win}} \|\mathbf{y}_n - \mathbf{A}_n \mathbf{X}_{:n}\|_2^2 + \lambda_1 \|\mathbf{X}\|_1 \tag{6}$$

$$g(\mathbf{X}) = \lambda_2 \sum_{p=1}^{P} \|\mathbf{H}\mathbf{X}_{p:}\|_1 \tag{7}$$

Here, λ_1 and λ_2 are user parameters indicating the relative weighting between the different cost functions, corresponding to the relative importance of the penalties given by the weights C and L in (4).

Thus, the effects of the smoothness constraint can be intuitively understood by considering the Karush-Kuhn-Tucker conditions; for example, if we consider a simple situation with 2 time windows, where all the elements $\mathbf{X}_{:1}^*$ are zero at their optima, it can be seen that the sparsity is enhanced in $\mathbf{X}_{:2}^*$, such that the first non-zero parameter is not simply the least squares estimate shrunk by λ_1, as one get with only an ℓ_1 penalty, but in fact shrunk by $\lambda_1 + \lambda_2$. If, on the other hand, $\mathbf{X}_{:1}^*$ is non-zero at their optima, a similar analysis shows that the sparsifying effect is instead reduced. Thus, the overall effect is that, if a dictionary element would have become activated if no smoothness condition was imposed, then it can be seen as reducing the penalty on all adjacent windows, decreasing the sparsity for that specific dictionary element, which would help reducing the bias usually associated with ℓ_1 penalties for the windows where the amplitude of the dictionary element is transitioning to and from being zero.

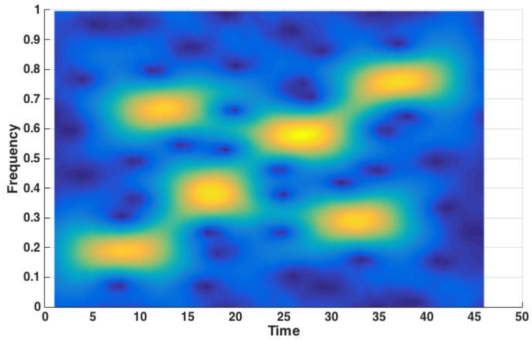

Fig. 1. Spectrogram of a time-evolving signal.

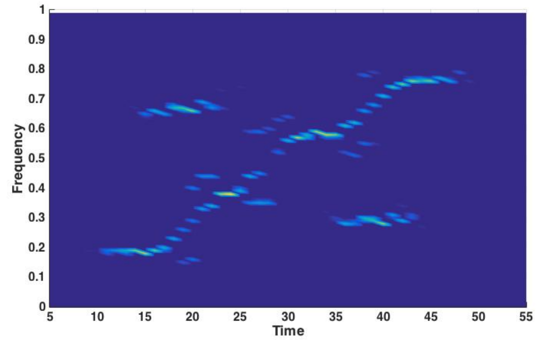

Fig. 2. The Lasso estimate of a time-evolving signal.

3. EFFICIENT IMPLEMENTATION

In order to form a computationally attractive minimizer to (5), we utilize the ADMM framework (see, e.g., [16]), which enables a solver requiring only $\mathcal{O}(N^2P^2)$ operations per iteration ($N < P$), as compared to a $\mathcal{O}(N^3P^3)$ as would be required by a standard interior point based solver. To do so, reformulate (5) as

$$\min_{\mathbf{X},\mathbf{Z}} \quad f(\mathbf{X}) + g(\mathbf{Z}) \quad \text{subject to} \quad \mathbf{X} = \mathbf{Z} \tag{8}$$

With this notation, the augmented Lagrangian of the considered problem may be formed as

$$L_p(\mathbf{X},\mathbf{Z},\mathbf{D}) = f(\mathbf{X}) + g(\mathbf{Z}) + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Z} + \mathbf{D}\|_F^2 \tag{9}$$

where $\mathbf{D}$ is the scaled dual variable and ρ is the step length of the algorithm. The basic idea behind ADMM is to iteratively solve (9), for each variable, until convergence. Thus, for $\mathbf{X}$, and for each window n, one needs to find the nth column of $\mathbf{X}$, say $\mathbf{x}$, such that

$$\min_{\mathbf{x}} \|\mathbf{y}_n - \mathbf{A}_n\mathbf{x}\|_2^2 + \lambda_1\|\mathbf{x}\|_1 + \frac{\rho}{2}\|\mathbf{x} - \mathbf{Z}_{:n} + \mathbf{D}_{:n}\|_2^2 \tag{10}$$

Reformulating (10) yields the following $\mathbf{x}$-step

$$\mathbf{X}_{:n}^{(i)} = \arg\min_{\mathbf{x}} \|\tilde{\mathbf{y}}_n^{(i-1)} - \tilde{\mathbf{A}}_n\mathbf{x}\|_2^2 + \lambda_1\|\mathbf{x}\|_1 \tag{11}$$

where

$$\tilde{\mathbf{y}}_n^{(i-1)} = \begin{bmatrix} \mathbf{y}_n^T & (\mathbf{D}_{:n}^{(i-1)} - \mathbf{Z}_{:n}^{(i-1)})^T \end{bmatrix}^T \tag{12}$$

$$\tilde{\mathbf{A}}_n = \begin{bmatrix} \mathbf{A}_n^T & \mathbf{I}_p \end{bmatrix}^T \tag{13}$$

with $\mathbf{I}_p$ denoting the $p \times p$ identity matrix, $(\cdot)^T$ the transpose, and the superscript i the iteration. The optimization problem in (11) may then be solved using, e.g., an interior-point solver or again using the ADMM. The step for the $\mathbf{Z}$ variable is

$$\mathbf{Z}^{(i)} = \arg\min_{\mathbf{Z}} \lambda_2 \sum_{p=1}^{P} \|\mathbf{H}\mathbf{Z}_{p:}\|_1 + \frac{\rho}{2}\|\mathbf{X}_{p:}^{(i)} - \mathbf{Z}_{p:} + \mathbf{D}_{p:}^{(i-1)}\|_2^2 \tag{14}$$

which clearly separates into p smaller optimization problems, and which may, depending on the structure of the smoothness matrix $\mathbf{H}$, be solved very efficiently. If, for example, $\mathbf{H}$ is a first order difference matrix, i.e., a matrix with ones on the diagonal and -1 on the superdiagonal, as would occur for a signal where one want to restrict an approximation of the first derivative, one may use the method described in [17], solving (14) in $\mathcal{O}(PN_{\text{win}})$, or using method described in [18]. If, instead, one wishes to restrict an approximation of the second derivative, $\mathbf{H}$ can be set as a second order difference matrix, i.e., constructed to have ones on the super- and subdiagonals, and with the diagonal having -2, one may solve (14) using, e.g., the ADMM [19].

The final step in the ADMM algorithm required to solve (8) is to update the dual variable

$$\mathbf{D}^{(i)} = \mathbf{X}^{(i)} - \mathbf{Z}^{(i)} + \mathbf{D}^{(i-1)} \tag{15}$$

The resulting SERF algorithm is summarized in Algorithm 1. It may be noted that instead of using a fixed maximum number of iterations, one may use a variety of alternative termination criteria (see, e.g., [16] for a detailed discussion on such alternatives). Note that, in step 4, we have here assumed that a ℓ_1 penalized least square problem is solved, however, other sparsity constraint such as block sparsity [20], adaptive Lasso, Fused Lasso, or even, non-convex sparsity promoting penalties such as ℓ_q [21], could be used instead depending on the structure of the dictionary matrix.

4. NUMERICAL EXAMPLES

We proceed to examine the performance of the proposed algorithm. We consider a $N = 60$ samples long signal consisting of several unit amplitude sinusoids, each only being active for a short time span. Without making any assumption of knowing the length of the duration of each sinusoid, we use using windows of length $M = 10$ that are fully overlapping for but one sample, this segmentation will ensure that many windows will contain the signal of several sinusoids, as well as onsets and endings of the various tones. Figures 1 and 2 show a typical realization of the spectral estimates resulting

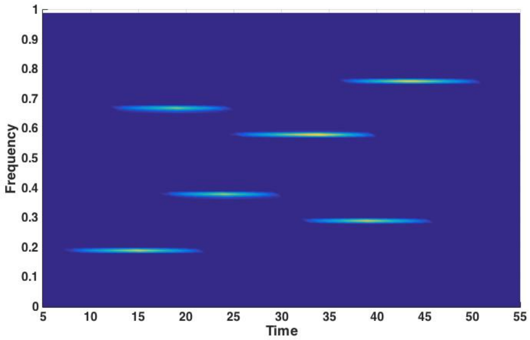

Fig. 3. The SERF estimate of a time-evolving signal.

from using the spectrogram (using a zero-padding of 100 grid points) and the Lasso. As is well known, the spectrogram will have a poor resolution in both time and frequency due to the short windows and the short duration of each tone. The Lasso estimate is also seen to have notable problems to capture the quickly changing nature of the signal. Figure 3 shows the corresponding result for the proposed algorithm, clearly illustrating how the added smoothness constraint allows the different windows to use the contributions from the other windows constructively. In this case, the estimate closely follows the ground truth, without the blurring problems the other methods suffer from.

Finally, using the above tuning parameters, we examine the consistency of the presented results, examining the number of true and false positive detections, obtained using 100 Monte-Carlo simulations. This was done by truncating all values below 0.5 of the normalized spectrogram, comparing the activated pixels with the reference signalThe false positives were measured as the number of activated indices that are not in the reference signal, whereas the true positives measure the ratio between the number of correctly activated pixels and the cardinality of the ground truth index set. Table 1 shows the resulting measures. As the Lasso estimate is too sparse, it has few false positives, but also fail to correctly find the signal, whereas the spectrogram, activating too many pixels shows the opposite behavior. The SERF estimator, has balanced score, indicating its preferable performance.

5. REFERENCES

[1] M. Elad, *Sparse and Redundant Representations*, Springer, 2010.

[2] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, April 2004.

[3] D. Malioutov, M. Cetin, and A. S. Willsky, "A Sparse Signal Reconstruction Perspective for Source Localization With Sensor Arrays," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3010–3022, August 2005.

[4] X. Tan, W. Roberts, J. Li, and P. Stoica, "Sparse Learning via Iterative Minimization With Application to MIMO Radar Imaging," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 1088–1101, March 2011.

[5] S. I. Adalbjörnsson, T. Kronvall, S. Burgess, K. Åström, and A. Jakobsson, "Sparse Localization of Harmonic Audio Sources," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 117–129, Jan. 2016.

[6] P. Stoica, P. Babu, and J. Li, "SPICE : a novel covariance-based sparse estimation method for array processing," *IEEE Trans. Signal Process.*, vol. 59, no. 2, pp. 629 –638, Feb. 2011.

[7] J. J. Fuchs, "On the Use of Sparse Representations in the Identification of Line Spectra," in *17th World Congress IFAC*, Seoul, jul 2008, pp. 10225–10229.

[8] P. Stoica and R. Moses, *Spectral Analysis of Signals*, Prentice Hall, Upper Saddle River, N.J., 2005.

[9] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 101–111, jan. 2003.

[10] S. I. Adalbjörnsson, A. Jakobsson, and M. G. Christensen, "Multi-Pitch Estimation Exploiting Block Sparsity," *Elsevier Signal Processing*, vol. 109, pp. 236–247, April 2015.

[11] M. D. Plumbley, S. A. Abdallah, T. Blumensath, and M. E. Davies, "Sparse representations of polyphonic music," *Signal Processing*, vol. 86, no. 3, pp. 417–431, March 2006.

[12] D. Zachariah and P. Stoica, "Online Hyperparameter-Free Sparse Estimation Method," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3348–3359, July 2015.

[13] B. Dumitrescu, A. Onose, P. Helin, and I. Tăbuş, "Greedy Sparse RLS," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2194–2207, May 2012.

[14] C. Kereliuk and P. Depalle, "Sparse and structured audio decomposition based on time-frequence adjacency," `http://www.music.mcgill.ca/ corey/ieeetsalp2013/SparseAndStructuredAudioAdjacency.pdf`.

[15] J. Swärd, J. Brynolfsson, A. Jakobsson, and M. Hansson-Sandsten, "Sparse Semi-Parametric Estimation of Harmonic Chirp Signals," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1798–1807, April 2016.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[17] N. A. Johnson, "A Dynamic Programming Algorithm for the Fused Lasso and ℓ_0-Segmentation," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 246–260, 2013.

[18] B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang, "An ADMM Algorithm for a Class of Total Variation Regularized Estimation Problems," in *Proceedings 16th IFAC Symposium on System Identification,*, Michel Kinnaert, Ed., July 2012, vol. 16.

[19] S. J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 Trend Filtering," *SIAM Review*, vol. 51, no. 2, pp. 339–360, 2009.

[20] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A Sparse-Group Lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[21] R. Chartrand and V. Staneva, "Restricted Isometry Properties and Nonconvex Compressive Sensing," *Inverse Problems*, vol. 24, no. 035020, pp. 1–14, 2008.