Abstract— In the context of gridless sparse optimization, the Sliding Frank-Wolfe algorithm recently introduced has shown interesting analytical and practical properties. Nevertheless, it is application to large data, such as in the case of 3D deconvolution, is computationally heavy. In this paper, we investigate a strategy for leveraging this burden, in order to make this method more tractable for 3D deconvolution. We show that a boosted SFW can achieve the same results in a significantly reduced amount of time.

1 Introduction

1.1 Observation model

In this paper, we consider the problem of 3D deconvolution of some volume \( y \) containing a small number of atoms. We assume that \( y \) results from some measure \( \mu_{w, \theta} \) observed through an imaging operator \( \Phi \):

\[
y = \Phi \mu_{w, \theta} + \epsilon
\]  

Here and in the following, we consider for the measures \( \mu_{w, \theta} \) a positive weighted Dirac mass sum of the form \( \sum_{n=1}^{N} w_n \delta_{\theta_n} \), with the weight vector \( w = \{w_1, \ldots, w_N\} \in \mathbb{R}_{+}^{N} \). \( \theta_n \) locates, for each atom, its parameters within the bounded domain \( D \). Without loss of generality, we choose to use generalized isotropic Gaussians as atoms, such that \( \forall 1 \leq n \leq N \) and \( \forall s \in \mathbb{R}^3 \):

\[
G(\theta_n, w_n; s) = w_n \exp \left( -\frac{1}{2 \sigma_{\theta_n}^2} \|m_n - s\|_2^2 \right)
\]

so that \( \theta_n = \{m_n, \sigma_n, d_n\} \in D \subset \mathbb{R} \times \mathbb{R} \times \mathbb{R}^2 \).

Furthermore, \( \Phi \) embeds a point spread function (PSF) \( H \) that blurs the observations of atoms. Rephrasing (1), we have:

\[
y = H * \sum_{n=1}^{N} G(\theta_n, w_n) + \epsilon
\]

The problem handled here consists in estimating \( N \), and \( \{\theta_n, w_n\}_{n=1}^{N} \) while knowing only \( y \) and \( H \).

1.2 Continuous sparsity and the SFW algorithm

The problem handled here is stated in a sparse continuous fashion: there are a few atoms to search for, and they do not lie in a pre-established grid or dictionary. Hence, it is desirable to work in a fully continuous setting \([3][4][5]\), replacing the sparsity-promoting \( \ell_1 \) norm with its continuous counterpart, the total variation of measures.

To solve the problem, coined BLASSO by \([1]\), several approaches exist. In \([3][8]\), the problem is recast as a semi-definite program, whereas the ADCG solver proposed in \([2][3]\) relies on an alternating gradient based method which progressively adds Dirac masses. Recently, a variant of the ADCG called Sliding Frank-Wolfe (SFW) appeared in \([6]\), which is guaranteed to converge in a finite number of steps under suitable assumptions.

In this paper, we focus on the SFW algorithm and its application to the inversion of \([3]\). To retrieve \( \mu_{w, \theta} \) in a sparse fashion, we minimize among all non-negative measures \( \mu \) the following criterion:

\[
C(y, \mu_{w, \theta}, \lambda) = \frac{1}{2} \|y - \Phi \mu_{w, \theta}\|_2^2 + \lambda \mu_{w, \theta}(D)
\]

with \( \lambda > 0 \) the regularization parameter, and \( \mu(D) \) denotes the total mass of the non-negative measure \( \mu \). In the case of a sum of Dirac masses, \( \mu_{w, \theta}(D) = \sum_{n=1}^{N} w_n \).

SFW is a greedy algorithm, which adds iteratively Dirac masses to the estimated measure \( \mu \). To do so, SFW is ruled by a certificate on the decrease of the \( \mu_{w, \theta}(D) \).

Informally, it can be seen as the result of the convolution between the residual and the volume of an atom located at one point in \( D \). Algorithm \([4]\) depicts SFW, and further details on this algorithm can be found in \([6]\).

2 Boosting the SFW solver

When applying the SFW solver to invert \([3]\), several observations can be made:

- in most cases SFW is efficient and retrieves the \( K \) atoms provided in exactly \( K \) steps. This has already been noted in \([6]\), in which the convergence in a finite number of steps has been proven under mild assumptions.
• most computation time is spent in the local descents of step 4, which handle all \(k \times 6\) parameters (6 being, in our case, the dimension of \(D\)). This seems computationally wasteful: in all iterations except the last, the fine-tuned parameters are modified afterwards.

• this step only marginally decreases \(C\) as defined in (4).

We noted that this is related to the precision of the result of step 1: when the local ascent maximizing \(\eta\) yields a relevant result, the local descent of step 4 does not decrease \(C\).

Based on these observations, we propose a boosted version of the SFW algorithm, denoted BSFW, which removes all but the last local descent. As in SFW, the certificate \(\eta\) locates new atoms and indicates when the algorithm should stop. Then, at iteration \(k\), either:

\[
\max_{\theta \in D} \eta(k) > 1, \quad \text{so a new Dirac mass is added, and \(w\) is adjusted,}
\]

\[
\max_{\theta \in D} \eta(k) < 1, \quad \text{so a local descent is made to adjust \((w, \theta)\), and then BSFW stops.}
\]

The procedure is summarized in Algorithm 2.

Algorithm 2 Boosted Sliding Frank-Wolfe

**Input:** \(y, \text{ PSF } H, \lambda\)

**Output:** Estimated minimizer \(\hat{\mu}_{w, \theta}\) of (4)

Initialization: \(\mu_{w(0), \theta(0)} = 0\)

Repeat:

1. Compute \(\max_{\theta \in D} \eta(k)\) by local ascent using e.g. BFGS, starting from a maximum attained on a grid.

   if \(\max_{\theta \in D} \eta(k) > 1:\)

   2. Augment the support: \(\theta'(k) = \theta(k-1) \cup \arg \max_{\theta \in D} \eta(k)\)

   3. Adjust weights only (LASSO):

      \[
      \tilde{w}(k) = \arg \min_{w \in \mathbb{R}^+} C(y, \mu_{w(k), \theta(k)}, \lambda)
      \]

   4. Remove zero-weighted masses, update the measure:

      \[
      \mu_{w(k), \theta(k)} = \sum_{n=1}^{k} \tilde{w}(n) \delta_{\theta(n)}
      \]

   else:

   2b. Local descent on all parameters using e.g. BFGS, starting at \(\mu_{w(k), \theta(k)}:\)

      \[
      w(k), \theta(k) = \text{local descent of } C(y, \mu_{w(k), \theta(k)}, \lambda)
      \]

   3b. Remove zero-weighted masses, update the measure:

      \[
      \mu_{w(k), \theta(k)} = \sum_{n=1}^{k} w_n(k) \delta_{\theta(n)}
      \]

end of BSFW

Note that steps 1–4 of the BSFW algorithm form a Frank-Wolfe algorithm, so BSFW has similar convergences properties. However, the BSFW does not benefit from the nicer convergence properties of SFW (explained in [6]), because local descents were removed.

In addition to the new algorithm presented above, several implementation points have been leveraged to improve the speed of both SFW and BSFW:

• convolutions are made in Fourier domain,

• \(\nabla C\) is computed analytically, so as to avoid a costly numerical approximation,

• the computation of \(C\) and \(\nabla C\) is parallelized across atoms,

• a lookup table for convolution in \(D\) is computed once in order to help the computations of \(C\) at step 1.

### 3 Numerical results

In this section, we compare the numerical behavior of SFW and BSFW. We use a PSF \(H\) appearing in tomographic diffractive microscopy [7], which significantly blur along the optical axis. Figure 1 exemplifies the volume handled in our problem.

When inverting 3 with SFW and BSFW, we are mainly interested in (a) the computation time and (b), the value of \(C\), in order to measure the loss of precision (or no) from SFW to BSFW.

Both algorithms were studied while varying the size of the considered volumes (from 60³ to 100³ voxels) and the number of Dirac mass to retrieve (4, 6 and 8) using a fixed \(\lambda = 0.2\) and a Gaussian white noise. Each experiment was repeated 30 times with different values of the real \(\mu^*\), and Fig. 2 presents the averaged results.

These results can be summarized as follows:

• BSFW attains almost exactly the same criterion value as SFW in all the studied cases.

• BSFW improves notably the computation times: on average it runs 31% faster than FSW for the same task.

However the improvements are not as important as expected from the removal of almost all the costly local descents. After inspecting the results, we observed that in some cases, repeating steps 1–4 yields more Dirac masses then necessary, which are costly to manage within the local descents of step 2b. These additional masses are only removed at step 3b, so the final results does yield the adequate number of atoms.

### 4 Discussion

We studied the potential speed improvements of the SFW algorithm, by removing costly intermediate local descent steps. We showed that this alternative does run significantly faster than SFW, at the cost of following a non-optimal optimization path and thus preventing BSFW to run even faster.

Several perspectives are considered as future works, among which the improvement of the optimization path within BSFW, and the joint search for an adequate regularization parameter.
References

[1] Jean-Marc Azais, Yohann De Castro, and Fabrice Gamboa. Spike detection from inaccurate samplings. *Applied and Computational Harmonic Analysis*, 38(2):177–195, 2015.

[2] Nicholas Boyd, Geoffrey Schiebinger, and Benjamin Recht. The alternating descent conditional gradient method for sparse inverse problems. *SIAM Journal on Optimization*, 27(2):616–639, 2017.

[3] Kristian Bredies and Hanna Katriina Pikkarainen. Inverse problems in spaces of measures. *ESAIM: Control, Optimisation and Calculus of Variations*, 19(1):190–218, 2013.

[4] Emmanuel J Candès and Carlos Fernandez-Granda. Towards a mathematical theory of super-resolution. *Communications on Pure and Applied Mathematics*, 67(6):906–956, 2014.

[5] Yohann De Castro and Fabrice Gamboa. Exact reconstruction using Beurling minimal extrapolation. *Journal of Mathematical Analysis and Applications*, 395(1):336–354, 2012.

[6] Quentin Denoeye, Vincent Duval, Gabriel Peyré, and Emmanuel Soubies. The sliding frank–wolfe algorithm and its application to super-resolution microscopy. *Inverse Problems*, 36(1):014001, 2019.

[7] Bertrand Simon and Olivier Haeberlé. Tomographic diffractive microscopy: Principles, implementations, and applications in biology. In *Label-Free Super-Resolution Microscopy*, pages 85–112. Springer, 2019.

[8] Gongguo Tang, Badri Narayan Bhaskar, Parikshit Shah, and Benjamin Recht. Compressed sensing off the grid. *IEEE transactions on information theory*, 59(11):7465–7490, 2013.