A DYNAMIC SCREENING PRINCIPLE FOR THE LASSO

Antoine Bonnefoy[⋆], *Valentin Emiya*[⋆], *Liva Ralaivola*[⋆], *Rémi Gribonval*[∘]

[⋆] Aix-Marseille Université, CNRS UMR 7279 LIF
[∘]Inria

ABSTRACT

The *Lasso* is an optimization problem devoted to finding a *sparse representation* of some signal with respect to a predefined dictionary. An original and computationally-efficient method is proposed here to solve this problem, based on a *dynamic screening principle*. It makes it possible to accelerate a large class of optimization algorithms by iteratively reducing the size of the dictionary during the optimization process, discarding elements that are provably known not to belong to the solution of the *Lasso*. The iterative reduction of the dictionary is what we call *dynamic screening*. As this screening step is inexpensive, the computational cost of the algorithm using our dynamic screening strategy is lower than that of the base algorithm. Numerical experiments on synthetic and real data support the relevance of this approach.

Index Terms— Screening test, Dynamic screening, *Lasso*, First-order algorithms, ISTA.

1 Introduction

The *Lasso* [9] is an optimization problem that aims at finding a sparse solution to a least square problem by minimizing the sum of an ℓ_2-fitting term and an ℓ_1-regularization term. Given some observation/signal $\mathbf{y} \in \mathbb{R}^N$ and a dictionary matrix $\mathbf{D} \in \mathbb{R}^{N\times K}$ with $N \leq K$, this problem writes

$$\mathcal{P}(\lambda, \mathbf{D}, \mathbf{y}) : \tilde{\mathbf{x}} \triangleq \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1, \qquad (1)$$

where the parameter $\lambda > 0$ governs the sparsity of $\tilde{\mathbf{x}}$. We would like to be able to handle (1) when both N and K may be large, which occurs in many practical applications resorting to the *Lasso*: denoising, inpainting or classification. Algorithms relying on first-order information (*e.g.* gradient) only are particularly suited for these problems, as second-order based methods (*e.g.* using the Hessian) imply too computationally demanding iterations. These first-order algorithms include primal [1, 4, 6, 12] and primal-dual [3, 8] algorithms.

Accelerating these algorithms is yet a key challenge: even though they provably have fast convergence, they remain captive of the dictionary size due to the required multiplications by $\mathbf{D}$ and $\mathbf{D}^T$ over the optimization process. To overcome this limitation, strategies based on *screening tests* [5, 7, 10, 11, 14, 13] have recently been proposed. They implement two steps: i) locate zeros of $\tilde{\mathbf{x}}$ thanks to a *screening test* and construct the reduced or *screened* dictionary $\mathbf{D}_0$ which is dictionary $\mathbf{D}$ trimmed off of its columns that correspond to the located zeros and ii) solve $\mathcal{P}(\lambda, \mathbf{D}_0, \mathbf{y})$ (see Algorithm 1).

We propose a new screening principle called *dynamic screening* in order to even more reduce the computational cost of first-order algorithms. We take the aforementioned concept of *screening test* one step further, and improve existing screening tests by embedding them in the iterations of first-order algorithms. We take advantage of the computation made during the optimization procedure to perform a *screening* at each iteration with a negligible computational overhead, and we consequently *dynamically and iteratively* reduce the size of $\mathbf{D}$. To our knowledge, this is the first time such a screening mechanism is envisioned. Opposing perspectives of the proposed *dynamic* screening and existing *static* screening are schematized in Algorithms 1 and 2.

Experiments show that the *dynamic screening* principle significantly reduces the computational cost of the optimization in a large range of λ values. The computational saving reaches up to 90% with respect to the base algorithm, or up to 70% with respect to the algorithm with static screening.

Algorithm 1
Static screening strategy

```
$\mathbf{D}_0 \leftarrow$ Screen $\mathbf{D}$
loop k
    $\mathbf{x}_{k+1} \leftarrow$Update $\mathbf{x}_k$ using $\mathbf{D}_0$
end loop
```

Algorithm 2
Dynamic screening strategy

```
$\mathbf{D}_0 \leftarrow \mathbf{D}$
loop k
    $\mathbf{x}_{k+1} \leftarrow$ Update $\mathbf{x}_k$ using $\mathbf{D}_k$
    $\mathbf{D}_{k+1} \leftarrow$ Screen $\mathbf{D}_k$ using $\mathbf{x}_{k+1}$
end loop
```

Section 2 introduces the tools we build our work upon. The dynamic screening principle is then presented and analyzed in Section 3. Section 4 is devoted to numerical experiments. Finally we discuss several extensions that can emerge from this work in Section 5.

This work was supported by Agence Nationale de la Recherche (ANR), project GRETA 12-BS02-004-01.

Centre Inria Rennes-Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes, France. R.G. acknowledges funding by the European Re-search Council within the PLEASE project under grant ERC-StG-2011- 277906

2 Screening tests and algorithms

In this section, we set the notation, introduce previous works on screening tests for the *Lasso* and recall state-of-the-art algorithms to solve this problem, pointing out their computational limitations.

2.1 Notation

$\mathbf{D} \triangleq [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{N\times K}$ denotes a *dictionary* and $\Omega \triangleq \{1, \ldots, K\}$ denotes the set of integers indexing the columns, or atoms, of $\mathbf{D}$. The i-th component of $\mathbf{x}$ is denoted $\mathbf{x}(i)$. The observation $\mathbf{y} \in \mathbb{R}^N$ is assumed to have a sparse representation $\mathbf{x} \in \mathbb{R}^K$ in $\mathbf{D}$, *i.e.* $\|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2$ and $\|\mathbf{x}\|_0$ are small. Without loss of generality, observation $\mathbf{y}$ and atoms $\mathbf{d}_i$ are assumed to have unit ℓ_2 norm. The dual problem associated to (1) is [7, 14]:

$$\tilde{\boldsymbol{\theta}} \triangleq \arg\max_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{\lambda^2}{2}\left\|\boldsymbol{\theta} - \frac{\mathbf{y}}{\lambda}\right\|_2^2 \tag{2a}$$

$$\text{s.t. } \forall i \in \Omega, |\boldsymbol{\theta}^T\mathbf{d}_i| \le 1. \tag{2b}$$

A dual point $\boldsymbol{\theta}$ is said *feasible* if it complies with the constraints (2b). The solutions of the primal (1) and dual (2) problems, $\tilde{\mathbf{x}}$ and $\tilde{\boldsymbol{\theta}}$ respectively, are linked by the relation:

$$\mathbf{y} = \mathbf{D}\tilde{\mathbf{x}} + \lambda\tilde{\boldsymbol{\theta}}, \quad \forall i \in \Omega, \begin{cases} |\tilde{\boldsymbol{\theta}}^T\mathbf{d}_i| < 1 & \text{if } \tilde{\mathbf{x}}(i) = 0 \\ |\tilde{\boldsymbol{\theta}}^T\mathbf{d}_i| = 1 & \text{if } \tilde{\mathbf{x}}(i) \neq 0 \end{cases} \tag{3}$$

We additionally define: $\mathbf{d}_* = \arg\max_{\mathbf{d}\in\{\pm\mathbf{d}_i\}_{i=1}^K} \mathbf{d}^T\mathbf{y}$, and $\lambda_* = \mathbf{d}_*^T\mathbf{y}$. To avoid the null solution: $\lambda \in [0, \lambda_*[$.

2.2 Screening Tests

The sparsity inducing regularization $\lambda\|\cdot\|_1$ entails an optimum $\tilde{\mathbf{x}}$ that may contain many zeros, and the goal of a screening test is precisely to locate them; an efficient screening test locates many zeros. From the located zeros a *screened dictionary* $\mathbf{D}_0$ can be defined removing the corresponding atoms, called inactive atoms, from $\mathbf{D}$. Finally the solution of $\mathcal{P}(\lambda, \mathbf{D}_0, \mathbf{y})$ can be readily reconstructed from that of $\mathcal{P}(\lambda, \mathbf{D}, \mathbf{y})$. Any optimization procedure using the screened dictionary $\mathbf{D}_0$ therefore computes the solution of $\mathcal{P}(\lambda, \mathbf{D}, \mathbf{y})$ at lower computational cost.

Screening tests [7, 13, 14] are based on an idea emerging from the relation (3) between primal and dual optima, $\tilde{\mathbf{x}}$ and $\tilde{\boldsymbol{\theta}}$ respectively. According to relation (3), atoms $\mathbf{d}_i$ such that $|\tilde{\boldsymbol{\theta}}^T\mathbf{d}_i| < 1$ correspond to inactive atoms. If $\tilde{\boldsymbol{\theta}}$ were known it would be easy to identify inactive atoms. It is obviously not the case then to locate inactive atoms screening tests build an upper-bound on $|\tilde{\boldsymbol{\theta}}^T\mathbf{d}_i|$ by constructing a region $\mathcal{R} \subset \mathbb{R}^N$ that contains $\tilde{\boldsymbol{\theta}}$ and hence satisfies $|\tilde{\boldsymbol{\theta}}^T\mathbf{d}_i| < \max_{\boldsymbol{\theta}\in\mathcal{R}} |\boldsymbol{\theta}^T\mathbf{d}_i|$. It allows one to remove every atom $\mathbf{d}_i$ verifying $\max_{\boldsymbol{\theta}\in\mathcal{R}} |\boldsymbol{\theta}^T\mathbf{d}_i| < 1$.

Screening tests essentially differ from one another on the region $\mathcal{R}$ they consider, when $\mathcal{R}$ is a sphere [7, 14] $\max_{\boldsymbol{\theta}\in\mathcal{R}} |\boldsymbol{\theta}^T\mathbf{d}_i|$ has a closed-form expression and gives the sphere test principle. Spheres that instantiate this principle are described below.

Lemma 1 (Sphere Test Principle [7])**.** *If the solution* $\tilde{\boldsymbol{\theta}}$ *of* (2) *satisfies* $\exists\{r, \mathbf{c}\} \in \mathbb{R} \times \mathbb{R}^N$, $\|\tilde{\boldsymbol{\theta}} - \mathbf{c}\|_2 \le r$, *then :*

$$|\mathbf{c}^T\mathbf{d}_i| < 1 - r \Rightarrow \tilde{\mathbf{x}}(i) = 0.$$

In practice, once a sphere $\mathcal{S}(\mathbf{c}, r)$ of center $\mathbf{c}$ and radius r has been defined, every atom i such that $|\boldsymbol{\tau}(i)| < 1 - r$ are removed, where $\boldsymbol{\tau} = \mathbf{D}^T\mathbf{c}$. The associated screening operator $\Pi_{\mathbf{c},r}(\cdot)$ is the operator that, given a dictionary $\mathbf{D}$, outputs the corresponding *screened* dictionary

$$\Pi_{\mathbf{c},r}(\mathbf{D}) \triangleq \Big[\mathbf{d}_i \text{ s.t. } i \in \Omega, |\mathbf{c}^T\mathbf{d}_i| \ge 1 - r\Big]. \tag{4}$$

The construction of such spheres [7, 14] is based on the following considerations. The dual optimum $\tilde{\boldsymbol{\theta}}$ is closer to $\mathbf{y}/\lambda$ than any *feasible* point in $\mathbb{R}^N$. Then from any feasible dual point $\boldsymbol{\theta}$ one can construct a sphere, centered on $\mathbf{y}/\lambda$ with radius $\|\boldsymbol{\theta} - \mathbf{y}/\lambda\|_2$, that contains $\tilde{\boldsymbol{\theta}}$. The SAFE method [7, 14], implements the feasible dual point $\boldsymbol{\theta} = \mathbf{y}/\lambda_*$.

2.3 First-order algorithms for the *Lasso*

The *Lasso* problem (1) may be solved with general-purpose first-order algorithms such as ISTA [6], TwIST [2], FISTA [1], SpaRSA [12], forward-backward splitting [4] or the Chambolle and Pock's primal-dual algorithm [3, 8]. For the sake of simplicity, ISTA is used as the archetype for first-order algorithms. The extension to all the aforementioned algorithms is described in section 3.

ISTA constructs a sequence of iterates $\{\mathbf{x}_k\}_{k\ge0}$ which converges to the optimal $\tilde{\mathbf{x}}$ by implementing the update:

$$\mathbf{x}_{k+1} \leftarrow \mathcal{T}_{\lambda/L_k}\left(\mathbf{x}_k - \frac{1}{L_k}\mathbf{D}^T(\mathbf{D}\mathbf{x}_k - \mathbf{y})\right), \tag{5}$$

where $\mathbf{x}_k$ is the k-th iterate computed by the procedure, L_k is the step size (set according to a backtracking rule see [1]), and $\mathcal{T}_t(\mathbf{x}) \triangleq \operatorname{sign}(\mathbf{x})\max(0, |\mathbf{x}| - t)$ is the soft-thresholding operator. In the following update (5) is denoted $\mathbf{x}_{k+1} \leftarrow p_k(\mathbf{x}_k, \mathbf{D})$ subsequently, in order to ease the notation and emphasize similarities in first-order algorithms.

Due to the matrix-vector products involving $\mathbf{D}$ and $\mathbf{D}^T$, the cost of one update is $\mathcal{O}(NK)$, assuming that $\mathbf{D}$ has no associated fast transform. In many applications, the dimensions can be large, e.g., $K \ge N \gg 100$. This explains the major interest of reducing the size of the dictionary K without affecting the solution of the Lasso.

3 Optimizing with Dynamic Screening

Existing screening strategies for the *Lasso* are static in the sense that they first screen the dictionary and use the screened dictionary to solve the *Lasso* (see Algorithm 1). We show in this section that calculations made during the optimization procedure can be employed to dynamically and iteratively reduce the dictionary by performing dynamic screening at each iteration.

Dynamic construction of better feasible points. Screening tests presented in Section 2.2 build on a feasible dual point. Therefore, producing at each iteration a feasible dual point that is cheap to compute and close to $\mathbf{y}/\lambda$ enable the iterative construction of new SAFE sphere with smaller radius, and thus the iterative construction of more efficient sphere tests.

ISTA directly computes potentially appropriate dual points. Indeed, each update requires the computation of the gradient $\nabla f(\mathbf{x}) = \mathbf{D}^T(\mathbf{D}\mathbf{x} - \mathbf{y})$ of the ℓ_2-fitting term $f(\mathbf{x}) \triangleq \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2$. The dual points $\boldsymbol{\theta}_k \triangleq \mathbf{D}\mathbf{x}_k - \mathbf{y}$ form a sequence $\{\boldsymbol{\theta}_k\}_{k \geq 0}$ that converges to $\lambda\hat{\boldsymbol{\theta}}$. Since $\boldsymbol{\theta}_k$ is not necessarily feasible, the following *dual scaling* strategy may be resorted to, in order to give a second sequence of appropriate *feasible* dual points $\hat{\boldsymbol{\theta}}_k \triangleq \mu_k \boldsymbol{\theta}_k$.

Lemma 2 (Dual Scaling [7])**.** *Among all feasible scaled versions of $\boldsymbol{\theta}_k$, the closest to $\mathbf{y}/\lambda$ is $\hat{\boldsymbol{\theta}}_k = \mu_k \boldsymbol{\theta}_k$ where:*

$$\begin{aligned} \mu_k &\triangleq \underset{\mu \in \mathbb{R}}{\arg\min} \left\| \mu\boldsymbol{\theta}_k - \frac{\mathbf{y}}{\lambda} \right\|_2 \; \textit{s.t.} \; \|\mathbf{D}^T \mu \boldsymbol{\theta}_k\|_\infty \leq 1 \qquad (6) \\ &= \min\left(\max\left(\frac{\boldsymbol{\theta}_k^T \mathbf{y}}{\lambda \|\boldsymbol{\theta}_k\|_2^2}, \frac{1}{\|\mathbf{D}^T \boldsymbol{\theta}_k\|_\infty} \right), \frac{-1}{\|\mathbf{D}^T \boldsymbol{\theta}_k\|_\infty} \right). \end{aligned}$$

Dynamic screening. Embedding this dual scaling strategy within ISTA permits to execute more efficient screening tests almost for free. Indeed, the center of all the spheres is the same, namely $\mathbf{y}/\lambda$, hence the test vector $\mathbf{D}^T\mathbf{c}$ is computed only once. Since $\boldsymbol{\theta}_k$ and $\mathbf{D}^T\boldsymbol{\theta}_k$ are computed by the update of ISTA, computing $\hat{\boldsymbol{\theta}}_k$ requires $\mathcal{O}(K + N)$ operations and computing the sphere radius $\|\hat{\boldsymbol{\theta}}_k - \mathbf{y}/\lambda\|_2$ requires $\mathcal{O}(N)$ additional operations. Given that $N \leq K$, the computational overhead of the screening test is in $\mathcal{O}(K)$. Finally, the total overhead is negligible compared with the $\mathcal{O}(KN)$ operations required for an optimization update.

The resulting ISTA with dynamic screening is presented in Algorithm 3. The input of the algorithm defines: the problem of interest through $\mathbf{D}$, $\mathbf{y}$ and λ; the initial state $\mathbf{x}_0$ (set by default to $\mathbf{0}$); the center $\mathbf{c}$ and the function $r(\cdot)$, which compute the radius of the sphere from $\boldsymbol{\theta}$, parameterizing which sphere test is embedded in ISTA—more information on centers and radius functions $r(\cdot)$ are given below in (7) and (8). The algorithm breaks up in two steps: the optimization update (see line 4) where $p_k(\cdot)$ returns iterate $\mathbf{x}_k$ as well as suitable computed vectors $\boldsymbol{\theta}_k$ and $\mathbf{D}^T\boldsymbol{\theta}_k$; and the screening step (see line 6-8): at line 6 the feasible dual point $\hat{\boldsymbol{\theta}}_k$ is computed with the dual scaling strategy, the radius is updated only when decreasing at line 7 and finally the screening operator is applied.

Algorithm 3 ISTA with dynamic screening

```
Require: $\mathbf{D}, \mathbf{y}, \lambda, \mathbf{x}_0, \mathbf{c}, r(.)$
 1: $\mathbf{D}_0 \leftarrow \mathbf{D}$, $r_0 \leftarrow +\infty$
 2: while stopping criteria on $\mathbf{x}_k$ do
 3:     ....................... Optimization Update .......................
 4:     $\{\mathbf{x}_{k+1}, \boldsymbol{\theta}_k, \mathbf{D}^T\boldsymbol{\theta}_k\} \leftarrow p_k(\mathbf{x}_k, \mathbf{D}_k)$    ▷ see (5)
 5:     .............................. Screening ..............................
 6:     $\hat{\boldsymbol{\theta}}_{k+1} \leftarrow \mu_k \boldsymbol{\theta}_k$    ▷ see (6)
 7:     $r_{k+1} \leftarrow \min(r(\hat{\boldsymbol{\theta}}_{k+1}), r_k)$
 8:     $\mathbf{D}_{k+1} \leftarrow \Pi_{\mathbf{c}, r_{k+1}}(\mathbf{D})$    ▷ see (4)
 9:     $k \leftarrow k + 1$
10: end while
```

Generalization to other first-order algorithms. As announced in section 2.3, the dynamic screening principle applies to other first-order algorithms as well. Considering that the update does not only modify the iterate $\mathbf{x}_k$ but some auxiliary variables as well, each first-order algorithm can be describe by its update $\bar{\mathbf{x}}_{k+1} \leftarrow p_k(\bar{\mathbf{x}}_k, \mathbf{D})$ where $\bar{\mathbf{x}}_k$ represents the set of updated variables. Both p_k and $\bar{\mathbf{x}}$ are given in Table 1 for a few other representative algorithms.

Table 1 gives two important pieces of information: first all $p_k(\cdot)$ have similar computational requirements as ISTA; second the expensive computations required for dynamic screening—computing a new $\boldsymbol{\theta}_k$ and $\mathbf{D}^T\boldsymbol{\theta}_k$—are provided by these updates. Then dynamic screening applies to these algorithms in the exact same way as it applies to ISTA.

Algorithms	Optimization *update* $\bar{\mathbf{x}}_{k+1} \leftarrow p_k(\bar{\mathbf{x}}_k, \mathbf{D})$
TwIST [2] $\bar{\mathbf{x}}_k = \{\mathbf{x}_k, \mathbf{x}_{k-1}\}$	$\mathbf{x}_{k+1} \leftarrow (1-\alpha)\mathbf{x}_{k-1} + (\alpha - \beta)\mathbf{x}_k$ $+\beta\mathcal{T}_\lambda\left(\mathbf{x}_k - \mathbf{D}^T(\mathbf{D}\mathbf{x}_k - \mathbf{y})\right)$
SpaRSA [12]	same as ISTA except that L_k is set with Brazilai-Borwein rule
FISTA [1] $\bar{\mathbf{x}}_k =$ $\{\mathbf{x}_k, \mathbf{z}_k, t_k\}$	$\mathbf{x}_{k+1} \leftarrow \mathcal{T}_{\frac{\lambda}{L_k}}\left(\mathbf{z}_k - \frac{1}{L_k}\mathbf{D}^T(\mathbf{D}\mathbf{z}_k - \mathbf{y})\right)$ $t_{k+1} \leftarrow \dfrac{1 + \sqrt{1 + 4t_k}}{2}$ $\mathbf{z}_{k+1} \leftarrow \mathbf{x}_{k+1} + (\frac{t_k - 1}{t_{k+1}})(\mathbf{x}_{k+1} - \mathbf{x}_k)$
Chambolle- Pock [3] $\bar{\mathbf{x}}_k = \{\mathbf{x}_k, \hat{\mathbf{x}}_k,$ $\boldsymbol{\theta}_k, \tau_k, \sigma_k\}$	$\boldsymbol{\theta}_{k+1} \leftarrow \dfrac{1}{1+\sigma_k}(\boldsymbol{\theta}_k + \sigma_k(\mathbf{D}\hat{\mathbf{x}}_k - \mathbf{y}))$ $\mathbf{x}_{k+1} \leftarrow \mathcal{T}_{\lambda\tau_k}\left(\mathbf{x}_k - \tau_k\mathbf{D}^T\boldsymbol{\theta}_{k+1}\right)$ $\varphi_k \leftarrow \frac{1}{\sqrt{1+2\gamma\tau_k}}; \tau_{k+1} \leftarrow \varphi_k\tau_k$ $\sigma_{k+1} \leftarrow \frac{\sigma_k}{\varphi_k}$ $\hat{\mathbf{x}}_{k+1} \leftarrow \mathbf{x}_{k+1} + \varphi_k(\mathbf{x}_{k+1} - \mathbf{x}_k)$

Table 1: *Updates* for first-order algorithms.

Direct extension to ST3. Section 2.2 presents SAFE spheres, another sphere called ST3 relying on the SAFE sphere have been proposed in [14]. Constructed from any feasible dual point $\boldsymbol{\theta}$, ST3 is the sphere centered on $\mathbf{c} = \mathbf{y}/\lambda - \delta\mathbf{d}_*$ with radius $r(\boldsymbol{\theta}) = \sqrt{\|\boldsymbol{\theta} - \mathbf{y}/\lambda\|_2^2 - \delta^2}$, where $\delta = \lambda_*/\lambda - 1$. The corresponding screening operator is $\Pi_{\mathbf{c},r}(\cdot)$ as defined in (4). Both SAFE and ST3 can be embedded in Algorithm 3

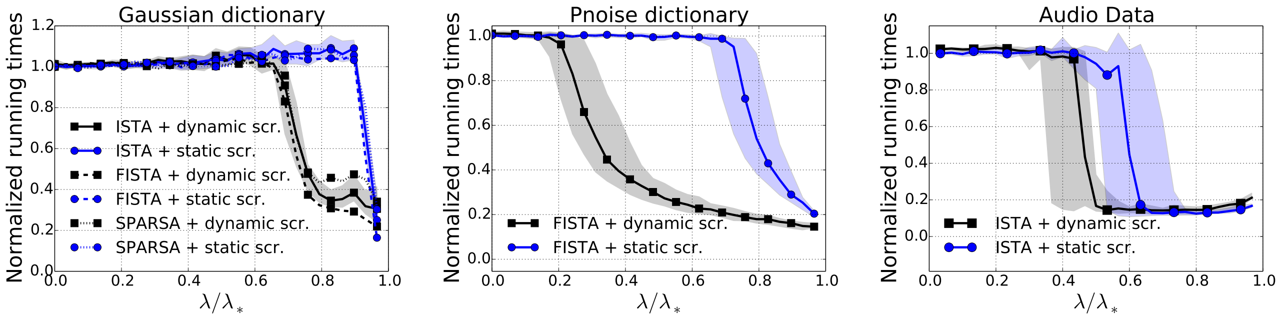

Fig. 1: Normalized running times on synthetic data (left, middle) and real data (right).

through parameters $\mathbf{c}$ and $r(\cdot)$:

$$\text{SAFE: } \mathbf{c} = \frac{\mathbf{y}}{\lambda}, \qquad r(\boldsymbol{\theta}) = \|\boldsymbol{\theta} - \frac{\mathbf{y}}{\lambda}\|_2 \tag{7}$$

$$\text{ST3: } \mathbf{c} = \mathbf{y}/\lambda - \delta\mathbf{d}_*, \quad r(\boldsymbol{\theta}) = \sqrt{\|\boldsymbol{\theta} - \mathbf{y}/\lambda\|_2^2 - \delta^2} \tag{8}$$

Convergence analysis. First-order algorithms with dynamic screening do not necessarily provide the same iterates as their base version but still converge to the global optimum:

Theorem 3. *If a first-order algorithm is proven to converge to the global optimum of the Lasso problem, then its version with dynamic screening converges to the global optimum too.*

Proof. As explained in section 2.2, *Lasso* problems $\mathcal{P}(\lambda, \mathbf{D}, \mathbf{y})$ and $\mathcal{P}(\lambda, \mathbf{D}_k, \mathbf{y})$ for all $k \geq 0$ have the same solutions. Since the sequence $\{r_k\}_{k\geq 0}$ is non-increasing, Lemma 1 ensures that the set of located inactive atoms is non-decreasing, indeed:

$$r \geq r' \Rightarrow (\forall i \in \Omega, |\mathbf{c}^T\mathbf{d}_i| < 1 - r \Rightarrow |\mathbf{c}^T\mathbf{d}_i| < 1 - r').$$

This set is upper bounded by the set of zeros in $\tilde{\mathbf{x}}$ the solution of $\mathcal{P}(\lambda, \mathbf{D}, \mathbf{y})$, so the set of located zeros converges in a finite number of iterations k_0. Then $\forall k \geq k_0, \mathbf{D}_{k_0} = \mathbf{D}_k$ and usual convergence proofs apply. □

4 Numerical Experiments

This section presents experiments used to assess the practical relevance of our approach. The code and data for numerical experiments are released for reproducible research purposes.[1]

Runnning Times. We have claimed that, compared with static screening, dynamic screening significantly accelerates the computation of the solution of the *Lasso* with first-order algorithms. This section evaluates the performance of our method in terms of running times. Note that since each version of the algorithm (no-screening, static screening, dynamic screening) converges to the same optimal $\tilde{\mathbf{x}}$ (see Theorem 3), we do not report the value of the objective at convergence .

We measured running times of the algorithm without screening test, with static screening and with dynamic screening. To emphasize the gain, running times are normalized with respect to running times required by the algorithm without screening.

Synthetic data. For experiments on synthetic data, we used two types of dictionaries. The first one is a Gaussian dictionary where observation $\mathbf{y}$ as well as all atoms $\mathbf{d}_i$ are drawn i.i.d. uniformly on the unit sphere by normalizing realizations of $\mathcal{N}(\mathbf{0}, \mathbf{I}_N)$. The second one is the so-called Pnoise introduced in [13], for which $\mathbf{y}$ and all $\mathbf{d}_i$ are drawn i.i.d. from the distribution $\mathbf{e}_1 + 0.1\kappa\mathbf{g}$ and normalized, where $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I_N})$, $\kappa \sim \mathcal{U}(0, 1)$ and $\mathbf{e}_1$ being the first natural basis vector. We set $N = 2000$ and $K = 10000$.

Audio Data. For experiments on real data we performed the estimation of the sparse representation of audio signals in a redundant Discrete Cosine Transform (DCT) dictionary, which is known to be adapted for audio data. Observations $\mathbf{y}$ were taken from 30 music and speech recordings with length $N = 1024$ and sampling rate 16 kHz. Experiments were run for $K = 3N$.

Experiments. Algorithms were run for several values of λ in order to compute the representation of the observation $\mathbf{y}$ in the dictionary $\mathbf{D}$ with different sparsity levels. The algorithm stops at iteration k when the ratio between the maximum variation of the functional in (1) and the average of the functional over the $M = 10$ previous iterations does not exceed the value of ϵ ($\epsilon = 10^{-6}$ for Gaussian, $\epsilon = 10^{-4}$ for Pnoise and $\epsilon = 10^{-6}$ for audio signals).

Figure 1 shows the normalized running times for algorithms with dynamic screening (black squares) and for the corresponding algorithms with static screening (circle) as a function of λ/λ_*. Low values account for fast computation. We used Gaussian (left), Pnoise (middle) and Audio (right) data with the ST3 screening test. We plotted the median values over 30 problems. The shaded area contains from 25% to 75% percentiles for plain curves only, in order to illustrate the typical distribution of the values. For all dictionaries, the dynamic screening performs significantly better and is effective in a larger range of λ than the static one. In the audio experiment savings could reach more than 90% over ISTA and up to 70% over ISTA with static ST3 (*e.g.* for $\lambda/\lambda^* \approx 0.6$). Both static and dynamic screening strategies tend to be more

[1] http://pageperso.lif.univ-mrs.fr/~antoine.bonnefoy

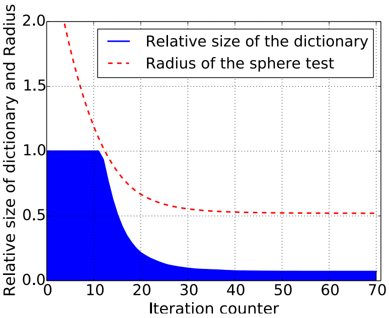

Fig. 2: Relative size K_k/K and radius r_k along the iterations.

efficient when the dictionary redundancy K/N increases.

Note that due to the normalization of running times, Figure 1 cannot be used to draw any conclusion on which of ISTA, FISTA or SpaRSA is the fastest algorithm.

Interpretation: Screening progression. To apprehend the effectiveness of the dynamic screening test, we represented how dynamic screening behaves along the iterations. Figure 2 shows on the same scale the evolution of two key values along the iterations: the radius r_k (red dashed line); and the relative size of the dictionary K_k/K (blue area)—where K_k is the size of the screened dictionary at iteration k— which represents the proportion of atoms remaining in the screened dictionary. Here dynamic ST3 was used in ISTA for a Gaussian dictionary with $\lambda = 0.7\lambda_*$. The reduction of the radius induced a nice improvement in the screening. The screening test may be totally inefficient in the first iterations, which shows the advantage of the dynamic screening strategy over the static one.

5 Discussion and Future Directions

We have shown that the dynamic screening principle is relevant theoretically and practically. Dynamic screening accelerates more first-order algorithms than static screening in the proposed experiments on synthetic and real data, and in a larger range of λ.

Dynamic screening has been shown to work for several algorithms and screening tests, and the question is whether the concept of dynamic screening can be further generalized. The answer is positive: it can be applied to much more algorithms and to other screening tests. As far as an optimization process computes the gradient of the ℓ_2-fitting term and the screening test rests upon the SAFE sphere, *e.g.* the dome test [13], they can combine into a dynamic screening strategy.

The proposed method raises several questions we plan to work on, some of them are addressed here as a conclusion. The SAFE test extends to the *Group Lasso* [15], but can it be refined dynamically along the iterations of the optimization process in a similar fashion? As in [7], we are curious to see how dynamic screening may show up when other than an ℓ_2 fit-to-data is studied: for example, this situation naturally occurs when classification-based losses are considered. As sparsity is often a desired feature for both efficiency (in the prediction phase) and generalization purposes, being able to work out well-founded results allowing dynamic screening is of the utmost importance.

REFERENCES

[1] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[2] J. M. Bioucas-Dias and M. A. T. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE TIP*, 16(12):2992–3004, 2007.

[3] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[4] P. L. Combettes and V.-R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

[5] L. Dai and K. Pelckmans. An ellipsoid based, two-stage screening test for bpdn. In *Proc. of the 20th Eur. Sig. Proc. Conf. (EUSIPCO)*, pages 654–658, 2012.

[6] I. Daubechies, M. Defrise, and C. De Mol. An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint. *Communications on Pure and Applied Mathematics*, 1457:1413–1457, 2004.

[7] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe Feature Elimination in Sparse Supervised Learning. Technical report, EECS Department, University of California, Berkeley, 2010.

[8] H. Uzawa. K. J. Arrow, L. Hurwicz. *Studies in linear and nonlinear programming, With contributions by Hollis B. Chenery [and others]*. Stanford University Press, Stanford, Calif, 1964.

[9] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[10] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *J. of the Royal Stat. Society: Series B*, 74(2):245–266, 2012.

[11] J. Wang, B. Lin, P. Gong, P. Wonka, and J. Ye. Lasso Screening Rules via Dual Polytope Projection. *CoRR*, pages 1–17, 2012.

[12] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. on Sig. Proc.*, 57(7):2479–2493, 2009.

[13] Z. J. Xiang and P. J. Ramadge. Fast lasso screening tests based on correlations. In *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc. (ICASSP)*, pages 2137–2140, 2012.

[14] Z. J. Xiang, H. Xu, and P. J. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *NIPS 2011, vol. 24*, pages 900–908, 2011.

[15] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2006.