HAL open science

# S-ISTA and Brothers : a Dynamic Screening test principle for the Lasso

Antoine Bonnefoy, Valentin Emiya, Liva Ralaivola, Rémi Gribonval

▶ **To cite this version:**

Antoine Bonnefoy, Valentin Emiya, Liva Ralaivola, Rémi Gribonval. S-ISTA and Brothers : a Dynamic Screening test principle for the Lasso. 2013. hal-00880787v1

**HAL Id: hal-00880787**

**https://hal.science/hal-00880787v1**

Submitted on 6 Nov 2013 (v1), last revised 24 Jun 2014 (v4)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# S-ISTA AND BROTHERS : A DYNAMIC SCREENING TEST 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 find a *sparse* representation of some signal with respect to some predefined dictionary. We propose an original and computationally efficient method to solve the *Lasso* problem. Our method rests upon the idea of *dynamic screening test* and allows us to accelerate first-order algorithms. At each iteration we take advantage of the computation done for the optimization process to reduce the size of the dictionary by discarding elements that will surely not enter the sparse representation. As this *screening* step is not expensive, the computational cost of the algorithm using the dynamically screened dictionary is cheaper than the standard algorithm. Numerical simulations on synthetic and real data support the relevance of our approach.

***Index Terms*—** Optimization, LASSO, Screening test, Algorithms, Sparsity

## 1. INTRODUCTION

The *Lasso* [10] is an optimization problem that consists in minimizing the sum of an $\ell_2$-fitting term and an $\ell_1$-regularization term aimed at promoting a sparse solution. Given an observation $\mathbf{y} \in \mathbb{R}^N$ and some dictionary matrix $\mathbf{D}$, this problem writes as

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

where $\lambda > 0$ is a parameter that governs the sparsity of its solution $\tilde{\mathbf{x}}$. Many algorithms have been developed to solve this problem, and we focus our study on first-order algorithms, which include primal algorithms [1, 4, 6, 15] and some primal-dual algorithms [3, 8]. As they rely on the Soft-Thresholding operator when applied to the *Lasso*, they will be referred to as IST —for "Iterative Soft Thresholding"— 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, approaches based on *screening tests* [5, 7, 11, 13, 17, 16] have recently been proposed. They rely on a two-step strategy: i) they locate zeros in $\tilde{\mathbf{x}}$ and ii) they solve a reduced version $\mathcal{P}(\lambda, \mathbf{D}_0, \mathbf{y})$ of (1), where $\mathbf{D}_0$ is the dictionary $\mathbf{D}$ trimmed off its columns that correspond to the identified zeros of $\tilde{\mathbf{x}}$ (see Algorithm 1).

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

We propose a new principle called *Dynamic Screening* to reduce the computational cost of IST algorithms even more. We take the aforementioned idea of *screening* one step further, and improve the screening efficiency of existing *static* screening tests by embedding them in the iterations of the IST algorithms. To our knowledge, this is the first time such a screening mechanism is envisioned. We perform *screening* at each iteration with a negligible computational overhead, and we consequently *dynamically* reduce the size of $\mathbf{D}$. The opposing perspectives are depicted in the generic view of *static* screening (Algorithm 1) vs. *dynamic* screening (Algorithm 2), where $\Xi\,(.)$ denotes an existing screening test and $p_k(.)$ denotes one iteration of an IST algorithm. Experiments exhibit that the proposed *dynamic screening* method significantly lowers the computational cost of the optimization in a large range of $\lambda$. The computational saving reaches 90% with respect to the algorithm alone, or up to 60% with respect to the algorithm run after existing screening tests.

**Algorithm 1**
Static screening

Initialize $\mathbf{x}_0$
- - - - Screening - - - -
$\mathbf{D}_0 \leftarrow \Xi\,(\mathbf{D})$
**loop** $k$
- - - - Iteration step - - - -
$\bar{\mathbf{x}}_{k+1} \leftarrow p_k(\bar{\mathbf{x}}_k, \mathbf{D}_0)$
**end loop**

**Algorithm 2**
Proposed Dynamic screening

Initialize $\mathbf{x}_0$
$\mathbf{D}_0 \leftarrow \mathbf{D}$
**loop** $k$
- - - - Iteration step - - - -
$\bar{\mathbf{x}}_{k+1} \leftarrow p_k(\bar{\mathbf{x}}_k, \mathbf{D}_k)$
- - - - Screening - - - -
$\mathbf{D}_{k+1} \leftarrow \Xi_k\,(\mathbf{D}_k)$
**end loop**

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

## 2. EXISTING 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

Let us denote by $\mathbf{D} \triangleq [\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{N\times K}$ a *dictionary*. $\Omega = \{1, \ldots, K\}$ denotes the set of integers indexing the atoms of $\mathbf{D}$. The observation $\mathbf{y} \in \mathbb{R}^N$ is supposed to have a sparse representation in $\mathbf{D}$ denoted by $\mathbf{x} \in \mathbb{R}^K$, *i.e.* $\mathbf{D}\mathbf{x}$ is aimed at approximating $\mathbf{y}$. For any vector $\mathbf{v}$, the $i$-th component is $\mathbf{v}(i)$. Without loss of generality,

we assume that the observation $\mathbf{y}$ and the atoms $\mathbf{d}_k$ have unit $\ell_2$ norm. The dual problem associated to (1) is [7, 17]:

$$\arg\max_{\boldsymbol{\theta}} \frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{\lambda^2}{2}\left\|\boldsymbol{\theta} - \frac{\mathbf{y}}{\lambda}\right\|_2^2 \text{ s.t. } \forall i \in \Omega, |\boldsymbol{\theta}^T \mathbf{d}_i| \leq 1, \quad (2)$$

The solutions of the primal (1) and dual (2) problems denoted by $\tilde{\mathbf{x}}$ and $\tilde{\boldsymbol{\theta}}$ respectively, are linked by the relation:

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

We additionally define $\mathbf{d}_*$, $\lambda_*$ and $\delta$:

$$\mathbf{d}_* = \underset{\mathbf{d} \in \{\pm \mathbf{d}_i\}_{i=1}^K}{\arg\max} \mathbf{d}^T \mathbf{y}, \qquad \lambda_* = \mathbf{d}_*^T \mathbf{y}, \qquad \delta = \frac{\lambda_*}{\lambda} - 1$$

We assume afterwards that $\lambda_* > \lambda$ to avoid the trivial null solution.

### 2.2. Screening Tests

The sparsity inducing regularization $\lambda\|.\|_1$ entails an optimum $\tilde{\mathbf{x}}$ that contains many zeros, and the goal of a screening test is precisely to locate as many of them; we say a screening test sharpens when the number of located zeros grows. The corresponding columns in $\mathbf{D}$ can be removed consequently without changing the solution of the problem, and the optimization procedure using the reduced dictionary may be performed with a lower computational cost.

Screening tests [7, 16, 17] are based on a general idea emerging from the relation (3) between $\tilde{\mathbf{x}}$ and $\tilde{\boldsymbol{\theta}}$. From the knowledge of a region $\mathcal{R} \subset \mathbb{R}^N$ containing $\tilde{\boldsymbol{\theta}}$, we have for all $i \in \Omega$:

$$\max_{\boldsymbol{\theta} \in \mathcal{R}} |\boldsymbol{\theta}^T \mathbf{d}_i| < 1 \Rightarrow |\tilde{\boldsymbol{\theta}}^T \mathbf{d}_i| < 1 \Rightarrow \tilde{\mathbf{x}}(i) = 0 \quad (4)$$

When $\mathcal{R}$ is a sphere, the left-hand side of (4) has a closed-form expression and it gives rise to the general sphere test principle:

**Lemma 1** (General Sphere Test Principle [7]). *If the solution $\tilde{\boldsymbol{\theta}}$ of (2) satisfies $\|\tilde{\boldsymbol{\theta}} - \mathbf{c}\|_2 \leq r$, then $|\mathbf{c}^T \mathbf{d}_i| < 1 - r \Rightarrow \tilde{\mathbf{x}}(i) = 0$.*

We define the sphere test operator $\Xi_{\mathbf{c},r}(\cdot)$ associated to the sphere $\mathcal{S}(\mathbf{c}, r)$ of center $\mathbf{c}$ and radius $r$, the operator that, given a dictionary $\mathbf{D}$, outputs the corresponding *screened* dictionary

$$\Xi_{\mathbf{c},r}(\mathbf{D}) \triangleq \left[\mathbf{d}_i \text{ s.t. } i \in [1..N], |\mathbf{c}^T \mathbf{d}_i| \geq 1 - r\right]. \quad (5)$$

Figure 1a illustrates the general sphere test in two dimensions: $\mathcal{S}(\mathbf{0}, 1)$ is the unit sphere on which the atoms $\mathbf{d}_i$ live; if $\tilde{\boldsymbol{\theta}}$ is in $\mathcal{S}(\mathbf{c}, r)$, then every atom in the red area is removed by $\Xi_{\mathbf{c},r}(\cdot)$.

Two instances of this sphere test principle, SAFE/ST1 [7, 17] and ST3 [17], are represented on Figure 1b. These screening tests are constructed from the feasible dual point $\hat{\boldsymbol{\theta}} = \mathbf{y}/\lambda_*$, where feasible means that it complies with the constraints of (2). This point allows one to construct the sphere test ST1 centered on the solution $\mathbf{y}/\lambda$ of the unconstrained dual problem. ST3 refines it relying on an other center but the same feasible point $\mathbf{y}/\lambda_*$. Xiang [16] proposed an alternative to refine ST3 when $\mathcal{R}$ is a dome. Equations of the centers and radius (seen as a function $\rho$ of $\hat{\boldsymbol{\theta}}$) are given for these tests in Table 1.

A key idea of our method is to exhibit better feasible dual points $\hat{\boldsymbol{\theta}}$ to reduce the radius of the spheres as shown is Figure 1b. Spheres constructed as ST1 or ST3 but from any feasible dual point $\hat{\boldsymbol{\theta}}$ are denoted by DST1 or DST3 respectively.

(a) General sphere test principle (b) Instances of sphere tests

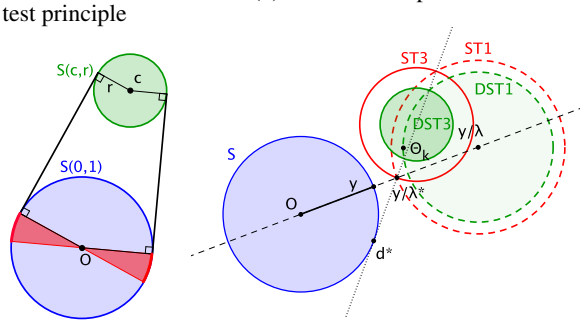


**Fig. 1**: The sphere tests

| | $\mathbf{c}$ | $r = \rho(\hat{\boldsymbol{\theta}})$; $\hat{\boldsymbol{\theta}} = \frac{\mathbf{y}}{\lambda_*}$ | Screening Operator |
|---|---|---|---|
| ST1 | $\frac{\mathbf{y}}{\lambda}$ | $\|\hat{\boldsymbol{\theta}} - \frac{\mathbf{y}}{\lambda}\|_2$ | $\Xi_{\mathbf{c},r}(\mathbf{D})$ |
| ST3 | $\frac{\mathbf{y}}{\lambda} - \delta\mathbf{d}_*$ | $\sqrt{\left\|\hat{\boldsymbol{\theta}} - \frac{\mathbf{y}}{\lambda}\right\|_2^2 - \delta^2}$ | $\Xi_{\mathbf{c},r}(\mathbf{D})$ |
| Dome | $\frac{\mathbf{y}}{\lambda} - \delta\mathbf{d}_*$ | $\sqrt{\left\|\hat{\boldsymbol{\theta}} - \frac{\mathbf{y}}{\lambda}\right\|_2^2 - \delta^2}$ | Dome [16] |

**Table 1**: characterization of existing screening tests

### 2.3. Solving the *Lasso* with IST Algorithms

The *Lasso* problem (1) may be solved with general-purpose algorithms such as ISTA [6], TwIST [2], FISTA [1], SpaRSA [15], forward-backward splitting [4] or first-order primal-dual algorithm *e.g.* [3, 8].

These algorithms construct a sequence $\{\bar{\mathbf{x}}_k\}_{k\geq 0}$, iterating the *step* $\bar{\mathbf{x}}_{k+1} = p_k(\bar{\mathbf{x}}_k, \mathbf{D})$. Each $\bar{\mathbf{x}}_k$ is a *set* of variables that contains an iterate $\mathbf{x}_k$ as well as auxiliary variables; the corresponding sequence $\{\mathbf{x}_k\}_{k\geq 0}$ converges to the optimal $\tilde{\mathbf{x}}$.

The objective function in (1) is naturally split into a sum of a convex and differentiable function $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2$ and a convex non differentiable function $g(\mathbf{x}) = \lambda\|\mathbf{x}\|_1$. To handle the non-smoothness of $g$, IST algorithms use its proximal operator, which reduces to the so called soft-thresholding operator $\mathcal{T}_t(\cdot)$:

$$\mathcal{T}_t(\mathbf{x})(i) \triangleq \text{sign}(\mathbf{x}(i)) \max(0, |\mathbf{x}(i)| - t) \quad (6)$$

Table 2 details the *step* function $p_k(\cdot)$ of various IST algorithms. This table additionally describes the set $\bar{\mathbf{x}}_k$. The step-size $L_k$ is set according to the backtracking rule in ISTA-FISTA and to the Brazilai-Borwein rule in SpaRSA; the other parameters $\alpha, \beta$ and $\gamma$ are set according to the recommendations provided in the relevant papers.

All $p_k(.)$ have similar computational requirements. In the general case, their computational costs are in $\mathcal{O}(NK)$, and for dictionaries associated with fast transforms, this may be lowered to $\mathcal{O}(K\log(N))$[1]. In many applications, the dimensions can be very large, e.g., $K \geq N \gg 100$, whence the major computational interest in reducing the size of the dictionary.

## 3. OPTIMIZING WITH DYNAMIC SCREENING TESTS

As the computational cost of a *step* of the optimization procedure is driven by the size of the dictionary, it is worth finding a way to sharpen existing screening tests at low computational cost.

[1] When $\mathbf{x}$ is sparse, computing $\mathbf{D}\mathbf{x}$ may be done with fewer operations.

| Algorithms | Optimization *Step* $\bar{\mathbf{x}}_{k+1} \leftarrow p_k(\bar{\mathbf{x}}_k, \mathbf{D})$ |
|---|---|
| ISTA $\bar{\mathbf{x}}_k = \{\mathbf{x}_k\}$ | $\mathbf{x}_{k+1} \leftarrow \mathcal{T}_{\lambda/L_k}\left(\mathbf{x}_k - \frac{1}{L_k}\mathbf{D}^T(\mathbf{D}\mathbf{x}_k - \mathbf{y})\right)$ |
| TwIST $\bar{\mathbf{x}}_k = \{\mathbf{x}_k, \mathbf{x}_{k-1}\}$ | $\mathbf{x}_{k+1} \leftarrow (1-\alpha)\mathbf{x}_{k-1} + (\alpha - \beta)\mathbf{x}_k$ $+\beta\mathcal{T}_\lambda\left(\mathbf{x}_k - \mathbf{D}^T(\mathbf{D}\mathbf{x}_k - \mathbf{y})\right)$ |
| SpaRSA | idem as ISTA with a different $L_k$ |
| FISTA $\bar{\mathbf{x}}_k = \{\mathbf{x}_k, \mathbf{z}_k, t_k\}$ | $\mathbf{x}_{k+1} \leftarrow \mathcal{T}_{\lambda/L_k}\left(\mathbf{z}_k - \frac{1}{L_k}\mathbf{D}^T(\mathbf{D}\mathbf{z}_k - \mathbf{y})\right)$ $t_{k+1} \leftarrow \dfrac{1+\sqrt{1+4t_k}}{2}$ $\mathbf{z}_{k+1} \leftarrow \mathbf{x}_{k+1} + (\frac{t_k - 1}{t_{k+1}})(\mathbf{x}_{k+1} - \mathbf{x}_k)$ |
| Chambolle-Pock $\bar{\mathbf{x}}_k = \{\mathbf{x}_k, \hat{\mathbf{x}}_k, \boldsymbol{\theta}_k, \tau_k, \sigma_k\}$ | $\boldsymbol{\theta}_{k+1} \leftarrow \dfrac{1}{1+\sigma_k}(\boldsymbol{\theta}_k + \sigma_k(\mathbf{D}\hat{\mathbf{x}}_k - \mathbf{y}))$ $\mathbf{x}_{k+1} \leftarrow \mathcal{T}_{\lambda\tau_k}\left(\mathbf{x}_k - \tau_k\mathbf{D}^T\boldsymbol{\theta}_{k+1}\right)$ $\varphi_k \leftarrow \frac{1}{\sqrt{1+2\gamma\tau_k}}; \tau_{k+1} \leftarrow \varphi_k\tau_k; \sigma_{k+1} \leftarrow \frac{\sigma_k}{\varphi_k}$ $\hat{\mathbf{x}}_{k+1} \leftarrow \mathbf{x}_{k+1} + \varphi_k(\mathbf{x}_{k+1} - \mathbf{x}_k)$ |

**Table 2**: *Steps* for the algorithm describe in section 2.3.

**The dynamic viewpoint.** We can consider that existing screening methods for the *Lasso* are static in the sense that they first screen the dictionary and then fix it to solve the *Lasso* (see Algorithm 1). We show in this section that calculations *made during the optimization procedure* can be employed to *dynamically* screen the dictionary iteration after iteration as conveyed by lines 5 to 8 of Algorithm 3. The *Dynamically Screened* version of an algorithm will be denoted preceded by S *e.g.* S-ISTA.

**Dynamic construction of better feasible points.** Screenings tests presented in Section 2.2 build on feasible dual points. We therefore would like to be able to compute interesting feasible dual points $\hat{\boldsymbol{\theta}}$ to refine the sphere tests, where interesting means cheap to compute and close to $\mathbf{y}/\lambda$. These $\hat{\boldsymbol{\theta}}$ may enable the construction of spheres DST1 or DST3 with smaller radius.

IST algorithms directly calculate possibly interesting $\boldsymbol{\theta}$. Indeed, iterates of the primal variables require the computation of the gradient $\nabla f(\mathbf{x}) = \mathbf{D}^T(\mathbf{D}\mathbf{x} - \mathbf{y})$ of the fitting term to perform one iteration. The points $\boldsymbol{\theta}_k = \mathbf{D}\mathbf{x}_k - \mathbf{y}$ form a sequence $\boldsymbol{\theta}_k$ that converges to $\lambda\tilde{\boldsymbol{\theta}}$ (cf. (3)). The primal-dual algorithms as well directly calculate a sequence of dual points. As these sequences converge to the optimal of (2) they are likely to entail decreasing radius and therefore sharper sphere test. Since $\boldsymbol{\theta}_k$ is not always good and feasible, the dual scaling strategy may be resorted to, in order to give:

$$\hat{\boldsymbol{\theta}}_k \triangleq \hat{s}\boldsymbol{\theta}_k \text{ where } \hat{s} = \underset{s\in\mathbb{R}}{\arg\min} \left\| s\boldsymbol{\theta}_k - \frac{\mathbf{y}}{\lambda}\right\|_2 \text{ s.t. } \|\mathbf{D}^T s\boldsymbol{\theta}_k\|_\infty \leq 1 \tag{7}$$

The solution of this problem is given in the following lemma:

**Lemma 2** (Dual Scaling [7])**.** *Among all feasible scaled versions of $\boldsymbol{\theta}_k$, the closest to $\mathbf{y}/\lambda$, i.e. the optimum of (7), is given by :*

$$\hat{\boldsymbol{\theta}}_k \triangleq \Theta(\boldsymbol{\theta}_k) = \mu\frac{\boldsymbol{\theta}_k}{\lambda_0}, \quad \textit{where} \quad \lambda_0 = \|\mathbf{D}^T\boldsymbol{\theta}_k\|_\infty \tag{8}$$

$$\textit{and} \quad \mu = \min\left(\max\left(\frac{\lambda_0\boldsymbol{\theta}_k^T\mathbf{y}}{\lambda\|\boldsymbol{\theta}_k\|_2^2}, 1\right), -1\right)$$

From $\hat{\boldsymbol{\theta}}_k$, the radius is computed in $\mathcal{O}(N)$ operations for the tests described in Table 1. Furthermore most of the computation required for the dual scaling —*i.e.* the computation of $\mathbf{D}^T\boldsymbol{\theta}_k$— is already done by the optimization procedures described in Table 2. Thus the computational overhead is just $\mathcal{O}(2K)$ (see Equation (8)). In addition, for a given initial sphere $\mathcal{S}(\mathbf{c}, r_0)$, the *test vector* $\mathbf{D}^T\mathbf{c}$ in Lemma 1 is calculated only once, as one only requires a smaller radius to sharpen the test at each iteration. Finally the total overhead of the dynamic screening is negligible compared to the $\mathcal{O}(KN)$ or $\mathcal{O}(K\log N)$ operations made during the optimization step.

The resulting general screened algorithm is presented in Algorithm 3. The optimization algorithm is parameterized by $p_k(.)$ and the screening test (ST1,ST3 or Dome) by $\mathbf{c}, \rho(.), \Xi(.)$. The input of the algorithm also specifies the problem of interest ($\mathbf{D}$, $\mathbf{y}$ and $\lambda$).

**Algorithm 3** General Dynamic Screening

**Require:** $\mathbf{D}, \mathbf{y}, \mathbf{c}, \rho(.), \Xi(.), p_k(.), \bar{\mathbf{x}}_0$
1: $\mathbf{D}_0 \leftarrow \mathbf{D}$, $r_0 \leftarrow +\infty$
2: **while** stopping criteria on $\bar{\mathbf{x}}_k$ **do**
3: ..... Optimization Step .....
4: $\{\bar{\mathbf{x}}_{k+1}, \boldsymbol{\theta}_{k+1}, \mathbf{D}^T\boldsymbol{\theta}_{k+1}\} \leftarrow p_k(\bar{\mathbf{x}}_k, \mathbf{D}_k)$
5: ..... Screening .....
6: $\hat{\boldsymbol{\theta}}_{k+1} \leftarrow \Theta(\boldsymbol{\theta}_{k+1})$
7: $r_{k+1} \leftarrow \min(\rho(\hat{\boldsymbol{\theta}}_{k+1}), r_k)$
8: $\mathbf{D}_{k+1} \leftarrow \Xi_{\mathbf{c}, r_{k+1}}(\mathbf{D})$
9: $k \leftarrow k+1$
10: **end while**

A dynamic screening test is at least as efficient as its corresponding static screening test. Stated in Lemma 3 this result is qualitative, but actually, dynamic screening has been designed to screen much more atoms than the existing sphere tests and, *in fine*, to drastically reduce the total computational cost. Such quantitative performance is assessed experimentally in Section 4.

**Lemma 3.** *For a given problem $\mathcal{P}(\lambda, \mathbf{y}, \mathbf{D})$, any atom screened by a given test based on SAFE is screened by its dynamic version at the first iteration if $\bar{\mathbf{x}}_0$ is initialized at $\mathbf{0}$.*

*Proof.* Since variables in $\bar{\mathbf{x}}_0$ are $\mathbf{0}$, we have $\boldsymbol{\theta}_1 = -\mathbf{y}$ and $\hat{\boldsymbol{\theta}}_1 = \frac{\mathbf{y}}{\lambda_*}$ for all the algorithm aforementioned. It is exactly the feasible dual point upon which the static versions of ST1/ST3/Dome rest. □

**Need for convergence analysis.** The screened algorithms do not necessary provide the same iterates than their usual version, this accounts for the need of a convergence analysis.

**Theorem 4.** *If an algorithm is proven to converge to the optimal of the problem, then its dynamically screened version converges too.*

*Proof.* The convergence holds since the number of possible screened dictionaries is finite. After finitely many iterations, the dictionary $\mathbf{D}_k$ becomes stable and usual convergence proofs apply. □

The proof of the convergence rate for ISTA given in [1] can be extended to S-ISTA following the same progression. Empirically dynamic screening conserves the convergence rates as experimentally S-ISTs converges in the same number of iterations than ISTs.

## 4. NUMERICAL SIMULATIONS

This section presents experiments used to assess the practical relevance of our approach. The code and data for all numerical simulations are released for reproducible research purposes.[2]

**Synthetic data.** For the experiments on synthetic data, we use two types of dictionaries. The first one is a Gaussian dictionary where the atoms $\mathbf{d}_i$ and the observation $\mathbf{y}$ are drawn i.i.d. uniformly on the unit sphere by normalizing realizations of $\mathcal{N}(\mathbf{0}, \mathbf{I}_N)$. The second

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

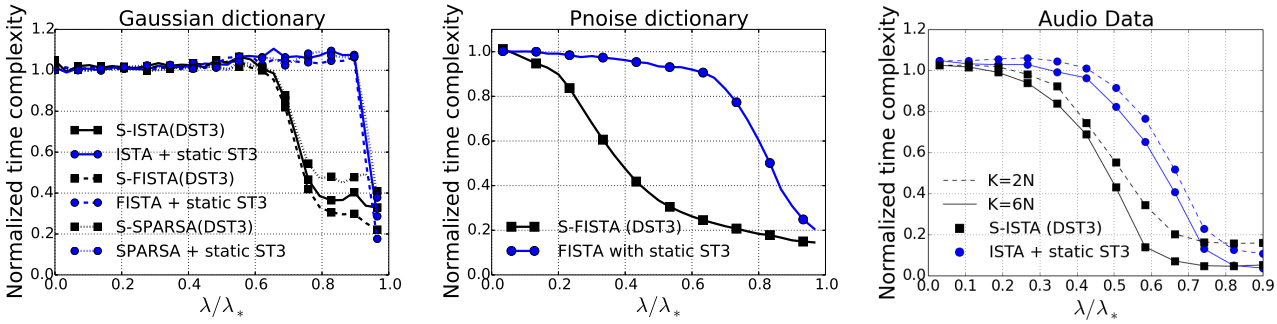


**Fig. 3**: Normalized time complexity on synthetic data (left, middle) and real data (right).

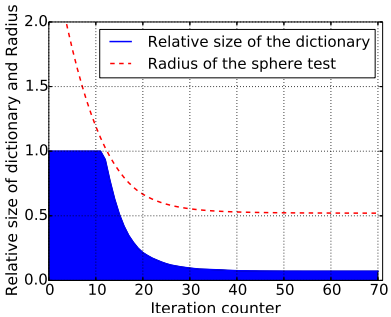


**Fig. 2**: Relative size $K_k/K$ and radius $r_k$ along the iterations.

one is the so-called Pnoise introduced in [16], which is a kind of correlated noise: $\mathbf{d}_i$ and $\mathbf{y}$ are drawn as $\mathbf{e}_1 + 0.1\kappa\mathbf{g}$ and normalized. $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I_N})$, $\kappa \sim \mathcal{U}(0,1)$, $\mathbf{e}_1$ being the first natural basis vector.

**Screening progression.** To appreciate the effectiveness of the dynamic screening, one may represent how the dynamic screening tests act along the iterations. Figure 2 shows on the same scale the evolution of two key values along the iterations: the relative size of the dictionary $K_k/K$ and the radius $r_k$. The blue area, represents the proportion of atoms remaining in the dictionary, it equals 1 when the screening does not locate any zeros.

Here dynamic ST3 is used in S-ISTA for a Gaussian dictionary with $\lambda = 0.7\lambda_*$. The reduction of the radius induces a nice improvement in the screening. We see here that the screening test may be totally inefficient in the first iterations. This shows the advantage of the dynamic screening strategy over the static one.

**Time complexity with synthetic data.** Algorithms are run for several values of $\lambda$ in order to decompose the observation $\mathbf{y}$ on the dictionary $\mathbf{D}$ with different sparsity levels. The algorithm stops at iteration $k$ when the ratio between the maximum variation of the functional and the mean of the functional in the $M = 10$ previous iterations does not exceed the value of $\epsilon$ ($10^{-6}$ for Gaussian, $10^{-6}$ for Pnoise).

The time complexity is measured in seconds and normalized by the time complexity of the algorithm used without any screening. Figure 3 (left and middle) shows the normalized time complexities for S-ISTA, S-FISTA and S-SpaRSA (black squares) and for the corresponding algorithm with static screening (circle) as a function of $\lambda/\lambda_*$, using $2000\times10000$ Gaussian (left) and Pnoise (middle) dictionaries. The values are averaged over 30 runs. For both dictionaries the dynamic screening performs significantly better and is effective in a larger range of $\lambda$ than the static one. Similar results are observed when using the dynamic Dome test, and are not detailed here.

**Time complexity in audio coding.** Finally a simple case of audio coding/denoising is presented. It consists in testing the computational efficiency of the dynamic screening by estimating the sparse representation of audio signals in a redundant Discrete Cosine Transform (DCT) dictionary, which is known to be adapted for audio data. 30 observations are taken from music and speech recordings with length at $N = 1024$. The experiments are run for two values $K$ with $\epsilon = 10^{-6}$.

The trends observed with synthetic data are confirmed with real data: the dynamic screening is significantly faster than the static screenings, and in a larger range of $\lambda$. Time savings can reach more than 90% over ISTA and up to 50% over ISTA with static ST3 (*e.g.* for $\lambda/\lambda^* \approx 0.6$). Both screening strategies are efficient when the dictionary redundancy $K/N$ increases.

## 5. DISCUSSION AND FUTURE DIRECTIONS

The idea of embedding screening tests within iterative optimization procedures has been proposed for several algorithms. It can be applied to many more algorithms but raises several questions; we here address some of these.

As it can be shown for ISTA, can we ensure that the dynamic screening preserves the convergence rate of any first-order algorithm? Answering this question would definitely anchor dynamic screening in a theoretical context.

In a recent work [14] Wang et. *al* introduce a way to adapt the static dome test in a continuation strategy. Seeing how the dome test can dynamically be adapted in an optimization procedure might be of great interest. The SAFE extends to the *Group Lasso* [18], but can it be refined dynamically along the iteration of the optimization process in similar fashion? This is another exciting subject that we plan to work on in a near future.

Given the nice theoretical and practical behavior of Orthogonal Matching Pursuit [9, 12], investigating how it can be paired with dynamic screening is a pressing and exciting matter but poses the problem of computing a value of $\lambda$ that is consistent with the sparsity targeted by orthogonal matching pursuit.

Lastly, as in [7], we are curious to see how dynamic screening may show up when other than an $\ell_2$ 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.

## 6. 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. *Image Processing, IEEE Transactions on*, 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 *Proceedings of the 20th European Signal Processing Conference (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 non-linear programming, With contributions by Hollis B. Chenery [and others]*. Stanford University Press, Stanford, Calif, 1964.

[9] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, pages 3397–3415, 1993.

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

[11] 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. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):245–266, March 2012.

[12] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

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

[14] Y. Wang, Z. J. Xiang, and P. L. Ramadge. Lasso screening with a small regularization parameter. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3342–3346, 2013.

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

[16] Z. J. Xiang and P. J. Ramadge. Fast lasso screening tests based on correlations. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2137–2140, 2012.

[17] Z. J. Xiang, H. Xu, and P. J. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *Advances in Neural Information Processing Systems*, volume 24, pages 900–908, 2011.

[18] 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.