Stable safe screening and structured dictionaries for faster ℓ_1 regularization

Cássio F. Dantas, *Student Member, IEEE,* and Rémi Gribonval, *Fellow, IEEE*

***Abstract*—In this paper, we propose a way to combine two acceleration techniques for the ℓ_1-regularized least squares problem: safe screening tests, which allow to eliminate useless dictionary atoms; and the use of fast structured approximations of the dictionary matrix. To do so, we introduce a new family of screening tests, termed *stable screening*, which can cope with approximation errors on the dictionary atoms while keeping the safety of the test (i.e. zero risk of rejecting atoms belonging to the solution support). Some of the main existing screening tests are extended to this new framework. The proposed algorithm consists in using a coarser (but faster) approximation of the dictionary at the initial iterations and then switching to better approximations until eventually adopting the original dictionary. A systematic switching criterion based on the duality gap saturation and the screening ratio is derived. Simulation results show significant reductions in both computational complexity and execution times for a wide range of tested scenarios.**

I. Introduction

Sparsity-constrained linear regression has found numerous applications in signal processing and machine learning, tackling under-determined inverse problems. These appear in many forms such as image or audio inpainting [1], source localization [2] or spike deconvolution [3], to cite only a few.

There are many computational approaches to estimate the sparse coefficient vector in such settings with two main families: greedy algorithms [4] [5], and convex optimization approaches relying on ℓ_1-norm minimization, where the regression problem is expressed as an ℓ_1 regularized optimization problem –known as Basis Pursuit [6] or LASSO [7]– which solution is computed using iterative convex optimization algorithms such as ISTA or its variants [8]–[12].

For very high-dimensional problems, however, iterative algorithms to solve ℓ_1 minimization problems can become computationally prohibitive, which is why accelerating techniques are still an intense research topic. This paper demonstrates how to combine two such techniques:

1) Fast structured operators [13]–[17] which provide faster matrix-vector products (see Section III-C);
2) Safe screening tests [18]–[23], which safely eliminate unused explanatory variables (see Section II).

This paper extends the results in [24] [25]. In [24] we introduced a safe screening sphere test that manipulates an approximate dictionary and in [25] the GAP Safe sphere test [21] was extended to this new setting. Here, we revisit the previous results under a broader formalism and propose a fast algorithm for ℓ_1-minimization which combines safe screening and (potentially multiple) fast approximate dictionaries.

A. Sparsity constrained regularization

Let N and K be respectively the dimension of the observed vector and that of the unknown coefficient vector. The observed vector is denoted $\mathbf{y} \in \mathbb{R}^N$ and modeled as $\mathbf{y} \approx \mathbf{A}\mathbf{x}_0$ where $\mathbf{x}_0 \in \mathbb{R}^K$ is sparse and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_K] \in \mathbb{R}^{N \times K}$. In the context of linear inverse problems, $\mathbf{A}$ is the so-called measurement matrix [26], while for sparse signal representations $\mathbf{A}$ would be the dictionary matrix [6], and in statistics the design matrix [7]. In this paper, we adopt the terminology of sparse signal representations hence $\mathbf{A}$ is called a dictionary and its columns, denoted $\mathbf{a}_j \in \mathbb{R}^N$, are called atoms.

The ℓ_1-regularized least squares, referred to as Lasso or Basis Pursuit, consists in finding a sparse coefficient vector $\mathbf{x} \in \mathbb{R}^K$, solution of the following optimization problem:

$$\mathcal{L}(\lambda, \mathbf{A}, \mathbf{y}): \quad \mathbf{x}^\star = \underset{\mathbf{x}}{\operatorname{argmin}} \underbrace{\frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1}_{P(\mathbf{x}|\mathbf{A})} \tag{1}$$

where $P(\mathbf{x}|\mathbf{A})$ is called the primal objective and the parameter $\lambda > 0$ controls the trade-off between data fidelity and sparsity of the solution. We suppose that $\lambda \leq \lambda_{\max} := \|\mathbf{A}^T\mathbf{y}\|_\infty$ (2) since otherwise $\mathbf{x}^\star = \mathbf{0} \in \mathbb{R}^K$ is the unique solution.

B. Iterative algorithms

First-order iterative algorithms, especially proximal-gradient methods (ISTA [8], FISTA [9], TwIST [10], SpaRSA [11], Chambolle-Pock [12]), are popular approaches for solving (1).

One can use an abstract notations to represent the update step for a generic iterative algorithm for the problem $\mathcal{L}(\lambda, \mathbf{A}, \mathbf{y})$: $\{\mathbf{x}_{t+1}, \boldsymbol{\alpha}_{t+1}\} \leftarrow p(\mathbf{x}_t, \mathbf{A}, \boldsymbol{\alpha}_t)$ where $\mathbf{x}_t$ is the current estimate of the primal variable and $\boldsymbol{\alpha}_t$ is a list of updated auxiliary scalars (e.g. the gradient step-size, and possibly a few previous estimates of the primal variable). For instance, the ISTA algorithm is given by

$$\mathbf{x}_{t+1} \leftarrow p(\mathbf{x}_t, \mathbf{A}, L_t) = \mathrm{ST}_{\frac{\lambda}{L_t}}\left(\mathbf{x}_t + \frac{1}{L_t}\mathbf{A}^T(\mathbf{y} - \mathbf{A}\mathbf{x}_t)\right) \tag{3}$$

where $\mathrm{ST}_u(z) = \operatorname{sign}(z)(|z| - u)_+$ denotes the soft-thresholding operation, which is the proximal operator associated to the ℓ_1-norm.

C. Computational bottlenecks

The main bottleneck of existing iterative algorithms in terms of computational complexity is the cost of the required matrix-vector products involving the dictionary matrix, which dominates the overall iteration cost. For example, ISTA requires two matrix-vector multiplications at each iteration (or at least one if the Gram matrix $\mathbf{A}^T\mathbf{A}$ is pre-computed and stored).

The authors are with PANAMA group, Univ Rennes, Inria, CNRS, IRISA, Campus de Beaulieu, 35042 Rennes cedex, France (e-mail: cassio.fraga-dantas@inria.fr).

D. Addressing the computational bottlenecks

A popular way to address this limitation is to constrain the dictionary matrix to a certain type of structure which would allow for fast matrix-vector products. Another approach is to use screening rules to identify a set indexing zero components of the solution $\mathbf{x}^\star$ *before even computing it*.

1) Acceleration with structured dictionaries: Different types of structure, more or less suited to a given application, can be imagined. In this kind of work, there is always a compromise between the flexibility (generality) of the structure and the provided speedup.

In [13], the authors constrain the dictionary to be the product of a fixed *base dictionary* that has a fast implementation (a DCT, for instance) and a column sparse matrix. This model is extended in [16] by replacing the base dictionary with cropped wavelets, which have a certain degree of adaptability. A more general model, which has the two previous ones as particular cases, was proposed in [15]. The dictionary is constrained to be the product of several sparse matrices.

Another interesting line of research, particularly suited to multi-dimensional signals, consists in finding dictionaries formed as the Kronecker (or tensorial) product of sub-dictionaries of smaller size. It was proposed in [27] for the Kronecker product of two sub-dictionaries aiming 2D signals (like images). The model was then extended to a sum of Kronecker products in [17] and [28] (respectively for 2-dimensional and higher-order tensorial data), increasing its flexibility in exchange of some additional complexity.

In practical applications, the dictionary matrix $\mathbf{A}$ is not necessarily structured. A possible strategy is to replace certain iterations $\mathbf{x}_{t+1} = p(\mathbf{x}_t, \mathbf{A}, \boldsymbol{\alpha}_t)$ with $\mathbf{x}_{t+1} = p(\mathbf{x}_t, \tilde{\mathbf{A}}, \boldsymbol{\alpha}_t)$ where $\tilde{\mathbf{A}}$ is a structured approximation of $\mathbf{A}$, i.e.,

$$\mathbf{A} = \tilde{\mathbf{A}} + \mathbf{E} \tag{4}$$

with approximation error $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_K] \in \mathbb{R}^{N \times K}$.

2) Acceleration with screening rules: For a given instance $\mathcal{L}(\lambda, \mathbf{A}, \mathbf{y})$ of (1), characterized by a regularization parameter λ, a dictionary $\mathbf{A}$, and an input vector $\mathbf{y}$, the safe screening approach, introduced by [18], consists in identifying and removing from the dictionary a subset of atoms which are guaranteed to have zero weight in a solution $\mathbf{x}^\star$, before solving the problem. By removing these so-called *inactive atoms* a more compact and readily solved problem is obtained, with decreased matrix-vector multiplication cost, while not affecting at all the original solution which can be obtained by simply zero-padding its restricted version.

Put differently, safe screening is a feature selection technique for a given instance of (1). However, unlike other previous feature selection heuristics [29], [30] based on correlation measures between the atoms $\mathbf{a}_j$ and the input signal $\mathbf{y}$, safe screening has zero risk of false rejections. Moreover, screening techniques are transparent to the underlying ℓ_1 solver and can be readily combined with almost any existing solvers.

Basically, three categories of screening rules can be distinguished: 1) Static; 2) Dynamic; 3) Sequential. The two first categories being mutually exclusive, but not the third one.

The earliest safe screening techniques [18] [19] [31], currently classified as *static rules*, where designed to be applied once and for all before starting the optimization. In contrast, the more recent *dynamic rules*, introduced in [20] and followed by [21], are repeatedly applied during the iterative optimization algorithm leveraging its current solution estimate and gradually increasing the set of rejected atoms.

Sequential rules [19], [32]–[34] exploit the fact that (1) is commonly solved over a grid of regularization parameters λ_i and reuse the results from a previous configuration λ_{i-1} to improve the screening performance.

E. Contributions

First, we introduce a formalism for defining safe screening tests which are robust to approximation errors on the dictionary matrix given that an error bound is provided. The resulting tests are called *stable screening* tests. The stable tests proposed here are general and apply to any context in which the atoms are known up to a certain error margin. The source of this error can be manifold, but we will focus on the case where it is a side effect of manipulating structured approximations of the true dictionary matrix. The proposed framework is also general with respect to the form of the safe region (be it a sphere, dome, etc) as well as to that of the error bound. Here, again, we stick to ℓ_p-balls, which allows us to exemplify the stable tests on some existing ℓ_2-sphere tests –one of which being the state-of-the-art GAP Safe [21]– for a general ℓ_q-ball error bound.

In a second part, we exploit the proposed stable screening in a fast algorithm for ℓ_1-minimization problems, by making use of fast structured approximations of the problem's dictionary matrix. It consists in starting the iterative optimization by using a coarser (but faster) version of the true dictionary and, as approaching convergence, finer approximations are progressively adopted until eventually the original dictionary takes over. Choosing an appropriate moment to switch to a more precise dictionary is crucial in the proposed algorithm. A robust switching criterion based on both the duality gap saturation and the screening ratio is proposed.

F. Related Work

Apart from the aforementioned structured dictionaries and safe screening tests –as well as other preceding correlation-based feature selection heuristics [29], [30]– some related acceleration strategies for sparsity-inducing optimization problems (or more specifically for problem (1)) are worth citing.

Instead of starting from the full problem and pruning the feature set, working set techniques [35], [36] start with small restricted problems and progressively include more promising features. In [37], the authors combine a working set strategy with safe screening and in [38] they incorporate a dual extrapolation technique to further enhance the screening performance and accelerate convergence. This idea is conceivably complementary to the techniques proposed here.

Joint screening [39] allows to screen many atoms which lie close together in one single test, reducing the number of required tests for a given dictionary. Interestingly, the resulting tests share many similarities and mathematical connections to the stable screening tests introduced here, despite arising from an essentially different premise.

G. Outline of the Paper

This paper is organized as follows. In Section II we briefly review standard (static and dynamic) safe screening rules. *Stable* safe screening rules are introduced in Section III. A complete description of the proposed algorithm is given in Section IV and experimental results in Section V.

II. Reminders on Safe Screening

Given a set $\mathcal{A}$ of non-repeating integers, $\mathbf{A}_{[\mathcal{A}]} := [\mathbf{a}_i]_{i\in\mathcal{A}}$ denotes a sub-matrix of $\mathbf{A}$ composed of the columns indexed by the elements in $\mathcal{A}$. The notation extends to vectors: $\mathbf{x}_{[\mathcal{A}]} := [\mathbf{x}(i)]_{i\in\mathcal{A}}$, where $\mathbf{x}(i)$ denotes the i-th component of $\mathbf{x}$. Screening rules allow to identify such sets of reduced cardinality $|\mathcal{A}_t| < K$, so that the complexity of applying $\mathbf{A}_{[\mathcal{A}]}$ or its transpose is reduced compared to that of applying $\mathbf{A}$, while ensuring that $\mathcal{A}_t$ contains the support of the solution $\{i : \mathbf{x}^\star(i) \neq 0\}$. This is achieved using the dual of (1):

$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta}\in\Delta_{\mathbf{A}}}{\operatorname{argmax}} \ \underbrace{\frac{1}{2}\|\mathbf{y}\|_2^2 - \frac{\lambda^2}{2}\left\|\boldsymbol{\theta} - \frac{\mathbf{y}}{\lambda}\right\|_2^2}_{D(\boldsymbol{\theta})} \tag{5}$$

where

$$\Delta_{\mathbf{A}} = \{\boldsymbol{\theta}\in\mathbb{R}^N : \ \|\mathbf{A}^T\boldsymbol{\theta}\|_\infty \leq 1\} \tag{6}$$

is the dual feasible set and $D(\boldsymbol{\theta})$ is the dual objective. The dual and primal solutions ($\boldsymbol{\theta}^\star$ and $\mathbf{x}^\star$) are linked through the relation $\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \lambda\boldsymbol{\theta}^\star$. Optimality conditions (KKT) at the dual solution $\boldsymbol{\theta}^\star$ read

$$\mathbf{a}_j^T\boldsymbol{\theta}^\star = \begin{cases} \operatorname{sign}(\mathbf{x}^\star(j)) & \text{if} \quad \mathbf{x}^\star(j) \neq 0 \\ \gamma \in [-1,1] & \text{if} \quad \mathbf{x}^\star(j) = 0 \end{cases}, \quad \forall j \in \{1,\ldots,K\}.$$

Hence, every dictionary atom for which $|\mathbf{a}_j^T\boldsymbol{\theta}^\star| < 1$ is inactive.

A. Notion of safe region

Since the optimal solution $\boldsymbol{\theta}^\star$ of the dual problem (5) is not known, the inner products $\mathbf{a}_j^T\boldsymbol{\theta}^\star$ cannot be evaluated. Fortunately, given only $\mathbf{A}$ and $\mathbf{y}$, it is possible to identify at a moderate computational cost a region $\mathcal{R} \subset \mathbb{R}^N$, called *safe region*, which is guaranteed to contain the optimal $\boldsymbol{\theta}^\star$.

Definition 1 (Safe region)**.** *A region $\mathcal{R} \subset \mathbb{R}^N$ is safe (with respect to the dual problem* (5)*) if and only if it contains the dual solution $\boldsymbol{\theta}^\star$, i.e. $\boldsymbol{\theta}^\star \in \mathcal{R}$.*

B. Screening an atom given a safe region

Consider an atom $\mathbf{a}_j$. If the inequality $|\mathbf{a}_j^T\boldsymbol{\theta}| < 1$ holds for all $\boldsymbol{\theta} \in \mathcal{R}$ where $\mathcal{R}$ is a safe region, then the above analysis ensures that $\mathbf{a}_j$ is inactive. This gives rise to a *screening test.*

Definition 2 (Screening test for an atom)**.** *Given a region $\mathcal{R}$, a screening test for the atom $\mathbf{a} \in \mathbb{R}^N$ is given by:*

$$\mu(\mathbf{a}|\mathcal{R}) = \sup_{\boldsymbol{\theta}\in\mathcal{R}} |\mathbf{a}^T\boldsymbol{\theta}| \tag{7}$$

A sufficient condition for an atom $\mathbf{a}_j$ to be inactive can be expressed as follows: if $\mathcal{R}$ is safe then

$$\mu(\mathbf{a}_j|\mathcal{R}) < 1 \quad \Longrightarrow \quad \mathbf{x}^\star(j) = 0.$$

In practice, for each atom $\mathbf{a}_j$, computing the test $\mu(\mathbf{a}_j|\mathcal{R})$ allows to eliminate or not the atom.

Formally, given a safe region, the atoms can be partitioned into a preserved set $\mathcal{A} = \{j \in \{1,\ldots,K\} : \mu(\mathbf{a}_j|\mathcal{R}) \geq 1\}$ and its complement, the rejection set $\mathcal{A}^{\mathsf{c}}$, that gathers the indices of the identified inactive atoms.

In practice, safe regions have simple parameterized shapes which parameters need to be identified with moderate computations from the only knowledge of $\mathbf{A}$, $\mathbf{y}$, and possibly the current iterate $\mathbf{x}_t$ of an iterative algorithm addressing (1).

The two most common shapes of safe regions in the literature are spheres and domes (i.e. intersection between a sphere and one or more half-planes).

This paper focuses on sphere tests. Assume we are given a safe region $\mathcal{R}$ which is a closed ℓ_p-ball with center $\mathbf{c}$ and radius R, denoted $B_p(\mathbf{c},R) = \{\mathbf{z} : \|\mathbf{z}-\mathbf{c}\|_p \leq R\}$. The screening test for this region has a closed form

$$\mu(\mathbf{a}|B_p(\mathbf{c},R)) = |\mathbf{a}^T\mathbf{c}| + R\|\mathbf{a}\|_{p^*} \tag{8}$$

where $\|\cdot\|_{p^*}$ denotes the dual norm associated to the p-norm, with $\frac{1}{p} + \frac{1}{p^*} = 1$. For simplicity, we omit subscripts for the ℓ_2-ball ($p = p^* = 2$) denoted $B(\mathbf{c},R)$.

C. Construction of a safe spherical region

A safe region should be as small as possible (to maximize the screening effect) while requiring as little computational overhead as possible. In light of (8), as $\|\mathbf{a}\|_{p^*}$ can be precomputed for each atom, the computational overhead of a screening test with a spherical region is governed by the cost of computing the radius R and the inner products $\mathbf{a}^T\mathbf{c}$ for all atoms that have not been screened out yet. This calls for techniques where the choice of R and $\mathbf{c}$ allows to reuse either quantities that have already been computed along previous iterations of the optimization algorithm.

The construction of the first (static) safe region, obtained in [18], follows from the simple observation that the solution $\boldsymbol{\theta}^\star$ of the dual problem (5) is the Euclidean projection of $\mathbf{y}/\lambda$ on the feasible set $\Delta_{\mathbf{A}}$. As a result, if some feasible point $\boldsymbol{\theta}_F \in \Delta_{\mathbf{A}}$ is known, then $\boldsymbol{\theta}^\star$ can't be further away from $\mathbf{y}/\lambda$ than $\boldsymbol{\theta}_F$ in the ℓ_2 sense, i.e. $\|\boldsymbol{\theta}^\star - \mathbf{y}/\lambda\|_2 \leq \|\boldsymbol{\theta}_F - \mathbf{y}/\lambda\|_2 \ \forall \boldsymbol{\theta}_F \in \Delta_{\mathbf{A}}$.

This leads to an ℓ_2-spherical safe region ($p = p^* = 2$),

$$\mathcal{R} = B(\mathbf{c} = \mathbf{y}/\lambda, R = \|\boldsymbol{\theta}_F - \mathbf{y}/\lambda\|_2), \tag{9}$$

which full specification requires a feasible point $\boldsymbol{\theta}_F \in \Delta_{\mathbf{A}}$.

To generate a feasible point $\boldsymbol{\theta}_F \in \Delta_{\mathbf{A}}$, one could compute the Euclidean projection of any given point $\mathbf{z} \in \mathbb{R}^N$ over the closed convex set $\Delta_{\mathbf{A}}$. As this is too computationally demanding, simpler scaling approaches are preferred: given $\mathbf{z} \in \mathbb{R}^{\mathrm{N}}$ we determine a scaling factor α so that $\Theta(\mathbf{z}|\mathbf{A}) := \alpha\mathbf{z} \in \Delta_{\mathbf{A}}$.

By definition of $\Delta_{\mathbf{A}}$, if $\|\mathbf{A}^T\mathbf{z}\|_\infty \leq 1$ no scaling is needed, otherwise one only needs to divide $\mathbf{z}$ by $\|\mathbf{A}^T\mathbf{z}\|_\infty$. This yields

$$\Theta(\mathbf{z}|\mathbf{A}) := \frac{\mathbf{z}}{\max\left(1, \|\mathbf{A}^T\mathbf{z}\|_\infty\right)}. \tag{10}$$

which is called *dual scaling*.

While dual scaling selects a feasible point to build a safe sphere (9) given an arbitrary $\mathbf{z}$, it neglects the fact that one would like to minimize the *radius* of this sphere to maximize the effect of screening. This calls for a more refined approach.

Proposition 1. *Denoting* $\mathrm{proj}_{\mathbf{z}} \frac{\mathbf{y}}{\lambda} = \frac{\mathbf{y}^T\mathbf{z}}{\lambda\|\mathbf{z}\|_2^2}\mathbf{z}$ *the Euclidean projection of* $\mathbf{y}/\lambda$ *onto the direction of* $\mathbf{z}$*, we have*

$$\Theta\left(\mathrm{proj}_{\mathbf{z}}\frac{\mathbf{y}}{\lambda} \mid \mathbf{A}\right) = \underset{\mathbf{z}'=\alpha\mathbf{z}}{\mathrm{argmin}} \|\mathbf{z}' - \mathbf{y}/\lambda\|_2 \quad \textit{s.t.} \ \|\mathbf{A}^T\mathbf{z}'\|_\infty \leq 1$$

i.e. among all points which are both dual feasible and proportional to $\mathbf{z}$*,* $\Theta\left(\mathrm{proj}_{\mathbf{z}}\frac{\mathbf{y}}{\lambda} \mid \mathbf{A}\right)$ *is the one closest to* $\mathbf{y}/\lambda$*.*

To obtain dynamic spherical regions, we need dual feasible points that change over iterations ($\boldsymbol{\theta}_F = \boldsymbol{\theta}_t$ at iteration t) and leverage the current state of the optimization algorithm. Following [20], this can be achieved by defining $\boldsymbol{\theta}_t$ proportional to the current residual $\boldsymbol{\rho}_t := \mathbf{y} - \mathbf{A}\mathbf{x}_t$, since the dual solution $\boldsymbol{\theta}^\star$ is proportional to $\mathbf{y} - \mathbf{A}\mathbf{x}^\star$. Taking $\mathbf{z} = \boldsymbol{\rho}_t$ and $\boldsymbol{\theta}_t = \Theta\left(\mathrm{proj}_{\boldsymbol{\rho}_t}\frac{\mathbf{y}}{\lambda} \mid \mathbf{A}\right)$ yields

$$\boldsymbol{\theta}_t = \left[\frac{\mathbf{y}^T\boldsymbol{\rho}_t}{\lambda\|\boldsymbol{\rho}_t\|_2^2}\right]_{-\alpha_t}^{\alpha_t} \boldsymbol{\rho}_t, \quad \text{with } \alpha_t = (\|\mathbf{A}^T\boldsymbol{\rho}_t\|_\infty)^{-1} \qquad (11)$$

where $[z]_a^b := \min(\max(z,a),b)$ denotes the projection of the scalar z onto the segment $[a,b]$.

The above principles yield existing safe spherical regions.

1) Static Safe sphere [18]: With $\mathbf{z} = \mathbf{y}/\lambda$ in (10) one gets $\theta_F = \Theta(\mathbf{y}/\lambda|\mathbf{A}) = \mathbf{y}/\lambda_{\max}$, and (9) reads $\mathcal{R} = B(\mathbf{c}, R)$ with

$$\text{Static Safe:} \quad \mathbf{c} = \mathbf{y}/\lambda, R = |\tfrac{1}{\lambda_{\max}} - \tfrac{1}{\lambda}| \cdot \|\mathbf{y}\|_2. \qquad (12)$$

2) Dynamic Safe sphere [20]: With the dual feasible point defined in (11), (9) yields $\mathcal{R} = B(\mathbf{c}, R)$ with

$$\text{Dynamic Safe:} \quad \mathbf{c} = \mathbf{y}/\lambda, R = \|\boldsymbol{\theta}_t - \mathbf{y}/\lambda\|_2. \qquad (13)$$

3) GAP Safe sphere [21]: Given any primal-dual feasible pair $(\mathbf{x}_t, \boldsymbol{\theta}_t) \in \mathbb{R}^K \times \Delta_{\mathbf{A}}$ (say the solution estimations at iteration t) and denoting

$$G(\mathbf{x}_t, \boldsymbol{\theta}_t|\mathbf{A}) := P(\mathbf{x}_t|\mathbf{A}) - D(\boldsymbol{\theta}_t) \qquad (14)$$

the corresponding duality gap, the GAP Safe sphere is $\mathcal{R} = B(\mathbf{c}, R)$ with

$$\text{GAP Safe:} \quad \mathbf{c} = \boldsymbol{\theta}_t, R = \frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t, \boldsymbol{\theta}_t|\mathbf{A})}. \qquad (15)$$

This region is provably safe for any dual feasible point $\boldsymbol{\theta}_t \in \Delta_{\mathbf{A}}$ [21, Theorem 2]. The authors, following [20], propose to use $\boldsymbol{\theta}_t$ defined by (11).

Note that the cited safe regions reuse quantities computed along the iterations of classical proximal algorithms such as the residual $\boldsymbol{\rho}_t$, the inner products $\mathbf{A}^T\boldsymbol{\rho}_t$, or the duality gap (which typically serves as a stopping criterion).

III. Stable Safe Screening

When iterating with $\tilde{\mathbf{A}}$ instead of $\mathbf{A}$, the computational overhead of accessing inner products $\mathbf{a}^T\mathbf{c}$ is no longer moderate, hence the screening techniques reviewed in Section II cannot be directly applied in this context. A possibility would of course be to recompute residuals / inner products / duality gaps associated to $\mathbf{A}$ (instead of $\tilde{\mathbf{A}}$) to implement the above screening tests, but the whole purpose of re-using quantities already computed with a structured (computationally efficient) matrix $\tilde{\mathbf{A}}$ would be lost.

In this section, we propose an alternative solution by deriving screening rules which remain safe w.r.t the original problem (1) *even when manipulating an approximate version of the dictionary*. These *stable* screening rules only requires some knowledge on the magnitude of certain approximation errors between $\tilde{\mathbf{A}}$ and $\mathbf{A}$.

A. Screening a zone *given a safe region*

In conventional screening tests (Definition 2) the dual solution $\boldsymbol{\theta}^\star$ is assumed to belong to a known safe region $\mathcal{R}$. This allows to screen (or not) an atom $\mathbf{a}$ that is perfectly known, (re)using computations of inner products *with this atom.*

When iterating with $\tilde{\mathbf{A}}$, efficient screening tests are only entitled to reuse inner products with approximated atoms, $\tilde{\mathbf{A}}^T\mathbf{c}$. Yet, they should be able to screen certain atoms $\mathbf{a}$, using the only knowledge that these are "close" to the corresponding approximated atoms $\tilde{\mathbf{a}}$.

One way to capture this knowledge is to consider $\mathcal{S} \subset \mathbb{R}^N$ some neighborhood of $\mathbf{a}$, assumed to be known. We will call $\mathcal{S}$ a *zone*. This gives rise to screening tests for zones.

Definition 3 (Screening test for a zone)**.** *Given a safe region* $\mathcal{R}$*, a screening test for the zone* $\mathcal{S} \subset \mathbb{R}^N$ *is given by:*

$$\mu(\mathcal{S}|\mathcal{R}) := \sup_{\mathbf{a}\in\mathcal{S}} \mu(\mathbf{a}|\mathcal{R}) = \sup_{\mathbf{a}\in\mathcal{S}} \sup_{\boldsymbol{\theta}\in\mathcal{R}} |\mathbf{a}^T\boldsymbol{\theta}| \qquad (16)$$

A sufficient condition for an atom $\mathbf{a}_j$ to be inactive can be expressed as follows: if $\mathcal{R}$ is safe and if $\mathbf{a}_j \in \mathcal{S}$, then

$$\mu(\mathcal{S}|\mathcal{R}) < 1 \implies \mathbf{x}^\star(j) = 0.$$

B. Stable *screening tests*

Given an approximate dictionary $\tilde{\mathbf{A}}$, error bounds ϵ_j such that $\|\tilde{\mathbf{a}}_j - \mathbf{a}_j\|_q \leq \epsilon_j$, $1 \leq j \leq K$ can be pre-computed, yielding spherical zones $\mathcal{S}_j := B_q(\tilde{\mathbf{a}}_j, \epsilon_j)$ known to contain the atoms: $\mathbf{a}_j \in \mathcal{S}_j$. Using (8), a spherical zone

$$\mathcal{S} := B_q(\tilde{\mathbf{a}}, \epsilon) \qquad (17)$$

gives rise to the following stable sphere tests:

$$\mu(\mathcal{S}|B_p(\mathbf{c}, R)) = |\tilde{\mathbf{a}}^T\mathbf{c}| + \epsilon\|\mathbf{c}\|_{q^*} + R\|\tilde{\mathbf{a}}\|_{p^*} + CR\epsilon. \qquad (18)$$

with $C = C_{p,q} := N^{\left(1-\frac{1}{p}-\frac{1}{q}\right)_+}$.

In practice, the norms $a_j = \|\mathbf{a}_j\|_{p^*}$ can be pre-computed and stored with negligible overhead, yielding restricted spherical zones

$$\mathcal{S}' := \mathcal{S} \cap \{\mathbf{a} : \|\mathbf{a}\|_{p^*} = a\}. \qquad (19)$$

For such zones we get

$$\mu(\mathcal{S}'|B_p(\mathbf{c}, R)) = |\tilde{\mathbf{a}}^T\mathbf{c}| + \epsilon\|\mathbf{c}\|_{q^*} + Ra. \qquad (20)$$

Lemma 1. *Consider a dictionary* $\mathbf{A}$ *and* $\mathcal{R} = B_p(\mathbf{c}, R)$ *a safe region with respect to the dual problem* (5)*. Let* $\tilde{\mathbf{A}}$ *be an approximate dictionary,* $1 \leq j \leq K$*, and* $\mathcal{S}_j$ *and* $\mathcal{S}'_j$ *defined by* (17)-(20) *with* $\tilde{\mathbf{a}} = \tilde{\mathbf{a}}_j$*,* $\epsilon \geq \|\tilde{\mathbf{a}}_j - \mathbf{a}_j\|_q$*, and* $a = \|\mathbf{a}_j\|_{p^*}$*.*

- *If* $\mu(\mathcal{S}_j|B_p(\mathbf{c},R)) < 1$ *then* $\mathbf{x}^\star_j = 0$*;*
- *If* $\mu(\mathcal{S}'_j|B_p(\mathbf{c},R)) < 1$ *then* $\mathbf{x}^\star_j = 0$*.*

Remark 1. *In [39], Definition 3 is used in a different context. No approximate atoms are considered and the idea is rather to simultaneously test multiple atoms which eventually lie within a same zone $\mathcal{S}$, reducing the total number of tests performed.*

C. Building a safe region using an approximate dictionary

As in classical screening, we have all ingredients to screen provided we can build a safe region using moderate computational overhead. The object of this section is precisely to adapt the constructions of safe regions from Section II-C, which depend on $\mathbf{A}$, to define new safe regions reusing computations done during the iterations *with* $\tilde{\mathbf{A}}$ *instead of* $\mathbf{A}$.

Remark 2. *Let us emphasize that a safe region is a region $\mathcal{R}$ containing the dual solution $\boldsymbol{\theta}^\star$ of the* original *dual problem* (5) *and* not necessarily *the dual solution of its approximate version with $\Delta_{\tilde{\mathbf{A}}}$ instead of $\Delta_{\mathbf{A}}$. Indeed, although $\mathbf{A}$ is approximated to speed up computations, variable elimination needs to be guaranteed with respect to the* original *problem* (1)*.*

The safe regions from Section II-C are built by determining a feasible dual point $\boldsymbol{\theta}_t \in \Delta_{\mathbf{A}}$. Given an arbitrary $\mathbf{z} \in \mathbb{R}^N$, the function $\Theta(\mathbf{z}|\mathbf{A})$ from (10) would provide such a feasible point, however it cannot be used as its computation requires multiplication by $\mathbf{A}^T$. A naive alternative could be to compute $\Theta(\mathbf{z}|\tilde{\mathbf{A}})$, however while this always belongs to $\Delta_{\tilde{\mathbf{A}}}$ (the feasible set for $\tilde{\mathbf{A}}$) it does not necessarily belong to the desired feasible set $\Delta_{\mathbf{A}}$. This can be fixed using a modified dual scaling approach that we propose to call *stable* dual scaling.

Considering error bounds $\boldsymbol{\epsilon} = (\epsilon_j)_{j=1}^K \in \mathbb{R}^K_+$ we define

$$\Theta'(\mathbf{z}|\tilde{\mathbf{A}},\boldsymbol{\epsilon}) := \frac{\mathbf{z}}{\max\left(1, \max_{1\le j\le K}(|\tilde{\mathbf{a}}_j^T\mathbf{z}| + \epsilon_j\|\mathbf{z}\|_{q^*})\right)} \quad (21)$$

For $\boldsymbol{\epsilon} = \mathbf{0}$ we recover $\Theta(\mathbf{z}|\mathbf{A}) = \Theta'(\mathbf{z}|\mathbf{A},\mathbf{0})$.

Lemma 2. *Assume that $\mathbf{a}_j \in \mathcal{S}_j = B_q(\tilde{\mathbf{a}}_j, \epsilon_j)$ for $1 \le j \le K$. Then, for any $\mathbf{z} \in \mathbb{R}^N$, $\Theta'(\mathbf{z}|\tilde{\mathbf{A}},\boldsymbol{\epsilon}) \in \Delta_{\mathbf{A}} \cap \Delta_{\tilde{\mathbf{A}}}$ is a feasible point w.r.t. both the original dual problem* (5) *and its modified version with $\tilde{\mathbf{A}}$ instead of $\mathbf{A}$.*

The proof is in Appendix A. Analogously to (11), we can now define a dual feasible point proportional to the residual $\tilde{\boldsymbol{\rho}}_t := \mathbf{y} - \tilde{\mathbf{A}}\mathbf{x}_t$ at iteration t, $\boldsymbol{\theta}'_t = \Theta'\left(\mathrm{proj}_{\tilde{\boldsymbol{\rho}}_t}\frac{\mathbf{y}}{\lambda} \mid \tilde{\mathbf{A}},\boldsymbol{\epsilon}\right)$, that is to say more explicitly

$$\boldsymbol{\theta}'_t = \left[\frac{\mathbf{y}^T\tilde{\boldsymbol{\rho}}_t}{\lambda\|\tilde{\boldsymbol{\rho}}_t\|_2^2}\right]_{-\alpha'_t}^{\alpha'_t}\tilde{\boldsymbol{\rho}}_t,$$

$$\text{with } \alpha'_t = \left(\max_{1\le j\le K}\left(|\tilde{\mathbf{a}}_j^T\tilde{\boldsymbol{\rho}}_t| + \epsilon_j\|\tilde{\boldsymbol{\rho}}_t\|_{q^*}\right)\right)^{-1} \quad (22)$$

These principles yield new *stable* safe ℓ_2-spherical regions.

1) Stable Static Safe sphere: The static safe sphere (12) depends on $\lambda_{\max} = \max_j |\mathbf{a}_j^T\mathbf{y}|$. To perform only computations with $\tilde{\mathbf{A}}$, we can reuse (12) with

$$\lambda'_{\max} := \max_j\left(|\tilde{\mathbf{a}}_j^T\mathbf{y}| + \epsilon_j\|\mathbf{y}\|_{q^*}\right) \ge \lambda_{\max}, \quad (23)$$

leading to $\mathcal{R} = B(\mathbf{c}', R')$ with

$$\text{Stable Static Safe:} \quad \mathbf{c}' = \mathbf{y}/\lambda, R' = |\tfrac{1}{\lambda'_{\max}} - \tfrac{1}{\lambda}|\cdot\|\mathbf{y}\|_2. \quad (24)$$

2) Stable Dynamic Safe sphere: we adapt the Dynamic Safe sphere in (13) to our approximate setting by replacing the dual feasible point $\boldsymbol{\theta}_t$ (11) by $\boldsymbol{\theta}'_t$ (22), leading to $\mathcal{R} = B(\mathbf{c}', R')$ with

$$\text{Stable Dynamic Safe:} \quad \mathbf{c}' = \mathbf{y}/\lambda, R' = \|\boldsymbol{\theta}'_t - \mathbf{y}/\lambda\|_2. \quad (25)$$

The fact that $\boldsymbol{\theta}'_t \in \Delta_{\mathbf{A}}$ directly implies that this sphere is safe by definition of the basic ℓ_2-spherical bound (9).

3) Stable GAP Safe sphere: to build a safe sphere $B(\mathbf{c}', R')$ with $\mathbf{c}' = \boldsymbol{\theta}'_t$ (22) instead of $\mathbf{c} = \boldsymbol{\theta}_t$ (11) we need to determine a safe radius R'. We reason as follows:

- $\boldsymbol{\theta}'_t$ is feasible with respect to $\mathbf{A}$, i.e. $\boldsymbol{\theta}'_t \in \Delta_{\mathbf{A}}$.
- A GAP safe sphere (15) would correspond to a safe radius $R = \frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t,\boldsymbol{\theta}'_t|\mathbf{A})} = \frac{1}{\lambda}\sqrt{2(P(\mathbf{x}_t|\mathbf{A}) - D(\boldsymbol{\theta}'_t))}$. This cannot be calculated since $P(\mathbf{x}_t|\mathbf{A})$ depends on $\mathbf{A}$.
- A computable quantity is $G(\mathbf{x}_t,\boldsymbol{\theta}'_t|\tilde{\mathbf{A}}) = P(\mathbf{x}_t|\tilde{\mathbf{A}}) - D(\boldsymbol{\theta}'_t)$. As $\boldsymbol{\theta}'_t$ is also feasible with respect to $\tilde{\mathbf{A}}$, i.e. $\boldsymbol{\theta}'_t \in \Delta_{\tilde{\mathbf{A}}}$ (cf Lemma 2), $G(\mathbf{x}_t,\boldsymbol{\theta}'_t|\tilde{\mathbf{A}}) \ge 0$ is indeed a duality gap associated to the *approximate primal* $P(\mathbf{x}_t|\tilde{\mathbf{A}})$ and its dual. However the sphere $B(\boldsymbol{\theta}'_t, R'')$ with $R'' = \frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t,\boldsymbol{\theta}'_t|\tilde{\mathbf{A}})}$ is not a priori a safe region wrt to the original primal $P(\mathbf{x}_t|\mathbf{A})$.

Therefore, we add a computable security margin to R'' in order to get a radius $R' \ge R$, ensuring that $B(\boldsymbol{\theta}'_t, R')$ is safe. Recall the standard notation $\|\mathbf{M}\|_{p\to q} := \sup_{\|u\|_p\le 1}\|\mathbf{M}\mathbf{u}\|_q$.

Theorem 1 (Stable GAP Safe sphere)**.** *Consider*

$$\mathcal{E} \ge \|\mathbf{A} - \tilde{\mathbf{A}}\|_{r\to 2}$$

where $1 \le r \le \infty$. For any[1] $(\mathbf{x},\boldsymbol{\theta}) \in \mathbb{R}^K \times \Delta_{\mathbf{A}}$*, we have*

$$\boldsymbol{\theta}^\star \in B\left(\mathbf{c}' = \boldsymbol{\theta}, R' = \frac{1}{\lambda}\sqrt{2G(\mathbf{x},\boldsymbol{\theta}|\tilde{\mathbf{A}}) + 2\delta}\right) \quad (26)$$

$$\textit{with} \quad \delta := \|\mathbf{y} - \tilde{\mathbf{A}}\mathbf{x}\|_2\mathcal{E}\|\mathbf{x}\|_r + \frac{1}{2}\mathcal{E}^2\|\mathbf{x}\|_r^2 \quad (27)$$

In other words $B(\boldsymbol{\theta}, R')$ is safe.

The proof was given in [25]. By applying Theorem (1) to $\mathbf{x}_t$ and $\boldsymbol{\theta}'_t$ in (22) we obtain $\mathcal{R} = B(\mathbf{c}', R')$ with

$$\text{Stable GAP Safe:} \quad \mathbf{c}' = \boldsymbol{\theta}'_t, R' = \frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t,\boldsymbol{\theta}'_t|\tilde{\mathbf{A}}) + 2\delta_t}. \quad (28)$$

The first columns of Table I summarize the obtained safe regions alongside with their previously existing analogue.

Combining stable safe sphere regions with stable safe screening tests (cf Section III-B) yields screening rules which remain safe despite manipulating an approximate dictionary (last column of Table I). As before, these new rules also reuse the calculations performed by optimization algorithm with the approximate dictionary $\tilde{\mathbf{A}}$, making them suitable in practice.

[1] In this theorem, $\theta \in \Delta_{\mathbf{A}}$ can be arbitrary. In particular it is not necessarily expressed as in (22), hence it does not need to belong to $\Delta_{\tilde{\mathbf{A}}}$.

[2] Since the center $\mathbf{c} = \mathbf{y}/\lambda$ is independent of the iteration count t, the resulting test can be simplified provided that $\mathbf{A}^T\mathbf{y}$ can be pre-computed.

Sphere test	**Center**	**Radius**	**Test expression** - $\mu(\mathbf{a}\vert B(\mathbf{c},R))$ or $\mu(\mathcal{S}'\vert B(\mathbf{c}',R'))$
Static Safe	$\mathbf{y}/\lambda$	$\vert 1/\lambda_{\max}-1/\lambda\vert\cdot\Vert\mathbf{y}\Vert_2$	$\vert\mathbf{a}^T\mathbf{y}\vert/\lambda+\vert 1/\lambda_{\max}-1/\lambda\vert\cdot\Vert\mathbf{y}\Vert_2\cdot\Vert\mathbf{a}\Vert_2$
Stable Static Safe	$\mathbf{y}/\lambda$	$\vert 1/\lambda'_{\max}-1/\lambda\vert\cdot\Vert\mathbf{y}\Vert_2$	$\vert\tilde{\mathbf{a}}^T\mathbf{y}\vert/\lambda+\epsilon\Vert\mathbf{y}\Vert_{q^*}/\lambda+\vert 1/\lambda'_{\max}-1/\lambda\vert\cdot\Vert\mathbf{y}\Vert_2\cdot\Vert\mathbf{a}\Vert_2$
Dynamic Safe	$\mathbf{y}/\lambda$	$\Vert\boldsymbol{\theta}_t-\mathbf{y}/\lambda\Vert_2$	$\vert\mathbf{a}^T\mathbf{y}\vert/\lambda+\Vert\boldsymbol{\theta}_t-\mathbf{y}/\lambda\Vert_2\cdot\Vert\mathbf{a}\Vert_2$
Stable Dynamic Safe	$\mathbf{y}/\lambda$	$\Vert\boldsymbol{\theta}'_t-\mathbf{y}/\lambda\Vert_2$	$\vert\mathbf{a}^T\mathbf{y}\vert/\lambda+\Vert\boldsymbol{\theta}'_t-\mathbf{y}/\lambda\Vert_2\cdot\Vert\mathbf{a}\Vert_2,$ if $\mathbf{a}^T\mathbf{y}$ is given[2]
			$\vert\tilde{\mathbf{a}}^T\mathbf{y}\vert/\lambda+\epsilon\Vert\mathbf{y}\Vert_{q^*}/\lambda+\Vert\boldsymbol{\theta}'_t-\mathbf{y}/\lambda\Vert_2\cdot\Vert\mathbf{a}\Vert_2,$ otherwise
GAP Safe	$\boldsymbol{\theta}_t$	$\frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t,\boldsymbol{\theta}_t\vert\mathbf{A})}$	$\vert\mathbf{a}^T\boldsymbol{\theta}_t\vert+\frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t,\boldsymbol{\theta}_t\vert\mathbf{A})}\cdot\Vert\mathbf{a}\Vert_2$
Stable GAP Safe	$\boldsymbol{\theta}'_t$	$\frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t,\boldsymbol{\theta}'_t\vert\tilde{\mathbf{A}})+2\delta_t}$	$\vert\tilde{\mathbf{a}}^T\boldsymbol{\theta}'_t\vert+\epsilon\Vert\boldsymbol{\theta}'_t\Vert_{q^*}+\frac{1}{\lambda}\sqrt{2G(\mathbf{x}_t,\boldsymbol{\theta}'_t\vert\tilde{\mathbf{A}})+2\delta_t}\cdot\Vert\mathbf{a}\Vert_2$

TABLE I: Sphere center $\mathbf{c}$ and radius R for Static Safe, Dynamic Safe and GAP Safe ($\mathbf{c}'$ and R' for the stable versions). See definitions of $\lambda_{\max}$ in (2), of $\lambda'_{\max}$ in (23), of $\boldsymbol{\theta}_t$ in (11), of $\boldsymbol{\theta}'_t$ in (22), and of δ_t in (27) with $\mathbf{x}=\mathbf{x}_t$.
Test expressions for $\mu(\mathbf{a}|B(\mathbf{c},R))$ – see (8) – and $\mu(\mathcal{S}'|B(\mathbf{c}',R'))$ – see (20) where $\mathcal{S}'$ is defined in (19) with $a=\|\mathbf{a}\|_2$.

IV. A Fast Algorithm for ℓ_1 regularization

We now have the tools to combine stable safe screening and the use of fast structured dictionaries. The proposed Algorithm 1 consists in incorporating a stable dynamic screening rule to iterations of a proximal algorithm. Besides the vector $\mathbf{y}$ and the regularization parameter λ, the input of the algorithm consists in a sequence $\{\tilde{\mathbf{A}}^i\}_{i=0}^I$ of approximate dictionaries with $\tilde{\mathbf{A}}^I=\mathbf{A}$, and a corresponding sequence of error bounds $\{\boldsymbol{\epsilon}^i\}_{i=0}^I$, where $\boldsymbol{\epsilon}^i=(\epsilon_j^i)_{j=1}^K\in\mathbb{R}_+^K$ contains ℓ_2 error bounds on the atoms of the i-th dictionary $\tilde{\mathbf{A}}^i$, and of course $\boldsymbol{\epsilon}^I=\mathbf{0}$.

In general terms, the proposed strategy consists in gradually switching to more accurate dictionary approximations as the optimization algorithm approaches the solution. Note that any iterative optimization technique can be used by replacing the generic update function $p(\mathbf{x}_t,\mathbf{A},\boldsymbol{\alpha}_t)$ by the corresponding update step, for instance equation (3) for ISTA. A possible variation consists in performing the screening at regular intervals instead of every iteration.

Algorithm 1 $\mathbf{x}^\star=\texttt{FastL1}(\{\tilde{\mathbf{A}}^i\}_{i=0}^I,\{\boldsymbol{\epsilon}^i\}_{i=0}^I,\mathbf{y},\lambda,\Gamma)$

```
Initialize: $t=0$, $\mathbf{x}_0=\mathbf{0}$, $i=0$, $\mathcal{A}_0=\{1,\dots,K\}$,
$\boldsymbol{\alpha}_0$ according to the considered proximal algorithm
while stopping criterion not met do
    —– Restrict to preserved set —–
    $\tilde{\mathbf{A}}\leftarrow\tilde{\mathbf{A}}^i_{[\mathcal{A}_t]}$,    $\mathbf{x}_t\leftarrow(\mathbf{x}_t)_{[\mathcal{A}_t]}$
    —– Update preserved coefficients —–
    $\{\mathbf{x}_{t+1},\boldsymbol{\alpha}_{t+1}\}\leftarrow p(\mathbf{x}_t,\tilde{\mathbf{A}},\boldsymbol{\alpha}_t)$
    —– Dynamic Screening —–
    Set $\boldsymbol{\theta}'_t$ using (22) and $\mathbf{c}'$, $R'$ according to Table I
    $\mathcal{A}_{t+1}\leftarrow\{j\in\mathcal{A}_t:\mu(\mathcal{S}'_j|B(\mathbf{c}',R'))\geq 1\}$
    $t\leftarrow t+1$
    —– Switching criterion —–
    Compute $\gamma_t$ (cf. (29)) and $K_t$ (cf. (39))
    $i=\texttt{SwitchDictionary}(i,I,\gamma_t,\Gamma,K_t)$
```

To fully describe this algorithm we need to specify the stopping and switching criteria, which involve a parameter Γ.

A classical stopping criterion is to stop when the duality gap falls below a threshold, recalling that strong duality holds for problem (1), i.e. $G(\mathbf{x}^\star,\boldsymbol{\theta}^\star|\mathbf{A})=0$ [40]. The simplest way of computing the duality gap is to wait until the algorithm reaches $\tilde{\mathbf{A}}=\tilde{\mathbf{A}}^I=\mathbf{A}$. Alternatively one could upper bound the duality gap using approximate dictionaries, this is left to future work.

A switching criterion has two main motivations:

- **Convergence:** to avoid diverging from the solution of the original problem (1). The higher the error bounds $\boldsymbol{\epsilon}^i$, the higher the risk of moving away from the true solution, therefore the need to switch earlier in such cases.
- **Speed:** once screened enough, the original dictionary becomes faster to apply than the approximate ones. This is crucial at high regularization regimes ($\lambda/\lambda_{\max}\approx 1$) in which screening typically occurs very quickly.

A. Switching: convergence criterion

In [24], [25], we proposed a empirically-tuned heuristic convergence criterion. In the following, we derive a more theoretically grounded criterion.

Given a solution estimate $\mathbf{x}_t$ at iteration t, a natural convergence measure for the problem $\mathcal{L}(\lambda,\mathbf{A},\mathbf{y})$ is the duality gap $G(\mathbf{x}_t,\boldsymbol{\theta}_t|\mathbf{A})$ (14) with $\boldsymbol{\theta}_t$ calculated as in (11). However, when manipulating approximate dictionaries, this quantity is not accessible without unwanted additional computations.

As a surrogate, we propose to use a ratio

$$\gamma_t=\frac{G(\mathbf{x}_t,\tilde{\boldsymbol{\theta}}_t|\tilde{\mathbf{A}})}{G(\mathbf{x}_t,\boldsymbol{\theta}'_t|\tilde{\mathbf{A}})}\tag{29}$$

between two computable gaps arising from two different dual points $\boldsymbol{\theta}'_t$ and $\tilde{\boldsymbol{\theta}}_t$ that are respectively feasible for problems $\mathcal{L}(\lambda,\mathbf{A},\mathbf{y})$ and $\mathcal{L}(\lambda,\tilde{\mathbf{A}},\mathbf{y})$:

$$\boldsymbol{\theta}'_t\in\Delta_{\mathbf{A}}\cap\Delta_{\tilde{\mathbf{A}}}\longrightarrow G(\mathbf{x}_t,\boldsymbol{\theta}'_t|\tilde{\mathbf{A}})=P(\mathbf{x}_t|\tilde{\mathbf{A}})-D(\boldsymbol{\theta}'_t)\tag{30}$$

$$\tilde{\boldsymbol{\theta}}_t\in\Delta_{\tilde{\mathbf{A}}}\longrightarrow G(\mathbf{x}_t,\tilde{\boldsymbol{\theta}}_t|\tilde{\mathbf{A}})=P(\mathbf{x}_t|\tilde{\mathbf{A}})-D(\tilde{\boldsymbol{\theta}}_t)\tag{31}$$

where $\boldsymbol{\theta}'_t\in\Delta_{\mathbf{A}}\cap\Delta_{\tilde{\mathbf{A}}}$ has been previously calculated for the stable screening with (22), while $\tilde{\boldsymbol{\theta}}_t\in\Delta_{\tilde{\mathbf{A}}}$ is the conventional dual point (11) but calculated with $\tilde{\mathbf{A}}$. Then again, apart from

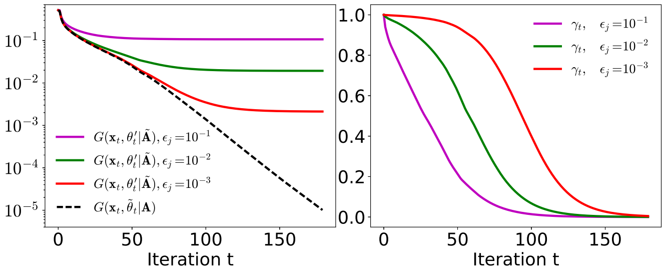


Fig. 1: Typical behaviour of duality gaps (left) and ratio γ_t (right) as a function of the iteration number and error bounds.

minor extra memory requirements, the computation of the quantities in (31) reuses most of the calculations[3] in (30).

If $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}^i$ was kept until convergence, we would have

$$\tilde{\mathbf{x}}_t \to \tilde{\mathbf{x}}^\star \qquad \text{and} \qquad \tilde{\boldsymbol{\theta}}_t \to \tilde{\boldsymbol{\theta}}^\star. \tag{32}$$

with $\tilde{\mathbf{x}}^\star$ a solution of $\mathcal{L}(\lambda, \tilde{\mathbf{A}}, \mathbf{y})$ and $\tilde{\boldsymbol{\theta}}^\star$ a solution of the corresponding dual problem. Then, as illustrated in Fig. 1 (left), the second quantity $G(\mathbf{x}_t, \tilde{\boldsymbol{\theta}}_t|\tilde{\mathbf{A}})$ –which is precisely the duality gap w.r.t. the problem $\mathcal{L}(\lambda, \tilde{\mathbf{A}}, \mathbf{y})$– would tend to zero, while the first quantity $G(\mathbf{x}_t, \boldsymbol{\theta}'_t|\tilde{\mathbf{A}})$ would typically saturate at a nonzero value since $\boldsymbol{\theta}'_t \nrightarrow \tilde{\boldsymbol{\theta}}^\star$. As a result, the ratio γ_t would shrink with the iterations as illustrated in Fig. 1 (right).

With that in mind, we propose to switch dictionaries when

$$\gamma_t \leq \Gamma \tag{33}$$

where the higher Γ, the greater sensibility to the gap saturation.

B. Speed criterion

In terms of speed, the general idea is to exploit tradeoffs between approximation and computational efficiency.

Denoting $\mathcal{C}(\mathbf{A}, \mathcal{A})$ the computational cost of multiplying matrix $\mathbf{A}_{[\mathcal{A}]}$ by a vector, it is reasonable to assume that $\mathcal{C}(\mathbf{A}, \mathcal{A}) \leq \mathcal{C}(\mathbf{A}, \mathcal{A}')$ if $\mathcal{A} \subset \mathcal{A}'$. As a result, given the nested structure of the sequence of preserved sets $\mathcal{A}_t$, the sequence $\mathcal{C}(\mathbf{A}, \mathcal{A}_t)$, $t \geq 0$ is non-increasing.

Consider two approximate dictionaries $\tilde{\mathbf{A}}^i$ and $\tilde{\mathbf{A}}^j$ and assume the initial "unscreened" computational costs satisfy $\mathcal{C}(\tilde{\mathbf{A}}^i, \mathcal{A}_0) < \mathcal{C}(\tilde{\mathbf{A}}^j, \mathcal{A}_0)$. Denote $\mathbf{E}^i$, $\mathbf{E}^j$ the corresponding approximation errors (cf (4)), and assume that $\mathbf{E}^i$ is "larger" than $\mathbf{E}^j$ (let us denote this $\mathbf{E}^i \succ \mathbf{E}^j$) in the sense that for each column the approximation error is larger. An irrevocable switching point would be an iteration T in which the coarser approximation $\tilde{\mathbf{A}}^i$ becomes more complex than $\tilde{\mathbf{A}}^j$

$$\forall t > T, \quad \mathcal{C}(\tilde{\mathbf{A}}^i, \mathcal{A}_t) \geq \mathcal{C}(\tilde{\mathbf{A}}^j, \mathcal{A}_t). \tag{34}$$

Note that this only happens if $\mathcal{C}(\tilde{\mathbf{A}}^i, \mathcal{A}_t)$ decreases faster than $\mathcal{C}(\tilde{\mathbf{A}}^j, \mathcal{A}_t)$ as screening progresses (i.e. as $|\mathcal{A}_t|$ gets smaller).

To use the general idea just described, one needs to specify a complexity model $\mathcal{C}(\mathbf{A}, \mathcal{A}_t)$ depending on the structure of the corresponding matrix. For a generic unstructured dictionary matrix $\mathbf{A} \in \mathbb{R}^{N \times K}$, a simple model is:

$$\mathcal{C}(\mathbf{A}, \mathcal{A}_t) = N|\mathcal{A}_t| \tag{35}$$

To quantify the complexity reduction on matrix-vector products entailed by an approximate dictionary we will use the concept of *Relative Complexity* (RC) [15], such that

$$\mathcal{C}(\tilde{\mathbf{A}}, \mathcal{A}_0) = \mathrm{RC}(\tilde{\mathbf{A}}) \times NK. \tag{36}$$

In a worst-case scenario, screening does not further reduce the cost and $\mathcal{C}(\tilde{\mathbf{A}}, \mathcal{A}_t) = \mathrm{RC}(\tilde{\mathbf{A}}) \times NK$. In more optimistic scenarios, some fast approximate dictionary structures might still benefit from speedups upon column removal.

Consider approximate dictionaries $\{\tilde{\mathbf{A}}^0, \ldots, \tilde{\mathbf{A}}^I\}$, with $\tilde{\mathbf{A}}^I = \mathbf{A}$, of increasing precision and decreasing complexity

$$\mathbf{E}^1 \succ \ldots \mathbf{E}^I = \mathbf{0} \;\text{ and }\; \mathrm{RC}(\tilde{\mathbf{A}}^0) < \ldots < \mathrm{RC}(\tilde{\mathbf{A}}^I) = 1. \tag{37}$$

The above observations suggest to switch from $\tilde{\mathbf{A}}^i$ to $\mathbf{A}$ if

$$|\mathcal{A}_t| \leq \mathrm{RC}(\tilde{\mathbf{A}}^i)\; K. \tag{38}$$

Also, as a consequence of complexity model (36) adopted for the approximate dictionaries $\tilde{\mathbf{A}}$, this criterion never causes a switching from $\tilde{\mathbf{A}}^i$ to $\tilde{\mathbf{A}}^j$ with $i < j \neq I$.

C. Heuristic look-ahead speed criterion

The preserved set size $|\mathcal{A}_t|$ used in (38) should ideally be the one associated to the dictionary *after* switching ($\mathbf{A}$ in this case), while in practice only the screening rate associated to the *current* approximation $\tilde{\mathbf{A}}^i$ is available. This motivates the introduction of a heuristic (that preserves the safety of screening) to anticipate the potentially smaller size of the preserved set $|\mathcal{A}_t|$ as soon as we switch to a more precise approximation (or to the original dictionary).

Indeed, stable screening generally leads to less atom eliminations due to the extra security margins (and more so for higher approximation errors). This is illustrated in Figure 2 (left) with a scatterplot in which the x-axis correspond to the "oracle" number of preserved atoms $|\mathcal{A}|$ obtained with a conventional Dynamic Safe screening mechanism having access to $\mathbf{A}$. This is the quantity we are interested in estimating for the speed criterion (38). The y-axis corresponds to the number of preserved atoms obtained by the corresponding stable test for a given approximation $\tilde{\mathbf{A}}$. Each plotted point compares the two mentioned screening ratios at a given iteration. The procedure was repeated a large number of times in order to give a full picture of how the two quantities correlate to one another. The main observation is that all points lie above the identity line which means that the stable test always overestimates (sometimes significantly) the value that $|\mathcal{A}_t|$ will reach after switching. This phenomenon is intensified for higher approximation errors.

As a heuristic, we propose to estimate $|\mathcal{A}_t|$ using the *conventional* test on the approximate atoms, i.e. $\mu(\tilde{\mathbf{a}}_j|\mathcal{R})$.

$$K_t = |\{j \in \{1, \ldots, K\} : \mu(\tilde{\mathbf{a}}_j|\mathcal{R}) \geq 1\}| \tag{39}$$

[3] Only N extra products for computing $\|\tilde{\boldsymbol{\theta}}\|$ and $|\mathcal{A}_t|$ comparisons for $\|\tilde{\mathbf{A}}^T \boldsymbol{\rho}_t\|_\infty$ are required.

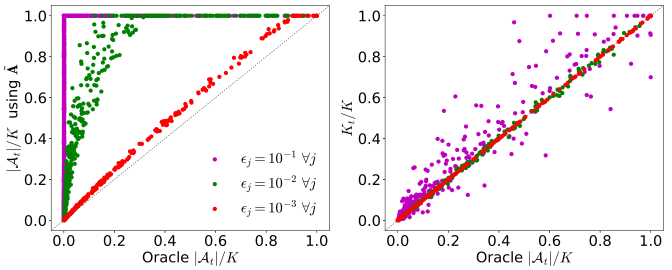


Fig. 2: Number of preserved atoms $|\mathcal{A}|$ using oracle conventional screening (x-axis) *vs* stable screening (y-axis, left plot), or *vs* proposed heuristic (y-axis, right plot). .

As shown in Fig. 2 (right graph), this correlates much better with the oracle value of $|\mathcal{A}_t|$, since it doesn't have the additional security margins used in the stable test. Even though the test used to compute K_t is unsafe with respect to the original problem (1), it has no impact on the safety of screening itself, which is still performed with the stable test. Moreover, the additional test is virtually costless since the same calculations of the first test are used.

D. Summary and example

The resulting switching strategy, shown in Algorithm 2, plays two roles:

(I) Decide when to stop using the current dictionary.
(II) Choose the next dictionary.

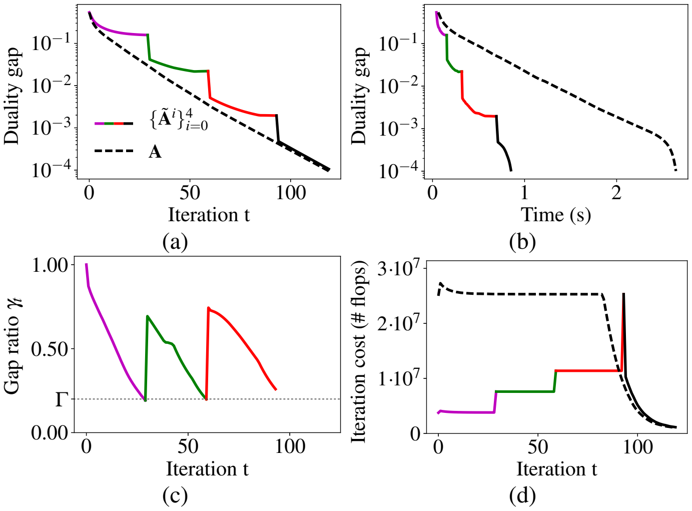
Algorithm 2 $j = \texttt{SwitchDictionary}(i, I, \gamma_t, \Gamma, K_t)$

```
if $K_t \leq \mathrm{RC}(\tilde{\mathbf{A}}^i)\ K$ then          ▷ Speed criterion
    $j \leftarrow I$
else if $\gamma_t \leq \Gamma$ then          ▷ Convergence criterion
    $j \leftarrow i + 1$
else
    $j \leftarrow i$
```

The speed criterion (section IV-B) triggers a switching directly to the original dictionary $\tilde{\mathbf{A}}^I = \mathbf{A}$. We suppose the Relative Complexity associated to the i-th approximation $\mathrm{RC}(\tilde{\mathbf{A}}^i)$ (cf. (36)) to be known and stored in memory. The convergence criterion (section IV-A) switches to the next available approximation $\tilde{\mathbf{A}}^{i+1}$. Note that if $\mathbf{A}$ is already adopted, i.e. input $i = I$, then $\mathbf{A}$ is kept.

Example: Figure 3 shows an example of the proposed algorithm, given a list of approximations $\{\tilde{\mathbf{A}}^i\}_{i=0}^I$ with $I = 4$, decreasing approximation error $\boldsymbol{\epsilon}^i$ and increasing relative complexity $\mathrm{RC}(\tilde{\mathbf{A}}^i) = 0.15(i+1)$, $0 \leq i < I$. As usual, $\tilde{\mathbf{A}}^I = \mathbf{A}$.

Figure 3a shows the duality gap evolution over the iterations with $\Gamma = 0.2$. As soon as the gap $G(\mathbf{x}_t, \boldsymbol{\theta}'_t|\tilde{\mathbf{A}}^i)$ saturates, a more precise approximation is adopted (the next on the list), avoiding to excessively delay the convergence compared to the conventional case where screening is performed with $\mathbf{A}$ from the beginning (dotted curve). Although some delay is introduced in terms of iterations until convergence, it is largely compensated by the fact that the initial iterations with the approximate dictionaries are much faster. This is illustrated in Fig. 3b in which the duality gap is plotted as a function of the execution time.



(a) (b) (c) (d)

Fig. 3: Application example of the proposed algorithm. (a) and (b) Duality gap over the iterations and time respectively. (c) Gap ratio γ_t (cf. (29)) for the convergence switching criterion, with threshold $\Gamma = 0.2$. (d) Computational complexity (flops) per iteration. $N = 2500$, $K = 10000$, $\lambda/\lambda_{max} = 0.2$.

The theoretical complexity per iteration is shown in Fig. 3d. As better approximations are adopted, the iteration complexity rises proportionally to the corresponding RC. Note that as soon as the conventional dictionary becomes faster than the current approximation, it is promptly adopted (by the speed criterion). In this example, the last approximation $\tilde{\mathbf{A}}^{i=3}$ was never used, since we switched directly from $\tilde{\mathbf{A}}^{i=2}$ to $\mathbf{A}$.

E. Complexity Analysis

Existing screening tests introduce only a minor computational overhead because they primarily reuse matrix-vector multiplications either already performed in the optimization algorithm update $p(\mathbf{x_t}, \mathbf{A}, \boldsymbol{\alpha_t})$ (typically the product $\mathbf{A}^T\boldsymbol{\rho}$) or that can be precalculated once for all ($\mathbf{A}^T\mathbf{y}$). The same holds for the stable tests proposed in this article. We now derive the involved number of floating point operations (flops). Since static screening rules represent only a fixed computational overhead, we concentrate on dynamic rules which could potentially (if not properly designed) lead to a significant overhead.

1) Screening cost: The most expensive computations associated to a dynamic screening rule – the ones potentially of the order of a matrix vector product $\mathcal{O}(NK)$ – are:

- Computation of a dual feasible point $\boldsymbol{\theta}_t$ in (11) (resp. $\boldsymbol{\theta}'_t$(22)) requires the product $\mathbf{A}^T\boldsymbol{\rho}$ (resp. $\tilde{\mathbf{A}}^T\tilde{\boldsymbol{\rho}}$) reused from the optimization algorithm.
- Sphere test $\mu(\mathbf{a}|\mathcal{R})$ (resp. $\mu(\mathcal{S}'_j|\mathcal{R})$) requires the product $\mathbf{a}_j^T\mathbf{c}$ (resp. $\tilde{\mathbf{a}}_j^T\mathbf{c}$) for all preserved atoms $j \in \mathcal{A}$, which comes down to the matrix-vector product $\mathbf{A}^T\mathbf{c}$ (resp. $\tilde{\mathbf{A}}^T\mathbf{c}$). Practical sphere regions have been designed to limit this potential overhead. While in the Dynamic Safe sphere it reduces to the precalculated product $\mathbf{A}^T\mathbf{y}$, in

the GAP Safe sphere it reduces to $\mathbf{A}^T\boldsymbol{\rho}$ (resp. $\tilde{\mathbf{A}}^T\tilde{\boldsymbol{\rho}}$) calculated in the optimization iteration.

The other required calculations are detailed in Appendix B. In short, the screening represents a rather low overhead $\mathcal{O}(N+|\mathcal{A}|)$ –even its stable version– compared to the optimization update: $\mathcal{O}(N|\mathcal{A}|)$ with screening or $\mathcal{O}(NK)$ without it, due to matrix-vector products.

2) Full iteration cost: Table IV shows the number of operations of a complete iteration in Algorithm 1 (ISTA update + screening), following [20] and adopting the complexity models in equations (35) and (36) for matrix vector multiplications. We denote $\text{flops}_{\mathbf{A}}(t)$ the cost of iteration t with the conventional screening and $\text{flops}_{\tilde{\mathbf{A}}}(t)$ with the stable screening. As a benchmark, we use the complexity of an ISTA iteration without screening, denoted $\text{flops}_N(t)$.

$\text{flops}_N(t)$	$(K+\Vert\mathbf{x}_t\Vert_0)N+4K+N$
$\text{flops}_{\mathbf{A}}(t)$	$(\vert\mathcal{A}\vert+\Vert\mathbf{x}_t\Vert_0)N+6\vert\mathcal{A}\vert+5N$
$\text{flops}_{\tilde{\mathbf{A}}}(t)$	$(\text{RC}\times K+\Vert\mathbf{x}_t\Vert_0)N+8\vert\mathcal{A}\vert+7N$

TABLE II: Complete iteration complexity

To obtain the total complexity of the algorithm, simply add up all iteration costs calculated with the corresponding active set size $|\mathcal{A}_t|$ and sparsity of the solution estimate $\|\mathbf{x}_t\|_0$.

V. Experiments

In this section we demonstrate the potential of the proposed algorithm in terms of complexity reduction and time saving for ℓ_1-minimization problems. This is done in a wide set of simulation scenarios, summarized in Table III, to evaluate the influence of the main parameters involved.

Problem parameters	**Values**
Regularization λ/λ_{max}	$[10^{-2}, 1]$ (logarithmically-spaced)
Convergence tolerance	$G(\mathbf{x}_t, \boldsymbol{\theta}_t\vert\mathbf{A}) < tol \in [10^{-6}, 10^{-3}]$
Data distribution $\mathbf{y}$	$\mathbf{Ax}$ with $\mathbf{x}$ Bernoulli-Gaussian
Algorithm parameters	**Values**
Optimization algorithm	ISTA, FISTA
Screening Test	(stable) Dynamic, (stable) GAP Safe
Switching threshold Γ	$[0.01, 0.8]$ (logarithmically-spaced)
Speedup-Error tradeoff $\tilde{\mathbf{A}}^i$	3 scenarios (see Section V-A)

TABLE III: Pool of parameters explored in the simulations.

The dimension of the observed vector $\mathbf{y}$ is $N = 2500$ and that of the coefficient vector $\mathbf{x}$ (or, equivalently, the number of atoms in the dictionary) is $K = 10000$. Unit-norm input data samples $\mathbf{y} = \mathbf{Ax}$ are generated from a sparse vector $\mathbf{x}$ with active set determined by a Bernoulli distribution with probability $p = 0.02$ and zero-mean standard Gaussian entries. The reported results are the average on 25 independent and identically distributed realizations of the input $\mathbf{y}$. The dictionary matrix $\mathbf{A} \in \mathbb{R}^{N\times K}$ is generated in such a way that it is more or less efficiently approximated by a fast structured matrix according to three representative scenarios (more details are given in Section V-A).

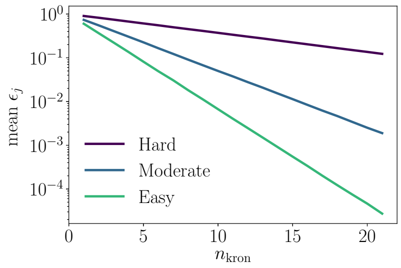


Fig. 4: Simulated scenarios: different trade-offs between approximation error and speedup of the structured dictionaries.

Although other problem dimensions (N and K) and data distributions were explored, we decided to keep these parameters fixed on the reported experiments since they were observed not to decisively influence the analyzed results.

In all figures, the GAP Safe and Dynamic Safe tests are denoted respectively by the acronyms **GAP** and **DST**.

A. Specifying the Fast Structured dictionaries

The performance of the proposed algorithm is directly associated to the *quality* of the available approximations $\tilde{\mathbf{A}}^i$. A good approximation would have both a small application complexity (RC) and small approximation error ($\boldsymbol{\epsilon}$). There is a compromise, however, since each of these features usually come to the price of the other.

In the experiments, we use a particular kind of fast structured dictionaries referred to as *SuKro* [17] which can be written as a sum of Kronecker products $\tilde{\mathbf{A}} = \sum_{k=1}^{n_{\text{kron}}} \mathbf{B}_k \otimes \mathbf{C}_k$. Its reduced multiplication cost comes from the fact that the sub-matrices $\mathbf{B}_k$, $\mathbf{C}_k \in \mathbb{R}^{\sqrt{N}\times\sqrt{K}}$ are much smaller than $\mathbf{A} \in \mathbb{R}^{N\times K}$. The choice of the SuKro structure is justified by the fact that it directly provides a range of speedup-error compromises by tweaking the number of Kronecker terms (n_{kron}) in the sum. A higher n_{kron} provides a more precise approximation although implying a higher RC.

We define three representative simulation scenarios – easy, moderate and hard scenarios – in which the dictionary matrix is poorly, moderately and efficiently approximated by the structured dictionary respectively. In practical terms, this translates to different approximation error decay profiles as a function of the number of Kronecker terms (n_{kron}) on the SuKro structure, as shown in Figure 4.

Although computational complexity associated to a certain SuKro operator can be calculated analytically, we measured the actual time speedup obtained in practice in order to have a more realistic value. Interestingly, the measured RC is lower than the theoretical prediction. [4]

B. Computational complexity and Time results

Let us denote F_N, $F_{\mathbf{A}}$ and $F_{\tilde{\mathbf{A}}}$ respectively the total computational complexity (in number of flops) of the optimization

[4] Some parallelization is introduced (in the summing terms) and matrix-matrix products are faster than the naive cubic complexity.

n_{kron}	**5**	**10**	**15**	**20**
Measured RC	0.08	0.15	0.21	0.28
Theoretical RC	0.15	0.30	0.45	0.60

TABLE IV: Relative Complexities (RC)

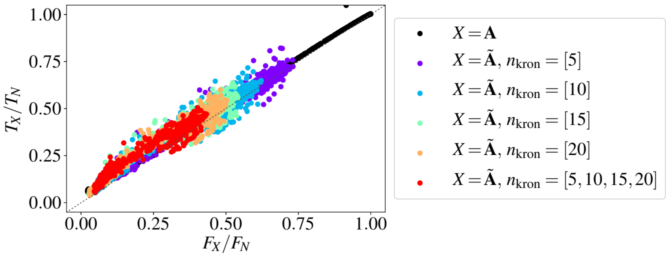


Fig. 5: Normalized number of flops vs. normalized running times for ISTA ($tol = 10^{-5}$, $\Gamma = 0.5$). Each point on the graph corresponds to an independent run at a given regularization $\lambda/\lambda_{\max} \in [10^{-2}, 1]$. Results for the usual GAP Safe test are shown in black and those using its stable version (as in Alg. 1) are shown in colors. The closer the dots lie to the identity line, the better the theoretical speedups are met in practice.

algorithm without screening, with the conventional screening and with the proposed approach, such that

$$F_X = \sum_{t=1}^{n_{\text{it}}} \text{flops}_X(t), \quad X \in \{N, \mathbf{A}, \tilde{\mathbf{A}}\} \tag{40}$$

where n_{it} is the number of iterations. When calculating $F_{\tilde{\mathbf{A}}}$, the current approximation at iteration t must be used in the expression of $\text{flops}_{\tilde{\mathbf{A}}}(t)$. Likewise, we denote T_N, $T_{\mathbf{A}}$ and $T_{\tilde{\mathbf{A}}}$ the measured running times.

As a main figure of merit, we evaluate the normalized number of flops ($F_{\mathbf{A}}/F_N$ and $F_{\tilde{\mathbf{A}}}/F_N$) and normalized running times ($T_{\mathbf{A}}/T_N$ and $T_{\tilde{\mathbf{A}}}/T_N$) [5].

In all simulated scenarios, the observed time reductions match closely the theoretical speedups predicted in terms of computational complexity, as illustrated in Figure 5 for the GAP Safe rule. Similar results are obtained for the Dynamic Safe rule, for other convergence tolerances and switching thresholds. Given this observation, in the remainder of the paper we report only running time results, which are more relevant in practice.

C. Choosing the switching threshold

The convergence-based switching criterion proposed in Section IV-A relies on a hyperparameter: the threshold Γ. As discussed then, this parameter determines how long the approximate dictionaries are kept.

We empirically observed that the choice of Γ is mostly dependent on the quality of the approximations $\tilde{\mathbf{A}}^i$ (represented here by the three simulation scenarios defined in Section V-A). In Figure 6 we show the normalized times as a function of $\Gamma \in [10^{-2}, 0.8]$ for each one of the three scenarios. Each line corresponds to a different regularization level λ. We can see that the low regularization configurations (dark blue curves) are more sensitive to choice of Γ, especially in the Hard scenario. However, a common behavior is observed regardless of the regularization and, for a given simulation scenario, a single Γ value can be chosen to obtain close to optimal execution times for any $\lambda/\lambda_{\max}$ in the tested range.

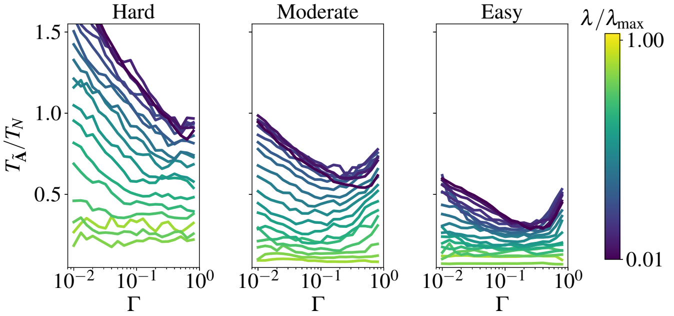


Fig. 6: Impact of the switching threshold Γ (x-axis) on the time results (y-axis) of Algorithm 1 using FISTA and GAP Safe. Each line corresponds to a regularization $\lambda/\lambda_{\max} \in [10^{-2}, 1]$.

Slightly different Γ values can be chosen to better adapt to each simulation scenario: $\Gamma = 0.1, 0.3, 0.5$ respectively on the easy, moderate and hard scenarios. In general terms, the *worse* the available approximations are (i.e. the harder the scenario) the higher the Γ to be chosen. This is intuitive, since it implies being more conservative in switching earlier when the available approximations are of lower quality.

If the aimed speedup-error compromise is not fixed, a good general compromise is picking $\Gamma = 0.5$, for instance.

D. Single vs. Multiple Approximations

In our experiments, using multiple dictionary approximations proved to be always advantageous when compared to using a single approximation, as soon as the switching parameter Γ is well-tuned (see Section V-C).

Figure 7 shows the normalized running times for an entire range of regularizations and corroborates the previous statement for both GAP and Dynamic Safe tests as well as for both ISTA and FISTA algorithms. The medians among 25 runs are plotted and the shaded area contains the 25%-to-75% percentiles. Although the single-approximation version of the proposed algorithm (as introduced in [24], [25]) already provides noticeable speedups to the tested gradient-based solvers, even with respect to their screening-based implementation (black curves), the use of multiple approximations (red curves) consistently leads to even better speedups. This ratifies the relevance of the generalization to multiple dictionaries introduced in this paper, by relying on more robust switching criteria. Similar results, which are omitted here, were obtained for other convergence tolerances and speedup-error compromise (these parameters are further explored in Sections V-E and V-F).

A smaller speedup is provided to the FISTA algorithm when compared to ISTA, which is expected since the former is already faster than the latter[6]. A similar argument applies to

[5] Time results are obtained in an Intel® Core™ i7-5600U CPU @ 2.60GHz, 16GB RAM. But since mostly time ratios are reported, the results here should be relatively consistent to other machine specifications.

[6] The results reported in the right graphs are already normalized w.r.t the FISTA time results, which are typically smaller than those of ISTA.

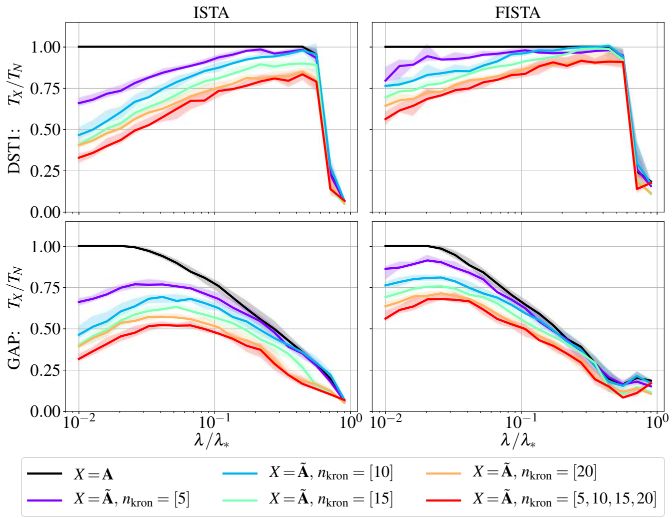


Fig. 7: Running times normalized w.r.t. T_N (ISTA or FISTA algorithms without screening). Conventional and Stable screening results respectively in black and colored lines. Left: ISTA. Right: FISTA. Top: Dynamic Safe screening. Bottom: GAP Safe. Moderate scenario, $tol = 10^{-5}$, $\Gamma = 0.5$.

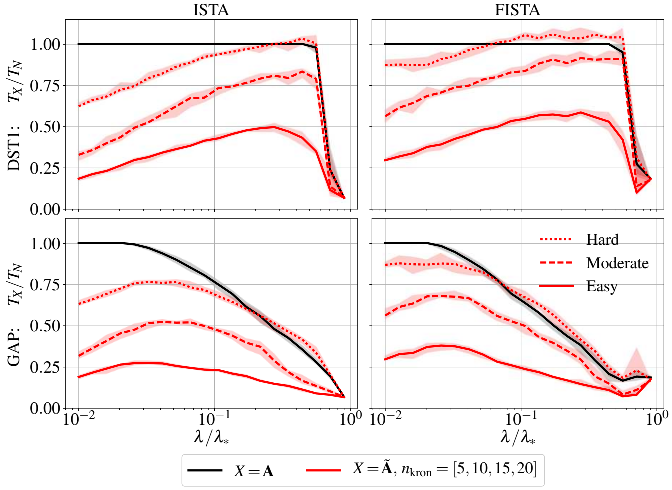


Fig. 8: Normalized running times in 3 different scenarios of speedup-error tradeoffs. $tol = 10^{-5}$, $\Gamma = 0.5$.

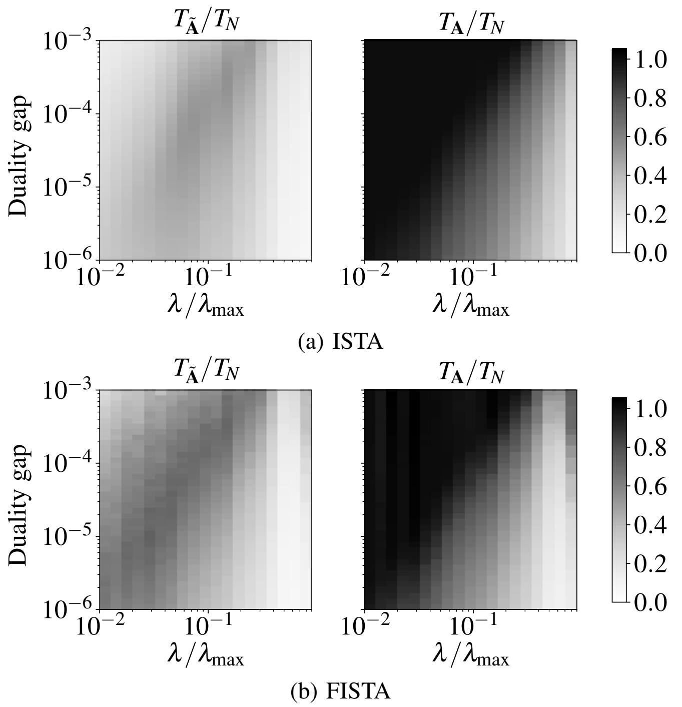
the GAP Safe rule, which has stronger screening capabilities than Dynamic Safe: our method still manages to provide some additional acceleration when combined to these already quite efficient techniques, especially in low-regularized scenarios.

For simplicity, only the results concerning multiple dictionary approximations will be reported from this point on.

E. Speedup-error compromise

It is reasonable to expect the quality of the approximations $\tilde{\mathbf{A}}^i$ to be decisive on the performance of the proposed algorithm. Its success depends on the possibility of providing fast yet precise approximations of the dictionary $\mathbf{A}$. Let us now evaluate the influence of this speedup-error compromise on the simulation time results.

Figure 8 shows the same type of normalized running times as in Figure 7, but it now compares the results of the proposed algorithm under three different scenarios in terms of speedup-error profiles (see Figure 4 for details on the approximation errors in each scenario). The impact on the algorithm performance is indeed significant. However, even on the Hard scenario, the proposed approach manages to provides non-negligible acceleration.



(a) ISTA

(b) FISTA

Fig. 9: Normalized execution times in grayscale (darker means slower) to reach a certain duality gap for a range of regularizations. Left: proposed technique (w/ stable screening). Right: Conventional screening.

F. Convergence tolerance

As previously highlighted, screening is known to work better on highly-regularized scenarios ($\lambda/\lambda_{\max} \approx 1$). The speedup provided by the screening is also more pronounced when a more precise convergence is required, because the final iterations are often less costly as most atoms have been screened out. These two aspects create the triangular profiles observed in Figure 9 (right plots). This figure shows the normalized execution times in grayscale (the darker, the slower) as a function of both the regularization (x-axis) and the convergence precision in terms of the duality gap (y-axis).

Note that the proposed method (left plots) efficiently complements the screening tests in its main weaknesses, namely: low regularizations and reduced convergence requirements. In weakly regularized scenarios, while screening tests struggle to eliminate atoms especially in initial iterations, the fast approximate dictionaries come at rescue by making those iterations faster. Besides that, when higher duality gaps are targeted, less iterations using the slow dictionary $\mathbf{A}$ are necessary. Thus, proportionally, the accelerated part of the algorithm (with fast dictionaries) is more significant.

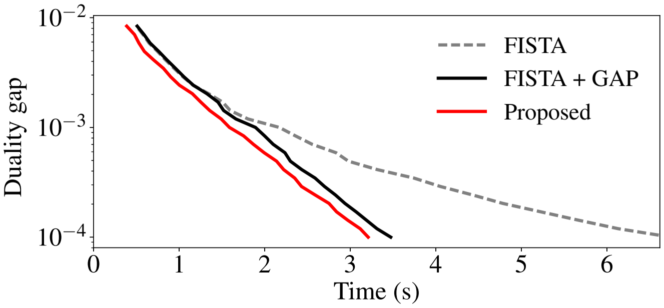


Fig. 10: Source localization using an MEG forward operator. Duality gap as a function of time for $\lambda/\lambda_{\max} = 10^{-1}$.

G. MEG source localization

In this experiment, we consider an MEG (magnetoencephalography) source localization problem, which consists in retrieving a limited set of active brain foci associated to a given MEG measured signal. A common way to achieve such a task is to solve a convex sparse inverse problem [41], [42].

We consider an MEG gain matrix (forward operator) $\mathbf{A} \in \mathbb{R}^{204\times 8193}$ obtained with the MNE software [43]. This operator can be efficiently approximated as a product of a few sparse matrices, as proposed in [15]. The speedups and corresponding approximation errors achieved by the so-called *FAμST* dictionaries adopted here are summarized in Table V.

The experiment consists in picking (uniformly) at random eight active sources with standard gaussian weights giving 8-sparse coefficient vectors $\mathbf{x} \in \mathbb{R}^{8193}$ to be recovered from the input signals $\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{R}^{204}$ by solving problem (1).

Mean ϵ_j	$2.1\cdot 10^{-1}$	$1.2\cdot 10^{-1}$	$6.6\cdot 10^{-2}$	$5.9\cdot 10^{-2}$
Measured RC	0.22	0.33	0.45	0.63

TABLE V: Approximation errors and Relative Complexities for the FAuST approximations of the MEG matrix.

Figure 10 shows the required time to reach a certain duality gap precision (averaged over 50 independent runs) for a fixed regularization $\lambda/\lambda_{\max} = 10^{-1}$, around which the correct number of sources is recovered. The proposed method is consistently advantageous, providing some speedup with respect to both the FISTA algorithm (grey dotted line) and its enhanced version with GAP Safe screening (black line). Although somewhat modest when compared to the latter, the provided gain is proportionally more pronounced at lower precision requirements in the duality gap. [7]

VI. Conclusion

Aiming to combine safe screening tests and fast structured operators in an effort to accelerate the solution of ℓ_1-minimization problems, we proposed a methodology for defining safe screening tests despite an inaccurate knowledge of the explanatory variables (dictionary atoms). The proposed methodology was exemplified on some existing screening tests (stable and dynamic). The resulting *stable screening tests* were then employed in a fast algorithm, which exploits a series of fast approximations of the dictionary matrix. Simulation results demonstrated the two combined strategies to be quite complementary, justifying the effort to conciliate them. The first part of the paper also provides better understanding on which error measures are more significant to the stable screening tests, potentially leading to more suited optimization criteria to dictionary approximation techniques.

In any case, the stable tests introduced here have a broader interest, as the source of imprecision need not be linked to the use of fast approximations. It can simply be seen as a robust screening tool for inaccurately-defined ℓ_1-minimization problems –for instance, due to intrinsic imprecisions on the measurement matrix of a given inverse problem.

In principle, the proposed framework can also be applied to other types of safe regions, error bounds as well as other screening tests than the ones explored in this paper. Another interesting perspective is to extend these zone-based screening tests to off-the-grid generalizations of ℓ_1-regularized problems.

Appendix

A. Proof of Lemma 2

By definition we have $\boldsymbol{\theta}_F := \Theta'(\mathbf{z}|\tilde{\mathbf{A}}, \boldsymbol{\epsilon}) = \alpha\mathbf{z}$ where

$$0 \leq \alpha \leq \frac{1}{\max_j \left(|\tilde{\mathbf{a}}_j^T \tilde{\mathbf{z}}| + \epsilon_j \|\mathbf{z}\|_{q^*}\right)}$$

As a result for any $1 \leq j \leq K$ we have

$$|\mathbf{a}_j^T \boldsymbol{\theta}_F| = |(\tilde{\mathbf{a}}_j + \mathbf{e}_j)^T \alpha\mathbf{z}| \leq |\alpha| \left(|\tilde{\mathbf{a}}_j^T \mathbf{z}| + \epsilon_j \|\mathbf{z}\|_{q^*}\right) \leq 1.$$

This implies that the dual point $\boldsymbol{\theta}_F$ is feasible, i.e. $\boldsymbol{\theta}_F \in \Delta_{\mathbf{A}}$. The fact that $\boldsymbol{\theta}_F \in \Delta_{\tilde{\mathbf{A}}}$ follows similarly (since $\epsilon_j \|\mathbf{z}\|_{q^*} > 0$).

B. Detailed complexities

We suppose that the p^*-norms of the atoms $\|\mathbf{a}_j\|_{p^*}$ and q-norms of the atom approximation error $\|\mathbf{e}_j\|_q$ are precalculated and stored and memory. The total screening complexity is summarized in Table IV and detailed in the following.

Screening	$\boldsymbol{\theta}_t$	$\mathcal{R}$	Test	**Total**
Dynamic Safe	$3N + \lvert\mathcal{A}\rvert$	N	$\lvert\mathcal{A}\rvert$	$4N + 2\lvert\mathcal{A}\rvert$
Stable Dynamic Safe	$4N + 2\lvert\mathcal{A}\rvert$	N	$\lvert\mathcal{A}\rvert$	$5N + 3\lvert\mathcal{A}\rvert$
GAP Safe	$3N + \lvert\mathcal{A}\rvert$	N	$\lvert\mathcal{A}\rvert$	$4N + 2\lvert\mathcal{A}\rvert$
Stable GAP Safe	$4N + 2\lvert\mathcal{A}\rvert$	N	$N + 2\lvert\mathcal{A}\rvert$	$6N + 4\lvert\mathcal{A}\rvert$

TABLE VI: Screening: number of floating-point operations.

1) Feasible point $\boldsymbol{\theta}_t$: A total of at most $3N + |\mathcal{A}_t|$ operations for the point in eq. (11), distributed as follows:

N dot product $\mathbf{y}^T \boldsymbol{\rho}_t$.
N norm $\|\boldsymbol{\rho}_t\|_2^2$ (if primal objective not calculated as a convergence criterion).
N product $\alpha\boldsymbol{\rho}_t$.
$|\mathcal{A}_t|$ comparisons for the infinity norm $\|\mathbf{A}^T\boldsymbol{\rho}\|_\infty$.

At most $N + |\mathcal{A}_t|$ extra operations for the point in eq. (22):

N norm $\|\tilde{\boldsymbol{\rho}}_t\|_{q^*}$ (only if $q^* \neq 2$).
$|\mathcal{A}_t|$ products $\epsilon_j \|\tilde{\boldsymbol{\rho}}_t\|_{q^*}$ on the calculation of α'_t.

[7] Neither of the two considered acceleration techniques affects the algorithms results in terms of source localization accuracy, since they solve the exact same problem at the same convergence precision. That is why task-specific performance measures are not discussed (see [42] for more details on the performance of ℓ_1-minimization techniques for brain source localization).

2) Safe region $\mathcal{R}$: At most N operations for computing $\|\boldsymbol{\theta}_t - \mathbf{y}/\lambda\|_2$ (if the dual objective $D(\boldsymbol{\theta}_t)$ is not already calculated as a convergence criterion) for the radius of either Dynamic Safe or GAP Safe sphere.

3) Screening test: Other operations (aside from the mentioned $\mathbf{A}^T\mathbf{c}$) sum up to $|\mathcal{A}_t|$ operations for the product $R\|\mathbf{a}_j\|_{p^*}\ \forall\ j \in \mathcal{A}_t$. The proposed stable screening tests require at most an extra $N + |\mathcal{A}_t|$ operations:

N	norm $\|\mathbf{c}\|_{q^*}$ (if $q^* \neq 2$ and $\mathbf{c}$ varies with t).
$\|\mathcal{A}_t\|$	products $\epsilon_j\|\mathbf{c}\|_{q^*}\ \forall\ j \in \mathcal{A}_t$ (if $\mathbf{c}$ varies with t).

References

[1] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "Audio Inpainting," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 922 – 932, Mar 2012.

[2] D. Malioutov, M. Cetin, and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 3010–3022, 2005.

[3] C. Dossal and S. Mallat, "Sparse spike deconvolution with minimum scale," in *Signal Processing with Adaptive Sparse Structured Representations (SPARS workshop)*. Citeseer, 2005, pp. 1–4.

[4] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[5] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. The Twenty-Seventh Asilomar Conference on*, Nov 1993, pp. 40–44 vol.1.

[6] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1998.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[8] 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*, vol. 57, no. 11, pp. 1413–1457, 2004.

[9] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[10] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Transactions on Image Processing*, vol. 16, no. 12, pp. 2992–3004, 2007.

[11] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, July 2009.

[12] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems withapplications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, May 2011.

[13] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1553–1564, 2010.

[14] O. Chabiron, F. Malgouyres, J.-Y. Tourneret, and N. Dobigeon, "Toward fast transform learning," *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 195–216, 2015.

[15] L. L. Magoarou and R. Gribonval, "Flexible multilayer sparse approximations of matrices and applications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 688–700, June 2016.

[16] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Transactions on Signal Processing*, vol. 64, no. 12, pp. 3180–3193, June 2016.

[17] C. Dantas, M. N. da Costa, and R. Lopes, "Learning dictionaries as a sum of kronecker products," *IEEE Signal Processing Letters*, 2017.

[18] L. El Ghaoui, V. Viallon, and T. Rabbani, "Safe feature elimination in sparse supervised learning," *EECS Department, University of California, Berkeley, Tech. Rep*, 2010.

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

[20] A. Bonnefoy, V. Emiya, L. Ralaivola, and R. Gribonval, "Dynamic screening: Accelerating first-order algorithms for the lasso and group-lasso," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5121–5132, 2015.

[21] O. Fercoq, A. Gramfort, and J. Salmon, "Mind the duality gap: safer rules for the lasso," in *Proc. ICML 2015*, July 2015.

[22] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon, "Gap safe screening rules for sparsity enforcing penalties," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 4671–4703, 2017.

[23] Z. J. Xiang, Y. Wang, and P. J. Ramadge, "Screening tests for lasso problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2016.

[24] C. F. Dantas and R. Gribonval, "Dynamic Screening with Approximate Dictionaries," in *XXVI colloque GRETSI*, Juan-les-Pins, France, 2017.

[25] ——, "Faster and still safe: combining screening techniques and structured dictionaries to accelerate the Lasso," in *ICASSP 2018 - IEEE International Conference on Acoustics, Speech and Signal Processing*, Calgary, AB, Canada, #Apr# 2018, pp. 1–5.

[26] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, 2013.

[27] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 438–445.

[28] C. F. Dantas, J. E. Cohen, and R. Gribonval, "Learning fast dictionaries for sparse representations using low-rank tensor decompositions," in *LVA/ICA 2018 - 14th International Conference on Latent Variable Analysis and Signal Separation*, Guildford, United Kingdom, Jul 2018.

[29] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, nov 2008.

[30] 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)*, vol. 74, no. 2, pp. 245–266, nov 2011.

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

[32] J. Wang, P. Wonka, and J. Ye, "Lasso screening rules via dual polytope projection," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1063–1101, 2015.

[33] A. Malti and C. Herzet, "Safe screening tests for LASSO based on firmly non-expansiveness," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2016.

[34] J. Liu, Z. Zhao, J. Wang, and J. Ye, "Safe screening with variational inequalities and its application to lasso," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, pp. II–289–II–297.

[35] J. Kim and H. Park, "Fast active-set-type algorithms for l1-regularized linear regression," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 397–404.

[36] T. Johnson and C. Guestrin, "Blitz: A principled meta-algorithm for scaling sparse optimization," in *International Conference on Machine Learning*, 2015, pp. 1171–1179.

[37] A. G. M. Massias and J. Salmon, "From safe screening rules to working sets for faster Lasso-type solvers," in *NIPS-OPT*, 2017.

[38] M. Massias, J. Salmon, and A. Gramfort, "Celer: a fast solver for the lasso with dual extrapolation," in *International Conference on Machine Learning*, 2018, pp. 3321–3330.

[39] C. Herzet and A. Drémeau, "Joint Screening Tests for LASSO," in *ICASSP 2018 - IEEE International Conference on Acoustic, Speech and Signal Processing*, Calgary, Canada, #Apr# 2018, pp. 1–5.

[40] M. R. Osborne, B. Presnell, and B. A. Turlach, "On the lasso and its dual," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 319–337, 2000.

[41] K. Matsuura and Y. Okabe, "Selective minimum-norm solution of the biomagnetic inverse problem," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 6, pp. 608–615, June 1995.

[42] A. Gramfort, M. Kowalski, and M. Hämäläinen, "Mixed-norm estimates for the m/eeg inverse problem using accelerated gradient methods," *Physics in Medicine & Biology*, vol. 57, no. 7, p. 1937, 2012.

[43] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "Mne software for processing meg and eeg data," *Neuroimage*, vol. 86, pp. 446–460, 2014.