Safe Feature Elimination in Sparse Supervised Learning

Laurent El Ghaoui
Vivian Viallon
Tarek Rabbani

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2010-126
http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-126.html

September 21, 2010

Safe Feature Elimination in Sparse Supervised Learning

Laurent El Ghaoui ELGHAOUI@EECS.BERKELEY.EDU
Vivian Viallon VIALLON@EECS.BERKELEY.EDU
Tarek Rabbani RABBANI@EECS.BERKELEY.EDU
Department of EECS
University of California
Berkeley, CA 94720

Date: September 17, 2010

Abstract

We investigate fast methods that allow to quickly eliminate variables (features) in supervised learning problems involving a convex loss function and a l_1-norm penalty, leading to a potentially substantial reduction in the number of variables prior to running the supervised learning algorithm. The methods are not heuristic: they only eliminate features that are *guaranteed* to be absent after solving the learning problem. Our framework applies to a large class of problems, including support vector machine classification, logistic regression and least-squares.

The complexity of the feature elimination step is negligible compared to the typical computational effort involved in the sparse supervised learning problem: it grows linearly with the number of features times the number of examples, with much better count if data is sparse. We apply our method to data sets arising in text classification and observe a dramatic reduction of the dimensionality, hence in computational effort required to solve the learning problem, especially when very sparse classifiers are sought. Our method allows to immediately extend the scope of existing algorithms, allowing us to run them on data sets of sizes that were out of their reach before.

Keywords: Sparse classification, sparse regression, LASSO, feature elimination.

Contents

1. Introduction

In recent years, "sparse" classification or regression problems, which involve an l_1-norm penalty on the problem variables, have been increasingly popular, due to their ability to strike a useful trade-off between a loss on the training data and predictive accuracy, and at the same time encouraging sparsity of the classification or regression optimal coefficients.

Several efficient algorithms have been developed for l_1-penalized regression or classification problems: in Kim et al. (2007); Efron et al. (2004); Candès and Romberg (2006); Donoho and Tsaig (2008); Park and Hastie (2007); Friedman et al. (2007) for the LASSO problem (which corresponds to the squared loss function), in Park and Hastie (2007); Friedman et al. (2007); Koh et al. (2007); Genkin et al. (2007); Lee et al. (2006); Goodman (2004); Roth (2004) for the case of logistic regression and related generalized linear models, in Fan et al. (2008); Bi et al. (2003); Fung and Mangasarian (2004); Zhu et al. (2003) for the case of support vector machines (hinge loss). However, the complexity of these algorithms, when it is known, grows fast with the number of variables. For example, the interior-point method in Kim et al. (2007) has a worst-case complexity of $O(n^2m + m^3)$, where n is the number of variables (features) and m that of data points. Hence it is of interest to be able to efficiently eliminate features in a pre-processing step.

Feature selection methods are often used to accomplish dimensionality reduction, and are of utmost relevance for data sets of massive dimension, see for example Fan and Lv (2010). These methods, when used as a pre-processing step, have been referred to in the literature as *screening* procedures Fan and Lv (2010, 2008). They typically rely on univariate models to score features, independently of each other, and are usually computationally fast. Classical procedures are based on correlation coefficients, two-sample t-statistics or chi-square statistics Fan and Lv (2010); see also Forman (2003) and the references therein for an overview in the specific case of text classification. Most screening methods might remove features that could otherwise have been selected by the regression or classification algorithm. However, some of them were recently shown to enjoy the so-called "sure screening" property Fan and Lv (2008): under some technical conditions, no relevant feature is removed, with probability tending to one.

Screening procedures typically ignore the specific classification task to be solved after feature elimination. In this paper, we propose to remove features based on the supervised learning problem considered. Our approach works on a large class of convex classification or regression problems, and eliminates features based on both the structure of the loss function and the problem data. The features are eliminated according to a sufficient, in general conservative, condition. Hence, we never remove features unless they are *guaranteed* to be absent if one were to solve the full-fledged classification or regression problem. The complexity of our SAFE (SAfe Feature Elimination) method grows as $O(mn)$, where m is the number of training points, and n the number of (features), with improved counts when the data matrix is sparse. Our test becomes more aggressive at removing features as the penalty parameter grows.

The specific application we have in mind involves large data sets of text documents, and sparse matrices based on occurrence, or other score, of words or terms in these documents. We seek extremely sparse optimal coefficient vectors, even if that means operating at values of the penalty parameter that are substantially larger than those dictated by a pure concern for predictive accuracy. The fact that we need to operate at high values of this parameter opens the hope that, at least for the application considered, the number of features eliminated by using our fast test is high enough to allow a dramatic reduction in computing time and memory requirements. Our experimental results indicate that for many of these data sets, we do observe a dramatic reduction in the number of variables, typically by an order of magnitude or more, expanding the reach of existing algorithms for large-scale data.

The paper is organized as follows. In section 2, we introduce a generic formalism for classification and regression, covering the squared, logistic, and hinge loss functions. Section 3 describes the feature elimination method for generic loss functions. Section 4 provides details of the generic method in

	loss function $f(\xi)$	conjugate function $f^*(\vartheta)$	domain of f^*
squared	$f_{\rm sq}(\xi) = (1/2)\xi^2$	$(1/2)\vartheta^2$	$\mathbb{R}$
logistic	$f_{\log}(\xi) = \log(1+e^{-\xi})$	$(-\vartheta)\log(-\vartheta) + (\vartheta+1)\log(\vartheta+1)$	$[-1,0]^m$
hinge	$f_{\rm hi}(\xi) = (1-\xi)_+$	$-\vartheta$	$[-1,0]^m$

Table 1: Expression for the conjugate of popular loss functions, adopting the convention $0\log 0 = 0$ for the logistic loss.

the specific case of the squared loss function; section 5 and 6 focus on the hinge and loss functions, respectively. Section 7 illustrates the approach in the context of text classification, and empirically demonstrates that, when the classifier sought should be extremely sparse, the approach provides a substantial reduction in problem size.

Notation. We use $\mathbf{1}$ to denotes a vector of ones, with size inferred from context. For a scalar a, a_+ denotes the positive part of a. For a vector a, this operation is component-wise, so that $\mathbf{1}^T a_+$ is the sum of the positive elements in a. We take the convention that a sum over an empty index sets, such as $\sum_{i=1}^k a_i$ with $k \leq 0$, is zero.

2. Problem Setting

We consider a generic supervised learning problem of the form

$$\mathcal{P}(\lambda) \;:\; \phi(\lambda) := \min_{w,v} \sum_{i=1}^m f(a_i^T w + b_i v + c_i) + \lambda\|w\|_1, \tag{1}$$

where the loss function f is convex, $a_i \in \mathbb{R}^n$, $i = 1,\ldots,m$, $b, c \in \mathbb{R}^m$ are given, and $\lambda > 0$ is a penalty parameter encouraging sparsity of the vector w. We define the *feature matrix* $A := [a_1, \ldots, a_m] \in \mathbb{R}^{n\times m}$, and denote its k-th row by $x_k \in \mathbb{R}^m$, $k = 1,\ldots,n$, so that $A^T = [x_1^T, \ldots, x_n^T]$.

Our formalism covers the well-known LASSO problem (see, *e.g* Efron et al. (2004))

$$\phi_{\rm sq}(\lambda) := \min_w \; \frac{1}{2}\sum_{i=1}^m (a_i^T w - y_i)^2 + \lambda\|w\|_1, \tag{2}$$

for which the loss function is the squared loss: $f = f_{\rm sq}$, with $f_{\rm sq}(\xi) = (1/2)\xi^2$, with $a_i \in \mathbb{R}^n$, $i = 1,\ldots,n$ the data points, $c = -y$ is the (negative) response vector, and $b = 0$. Logistic regression and support vector machine classification models as also covered in by our formalism, as detailed in sections 6 and 5, respectively.

We will denote by f^* the conjugate of the loss function f, which is the extended-value convex function defined as

$$f^*(\vartheta) := \max_\xi \; \xi\vartheta - f(\xi).$$

Beyond convexity, we make a few mild assumptions about the loss function f. First, we assume that it is non-negative everywhere, and that it is closed (its epigraph is closed), so that $f^{**} = f$. These assumptions are met with the squared, logistic and hinge loss functions, as well as other popular loss functions. The conjugate of the three loss functions: squared, logistic and hinge, which we henceforth refer to with the subscripts lo, hi and sq, are given in Table 1.

3. Generic Case

In this section, we describe the basic idea as it applies to the generic sparse supervised learning problem (1).

3.1 Dual problem

The first step is to devise the dual of problem (1), which is

$$\mathcal{P}(\lambda) \;:\; \phi(\lambda) = \max_{\theta} G(\theta) \;:\; \theta^T b = 0, \;\; |\theta^T x_k| \leq \lambda, \;\; k = 1, \ldots, n, \tag{3}$$

where

$$G(\theta) := c^T \theta - \sum_{i=1}^{m} f^*(\theta_i) \tag{4}$$

is the dual function, which is, by construction, concave. We assume that strong duality holds and primal and dual optimal points are attained. Due to the optimality conditions for the problem (see Boyd and Vandenberghe (2004)), constraints for which $|\theta^T x_k| < \lambda$ at optimum correspond to a zero element in the primal variable: $w_k = 0$.

3.2 Basic idea

Assume that a lower bound γ on the optimal value of the learning problem $\phi(\lambda)$ is known: $\gamma \leq \phi(\lambda)$. (Without loss of generality, we can assume that $0 \leq \gamma \leq \sum_{i=1}^{m} f(c_i)$.) Since γ is a lower bound on the dual function, we can safely add the corresponding lower bound constraint in the dual problem:

$$\phi(\lambda) := \max_{\theta} G(\theta) \;:\; G(\theta) \geq \gamma, \;\; \theta^T b = 0, \;\; |\theta^T x_k| \leq \lambda, \;\; k = 1, \ldots, n.$$

This implies that the test

$$\lambda > T(\gamma, x_k) := \max_{\theta} |\theta^T x_k| \;:\; G(\theta) \geq \gamma, \;\; \theta^T b = 0 \tag{5}$$

allows to eliminate the k-th feature. Figure 1 illustrates the basic idea.

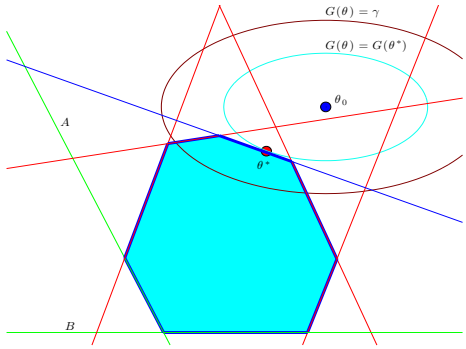

Figure 1: The basic idea of safe feature elimination, with a given lower bound γ. The feasible set of the dual problem is the shaded polytope. Two level sets of the dual function are shown, one corresponds to the optimal value and the other to the given lower bound γ. Constraints (A) and (B) (in green) are safely eliminated, but some inactive constraints (in red) are not. Here, θ_0 corresponds to the unconstrained maximum of the dual function.

For a given $x \in \mathbb{R}^m$, and $\gamma \in \mathbb{R}$, we have

$$T(\gamma, x) = \max(P(\gamma, x), P(\gamma, -x)),$$

where $P(\gamma, x)$ is the optimal value of a convex optimization problem with two constraints only:

$$P(\gamma, x) := \max_{\theta} \theta^T x \ : \ G(\theta) \geq \gamma, \ \ \theta^T b = 0. \tag{6}$$

Since $P(\gamma, x)$ decreases when γ increases, the closer $\phi(\lambda)$ is to its lower bound γ, the more aggressive (accurate) our test is.

By construction, the dual function G is decomposable as a sum of functions of one variable only. This particular structure allows to solve problem (6) very efficiently, using for example interior-point methods, for a large class of loss functions f. Alternatively, we can express the problem in dual form as a convex optimization problem with two scalar variables:

$$P(\gamma, x) = \min_{\mu > 0, \, \nu} -\gamma\mu + \mu \sum_{i=1}^{m} f\left(\frac{x_i + \mu c_i + \nu b_i}{\mu}\right). \tag{7}$$

Note that the expression above involves the perspective of the function f, which is convex (see Boyd and Vandenberghe (2004)). For many loss functions f, the above problem can be efficiently solved using a variety of methods for convex optimization, in (close to) $O(m)$ time. We can also set the variable $\nu = 0$, leading to a simple bisection problem over μ. This amounts to ignore the constraint $\theta^T b = 0$ in the definition of $P(\gamma, x)$, resulting in a more conservative test. More generally, any pair (μ, ν) with $\mu > 0$ generates an upper bound on $P(\gamma, x)$, which in turn corresponds to a valid, perhaps conservative, test.

3.3 Lower bound γ obtained by dual scaling

One way to get a lower bound γ is to find a dual point θ that is feasible for the dual problem $\mathcal{P}(\lambda)$, and then set $\gamma = G(\theta)$.

To get a dual feasible point, we can solve the problem for a higher value of $\lambda_0 \geq \lambda$ of the penalty parameter. (In the specific case examined below, we will see how to set λ_0 so that the vector $w = 0$ at optimum.) This provides a dual point θ_0 that is feasible for $\mathcal{P}(\lambda_0)$, which satisfies $\lambda_0 = \|X\theta_0\|_\infty$. In turn, θ_0 can be scaled so as to become feasible for $\mathcal{P}(\lambda)$. Precisely, we set $\theta = s\theta_0$, with $\|X\theta\|_\infty \leq \lambda$ equivalent to $|s| \leq \lambda/\lambda_0$. In order to find the best possible scaling factor s, we solve the one-dimensional, convex problem

$$\gamma(\lambda) := \max_{s} \ G(s\theta_0) \ : |s| \leq \frac{\lambda}{\lambda_0}. \tag{8}$$

Under mild conditions on the loss function f, the above problem can be solved by bisection in $O(m)$ time. By construction, $\gamma(\lambda)$ is a lower bound on $\phi(\lambda)$. We proceed by computing the quantities $P(\gamma(\lambda), x)$ (via expression (7)), $T(\gamma(\lambda), x)$ for $x = \pm x_k$, $k = 1, \ldots, n$, and apply the test (5). Assuming θ_0 is already available, the complexity of our test, when used on all the n features, grows as $O(nm)$, with a better count if the data is sparse.

3.4 A specific dual point θ_0

We can generate an initial point θ_0 by solving the problem with $w = 0$. We get

$$\min_{v} \sum_{i=1}^{m} f(b_i v + c_i) = \min_{v} \max_{\theta} \theta^T (bv + c) - \sum_{i=1}^{m} f^*(\theta_i) = \max_{\theta \, : \, b^T\theta = 0} G(\theta).$$

Solving the one-dimensional problem above can be often done in closed-form, or by bisection, in $O(m)$. Choosing θ_0 to be any optimal for the corresponding dual problem (the one on the right-hand side) generates a point that is dual feasible for it, that is, $G(\theta_0)$ is finite, and $b^T\theta_0 = 0$. The above specific construction is illustrated in Figure 2.

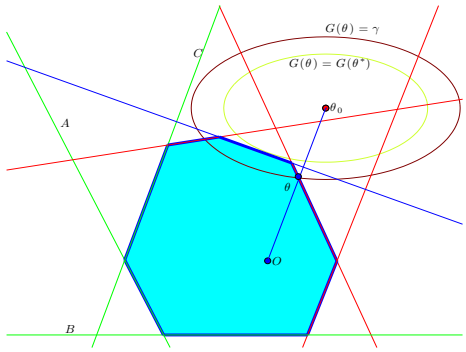

Figure 2: Safe feature elimination based on scaling the dual point θ_0 that is optimal for the problem where all features (constraints) can be removed. This choice leads to a better lower bound γ over that in Fig. 1, as now an additional constraint (C) is removed.

The point θ_0 satisfies all the constraints of problem $\mathcal{P}(\lambda)$, except perhaps for the constraint $\|X\theta_0\|_\infty \leq \lambda$. Hence, if $\lambda \geq \lambda_{\max} := \|X\theta_0\|_\infty$, then $w = 0$ (and θ_0) is optimal for $\mathcal{P}(\lambda)$. Note that, since θ_0 may not be uniquely defined, $\lambda_{\max}$ may not necessarily be the smallest value for which $w = 0$ is optimal for the primal problem.

3.5 Recursive SAFE

In many applications, we are interested in solving problem (3) for a given sequence of λ values. Alternatively, our objective is to find, using a binary search on λ, a value of λ that yields a certain cardinality (number of non-zero coefficients) of the optimal classifier. In both cases, we can use SAFE in order to reduce the overall computational load.

For example, if we want to solve the problem for a given sequence of decreasing λ values, say $\lambda_1 \geq \ldots \geq \lambda_N$, then at each step we can use the previously computed solution to find a bound γ. Specifically, if we have solved problem $\mathcal{P}(\lambda_t)$ for some $t \geq 1$, and are able to find a corresponding optimal dual point θ_t, then we can use the scaling method of section 3.4 to find a bound γ_{t+1} on the problem $\mathcal{P}(\lambda_{t+1})$.

In the context of binary search, we maintain upper and lower bounds on a "target" value of λ. For each upper bound, we proceed by scaling as before. For lower values of λ, no scaling is necessary, since the feasible set of the dual problem $\mathcal{P}(\lambda_0)$ is contained in that of problem $\mathcal{P}(\lambda)$ if $\lambda \geq \lambda_0$. However, we can still improve the corresponding bound γ by scaling, that is, solving problem (8) with $\lambda \geq \lambda_0$.

In both cases, we can potentially improve the bound by scaling all the dual points we have obtained so far, and choosing our bound γ to be the smallest of each corresponding bound.

4. LASSO

In this section we apply the approach outlined before to the LASSO problem (2). The dual problem is

$$\phi_{\text{sq}}(\lambda) = \max_\theta \; G_{\text{sq}}(\theta) \;\; : \;\; |\theta^T x_k| \leq \lambda, \;\; k = 1, \ldots, n,$$

where x_k is the k-th row of the $n \times m$ matrix $X := [a_1, \ldots, a_m]$, and

$$G_{\text{sq}}(\theta) := -y^T\theta - \frac{1}{2}\theta^T\theta = \frac{1}{2}y^Ty - \frac{1}{2}\|\theta + y\|_2^2.$$

The following relationship holds between optimal primal and dual variables w, θ:

$$\theta = X^T w - y. \tag{9}$$

4.1 Test, γ given

Let us first examine the case when a lower bound γ on the problem is known: $\gamma \leq \phi(\lambda)$. Without loss of generality, we may assume $\gamma \in [0, \gamma_{\max}]$, where $\gamma_{\max} := (1/2)y^T y$. Since $z = 0$, the quantity defined in (7) is given by

$$\begin{aligned} P_{\rm sq}(\gamma, x) &= \min_{\mu>0} -\gamma\mu + \mu \sum_{i=1}^{m} f_{\rm sq}\left(\frac{x_i - \mu y_i}{\mu}\right) \\ &= \min_{\mu>0} -\gamma\mu + \frac{1}{2\mu}\|x - \mu y\|_2^2 \\ &= \min_{\mu>0} \frac{\mu}{2}(y^T y - 2\gamma) + \frac{1}{2\mu}x^T x - y^T x \\ &= \|x\|_2 \cdot \sqrt{y^T y - 2\gamma} - y^T x, \end{aligned}$$

where we have exploited $\gamma \leq \gamma_{\max} = (1/2)y^T y$. We obtain

$$T_{\rm sq}(\gamma, x) = \max(P_{\rm sq}(\gamma, x), P_{\rm sq}(\gamma, -x)) = |y^T x| + D(\gamma)\|x\|_2, \text{ where } D(\gamma) := \sqrt{y^T y - 2\gamma}.$$

Our test for removing the k-th feature passes if

$$\lambda > |y^T x_k| + \sqrt{y^T y - 2\gamma} \cdot \|x_k\|_2. \tag{10}$$

4.2 Basic SAFE-LASSO theorem

Let us now examine a basic choice for the lower bound γ, based on the "dual scaling" approach described in section 3.4, and on choosing λ_0 such that $w = 0$ is optimal for $\mathcal{P}(\lambda_0)$.

We first find the smallest value $\lambda_{\max}$ of λ above which we can guarantee that $w = 0$ is optimal for $\mathcal{P}(\lambda)$. Due to the optimality condition (9), $w = 0$ is optimal implies that $\theta_{\max} := -y$ is optimal for the dual problem, hence it is feasible, which in turn implies

$$\lambda \geq \lambda_{\max} := \max_{1 \leq j \leq n} |x_j^T y| = \|Xy\|_\infty.$$

Conversely, if the above condition holds, then the point $\theta = \theta_{\max} = -y$ is dual feasible, and achieves the value attained for $w = 0$ (namely, $y^T y/2$), which proves that the latter is primal optimal.

We follow the scaling technique of section 3.4 with $\lambda_0 = \lambda_{\max}$, and assuming $\lambda \leq \lambda_{\max}$ from now on. We proceed by scaling the dual point $\theta_{\max} = -y$, which is feasible for $\mathcal{P}_{\rm sq}(\lambda_{\max})$, so that the scaled version $\theta = s\theta_{\max}$ is feasible for $\mathcal{P}_{\rm sq}(\lambda)$. The corresponding lower bound γ is found by solving

$$\gamma(\lambda) = \max_s G_{\rm sq}(s\theta_{\max}) \ : \ |s| \leq \frac{\lambda}{\lambda_{\max}}.$$

With $\theta_{\max} = -y$, and

$$G_{\rm sq}(-sy) = \frac{1}{2}y^T y - \frac{1}{2}\|y - sy\|_2^2 = \frac{1}{2}y^T y(1 - (1-s)^2),$$

the above problem can solved in closed form:

$$\gamma(\lambda) = \frac{1}{2}y^T y \left(1 - \left(1 - \frac{\lambda}{\lambda_{\max}}\right)^2\right), \ \ D(\lambda) := \sqrt{y^T y - 2\gamma(\lambda)} = \|y\|_2 \frac{\lambda_{\max} - \lambda}{\lambda_{\max}},$$

with the optimal scaling equal to $s^* = \lambda/\lambda_{\max}$. Our criterion to remove the k-th feature from problem $\mathcal{P}(\lambda)$ becomes

$$\lambda > |y^T x_k| + D(\lambda)\|x_k\|_2 = |y^T x_k| + \|y\|_2\|x_k\|_2 \cdot \frac{\lambda_{\max} - \lambda}{\lambda_{\max}}.$$

The test can be summarized as follows.

Theorem 1 (Basic SAFE-LASSO) *For the LASSO problem (2), and denoting by x_k the k-th row of the matrix X, the condition*

$$\lambda > \rho_k \lambda_{\max}, \text{ with } \rho_k := \frac{\|y\|_2\|x_k\|_2 + |y^T x_k|}{\|y\|_2\|x_k\|_2 + \lambda_{\max}}, \quad \lambda_{\max} := \max_{1\le j\le n} |y^T x_j|, \tag{11}$$

allows to safely remove the k-th feature.

The complexity of running this test through all the features is $O(mn)$, with a better count if the data is sparse. The main computational burden in the test is actually independent of λ, and can be done once and for all: it suffices to rank features according to the values of ρ_k, $k = 1, \ldots, n$. Note that this test accurately predicts the value of $\lambda = \lambda_{\max}$ for which all the features can be safely removed, that is, $w = 0$ at the optimum for $\mathcal{P}_{\rm sq}(\lambda)$.

In the case of scaled data sets, for which $\|x_k\|_2 = 1$ for every k, the expression above has a convenient geometrical interpretation:

$$\rho_k = \frac{1 + |\cos\alpha_k|}{1 + \max\limits_{1\le j\le n} |\cos\alpha_j|}, \tag{12}$$

where α_k is the angle between the k-feature and the response vector y. Our test then consists in eliminating features based on how closely they are aligned with the response, *relative* to the most closely aligned feature. For scaled data sets, our test is very similar to standard correlation-based feature selection Fan and Lv (2008); in fact, for scaled data sets, the ranking of features it produces is then exactly the same. The big difference here is that our test is not a heuristic, as it only eliminates features that are *guaranteed* to be absent when solving the full-fledged sparse supervised learning problem.

4.3 Starting from another dual point

As explained in the generic case in section 3.4, we can start from an arbitrary dual point θ_0 that is feasible for $\mathcal{P}_{\rm sq}(\lambda_0)$, where $\lambda_0 \ge \lambda$ is given (the previous section corresponds to $\lambda_0 = \lambda_{\max}$). One way to generate such a point is to start with w_0 that is optimal for $\mathcal{P}_{\rm sq}(\lambda_0)$ in primal form. Then the point

$$\theta_0 := X^T w_0 - y$$

is optimal for the dual problem $\mathcal{P}_{\rm sq}(\lambda_0)$, and satisfies $\lambda_0 = \|X\theta_0\|_\infty$. To avoid trivialities, we assume $\theta_0 \neq 0$, $\lambda_0 > 0$.

To find the lower bound γ, we use the scaled feasible dual point $\theta(s) := s\theta_0$, where $s \in \mathbb{R}$ is constrained so that $\|X\theta(s)\|_\infty \le \lambda$, that is, $|s| \le \lambda/\lambda_0$. We then set γ according to

$$\gamma(\lambda) = \max_s \left\{ G_{\rm sq}(s\theta_0) \ : \ |s| \le \frac{\lambda}{\lambda_0} \right\} = \max_s \left\{ \beta_0 s - \frac{1}{2} s^2 \alpha_0 \ : \ |s| \le \frac{\lambda}{\lambda_0} \right\},$$

where $\alpha_0 := \theta_0^T\theta_0 > 0$, $\beta_0 := |y^T\theta_0|$. We obtain

$$\gamma(\lambda) = \frac{\beta_0^2}{2\alpha_0} \left(1 - \left(1 - \frac{\alpha_0}{\beta_0} \frac{\lambda}{\lambda_0} \right)_+^2 \right).$$

The test takes the form (10), with $\gamma(\lambda)$ given above:

$$\lambda > |x_k^T y| + D(\lambda)\|x_k\|_2, \quad D(\lambda)^2 := y^T y - 2\gamma(\lambda) \quad = \quad \min_s \left\{ \|y - s\theta_0\|_2^2 \ : \ |s| \leq \frac{\lambda}{\lambda_0} \right\} \qquad (13)$$
$$= \alpha_0 \left(\frac{\beta_0}{\alpha_0} - \frac{\lambda}{\lambda_0} \right)_+^2 + y^T y - \beta_0^2/\alpha_0.$$

There is a closed-form expression of the form $\lambda > \rho_k \lambda_0$, where $\rho_k \in [0, 1]$ depends on θ_0, y via α_0, β_0, λ_0, and also on $|y^T x_k|$, $\|x_k\|_2$. Note that, for given λ the value of $\gamma(\lambda)$ can be computed only once as we run the test through all the features. The complexity of running this test through all the features is again $O(mn)$, or less for sparse data.

Our result can be summarized as follows.

Theorem 2 (SAFE-LASSO) *Consider the LASSO problem $\mathcal{P}_{\rm sq}(\lambda)$ in (2). Let $\lambda_0 \geq \lambda$ be a value for which a dual point $\theta_0 \in \mathbb{R}^m$ that is optimal for $\mathcal{P}_{\rm sq}(\lambda_0)$ is known, so that in particular $\lambda_0 = \|X\theta_0\|_\infty \geq \lambda$. Denote by x_k the k-th row of the matrix X. The condition*

$$\lambda > |x_k^T y| + D(\lambda)\|x_k\|_2, \ \textit{with } D(\lambda) = \alpha_0 \left(\frac{\beta_0}{\alpha_0} - \frac{\lambda}{\lambda_0} \right)_+^2 + y^T y - \beta_0^2/\alpha_0, \qquad (14)$$

with $\alpha_0 := \theta_0^T \theta_0$, $\beta_0 := |y^T \theta_0|$, allows to safely remove the k-th feature.

The theorem reduces the basic SAFE-LASSO test of theorem 1, with the choice $\theta_0 = \theta_{\max} = -y$, $\lambda_0 = \lambda_{\max} = \|Xy\|_\infty$.

4.4 LASSO, with intercept

The LASSO with intercept case corresponds to a slight modification of the LASSO problem (2). More precisely, it corresponds to the general problem (1), with $f = f_{\rm sq}$, $a_i \in \mathbb{R}^n$, $i = 1, \ldots, n$ the data points, $c = -y$ is the (negative) response vector, and $b_i = 1$, $i = 1, \ldots, n$. Therefore, it expresses as

$$\phi(\lambda) := \min_{w,v} \ \frac{1}{2} \sum_{i=1}^m (a_i^T w + v - y_i)^2 + \lambda \|w\|_1. \qquad (15)$$

As before we define the data matrix $X = [a_1, \ldots, a_m]$.

Since the intercept coefficient v is not penalized, we can solve the problem with w fixed, and obtain the optimality condition

$$v = \overline{y} - \overline{X}^T w, \qquad (16)$$

where $\overline{y} = (1/m)\mathbf{1}^T y$, and $\overline{X} = (1/m)X\mathbf{1}$ with $\mathbf{1}$ the vector of ones of appropriate size. This implies that the LASSO with intercept problem reduces to one without intercept, with data (X, y) replaced with its *centered* version $(X_{\rm cent}, y_{\rm cent})$, where $X_{\rm cent} := X - \overline{X}\mathbf{1}^T$, $y_{\rm cent} := y - \overline{y}\mathbf{1}$.

In particular, safe feature elimination rules in the LASSO with intercept case are similar to the rules we obtained for the LASSO without intercept, with (x_k, y) replaced by its centered version $(x_k - \overline{x}_k\mathbf{1}, y - \overline{y}\mathbf{1})$, where $\overline{x}_k = (1/m)x_k^T\mathbf{1}$. Note that even if X is sparse, $X_{\rm cent}$ is generally not sparse; however, since $X_{\rm cent}$ is a rank-one modification of the matrix X, it is straightforward to exploit sparsity.

5. Hinge Loss

We turn to the sparse support vector machine classification problem:

$$\mathcal{P}_{\rm hi}(\lambda) \ : \ \ \phi_{\rm hi}(\lambda) := \min_{w,v} \sum_{i=1}^m (1 - y_i(z_i^T w + v))_+ + \lambda \|w\|_1, \qquad (17)$$

where $z_i \in \mathbb{R}^n$, $i = 1, \ldots, m$ are the data points, and $y \in \{-1, 1\}^m$ is the label vector. The above is a special case of the generic problem (1), where $f = f_{\rm hi}$ is the hinge loss, $b = y$, $c = 0$, and the feature matrix A is given by $A = [y_1 z_1, \ldots, y_m z_m]$, so that $x_k = [y_1 z_1(k), \ldots, y_m z_m(k)]^T$.

We denote by $\mathcal{I}_+, \mathcal{I}_-$ the set of indices corresponding to the positive and negative classes, respectively, and denote by $m_\pm = |\mathcal{I}_\pm|$ the associated cardinalities. We define $\underline{m} := \min(m_+, m_-)$. Finally, for a generic data vector x, we set $x^\pm = (x_i)_{i \in \mathcal{I}_\pm} \in \mathbb{R}^{m_\pm}$, $k = 1, \ldots, n$, the vectors corresponding to each one of the classes.

The dual problem takes the form

$$\phi_{\rm hi}(\lambda) := \max_\theta \ -\mathbf{1}^T\theta \ : \ \ -\mathbf{1} \leq \theta \leq 0, \ \ \theta^T y = 0, \ \ |\theta^T x_k| \leq \lambda, \ \ k = 1, \ldots, n. \tag{18}$$

5.1 Test, γ given

Let γ be a lower bound on $\phi_{\rm hi}(\lambda)$. The optimal value obtained upon setting $w = 0$ in (17) is given by

$$\min_v \ \sum_{i=1}^m (1 - y_i v)_+ = 2\min(m_+, m_-) := \gamma_{\max}. \tag{19}$$

Hence, without loss of generality, we may assume $0 \leq \gamma \leq \gamma_{\max}$.

The feature elimination test hinges on the quantity

$$\begin{array}{rcl} P_{\rm hi}(\gamma, x) & = & \max\limits_\theta \ \theta^T x \ : \ \ -\mathbf{1}^T\theta \geq \gamma, \ \ \theta^T y = 0, \ \ -\mathbf{1} \leq \theta \leq 0 \\ & = & \min\limits_{\mu > 0, \, \nu} \ -\gamma\mu + \mu \sum\limits_{i=1}^m f_{\rm hi}\left(\dfrac{x_i - \nu y_i}{\mu}\right) \\ & = & \min\limits_{\mu > 0, \, \nu} \ -\gamma\mu + \sum\limits_{i=1}^m (\mu + \nu y_i - x_i)_+. \end{array} \tag{20}$$

In appendix B.1, we show that for any x, the quantity $P(\gamma, x)$ is finite if and only if $0 \leq \gamma \leq \gamma_{\max}$, and can be computed in $O(m \log m)$, or less with sparse data, via a closed-form expression. That expression is simpler to state for $P_{\rm hi}(\gamma, -x)$:

$$\begin{array}{rcl} P_{\rm hi}(\gamma, -x) & = & \displaystyle\sum_{j=1}^{\lfloor \gamma/2 \rfloor} \bar{x}_j - (\frac{\gamma}{2} - \lfloor \frac{\gamma}{2} \rfloor)(\bar{x}_{\lfloor \gamma/2 \rfloor + 1})_+ + \sum_{j=\lfloor \gamma/2 \rfloor + 1}^{\underline{m}} (\bar{x}_j)_+, \ \ 0 \leq \gamma \leq \gamma_{\max} = 2\underline{m}, \\ & & \bar{x}_j := x^+_{[j]} + x^-_{[j]}, \ \ j = 1, \ldots, \underline{m}, \end{array}$$

with $x_{[j]}$ the j-th largest element in a vector x, and with the convention a sum over an empty index set is zero. Note that in particular, since $\gamma_{\max} = 2\underline{m}$:

$$P_{\rm hi}(\gamma_{\max}, -x) = \sum_{i=1}^{\underline{m}} (x^+_{[j]} + x^-_{[j]}).$$

5.2 SAFE-SVM theorem

Following the construction proposed in section 3.4 for the generic case, we select $\gamma = G_{\rm hi}(\theta)$, where the point θ is feasible for (18), and can found by the scaling method outlined in section 3.3, as follows. The method starts with the assumption that there is a value $\lambda_0 \geq \lambda$ for which we know the optimal value γ_0 of $\mathcal{P}_{\rm hi}(\lambda_0)$.

Specific choices for λ_0, γ_0. Let us first detail how we can find such values λ_0, γ_0.

We can set a value λ_0 such that $\lambda > \lambda_0$ ensures that $w = 0$ is optimal for the primal problem (17). The value that results in the least conservative test is $\lambda_0 = \lambda_{\max}$, where $\lambda_{\max}$ is the smallest value of λ above which $w = 0$ is optimal:

$$\lambda_{\max} := \min_{\theta} \|X\theta\|_\infty \ : \ -\theta^T \mathbf{1} \geq \gamma_{\max}, \ \theta^T y = 0, \ -\mathbf{1} \leq \theta \leq 0. \tag{21}$$

Since $\lambda_{\max}$ may be relatively expensive to compute, we can settle for an upper bound $\overline{\lambda}_{\max}$ on $\lambda_{\max}$. One choice for $\overline{\lambda}_{\max}$ is based on the test derived in the previous section: we ask that it passes for all the features when $\lambda = \overline{\lambda}_{\max}$ and $\gamma = \gamma_{\max}$. That is, we set

$$\begin{aligned} \overline{\lambda}_{\max} &= \max_{1\leq k\leq n} \max\left(P_{\text{hi}}(\gamma_{\max}, x_k), P_{\text{hi}}(\gamma_{\max}, -x_k)\right) \\ &= \max_{1\leq k\leq n} \max\left(\sum_{i=1}^m (x_k^+)_{[j]} + (x_k^-)_{[j]}, \sum_{i=1}^m (-x_k^+)_{[j]} + (-x_k^-)_{[j]}\right). \end{aligned} \tag{22}$$

By construction, we have $\overline{\lambda}_{\max} \geq \lambda_{\max}$, in fact:

$$\begin{aligned} \overline{\lambda}_{\max} &= \max_{1\leq k\leq n} \max_{\theta} |x_k^T\theta| \ : \ -\theta^T\mathbf{1} \geq \gamma_{\max}, \ \theta^T y = 0, \ -\mathbf{1} \leq \theta \leq 0 \\ &= \max_{\theta} \|X\theta\|_\infty \ : \ -\theta^T\mathbf{1} \geq \gamma_{\max}, \ \theta^T y = 0, \ -\mathbf{1} \leq \theta \leq 0, \end{aligned}$$

The two values $\lambda_{\max}, \overline{\lambda}_{\max}$ coincide if the feasible set is a singleton, that is, when $m_+ = m_-$. On the whole interval $\lambda_0 \in [\lambda_{\max}, \overline{\lambda}_{\max}]$, the optimal value of problem $\mathcal{P}_{\text{hi}}(\lambda_0)$ is $\gamma_{\max}$.

Dual scaling. The remainder of our analysis applies to any value λ_0 for which we know the optimal value $\gamma_0 \in [0, \gamma_{\max}]$ of the problem $\mathcal{P}_{\text{hi}}(\lambda_0)$.

Let θ_0 be a corresponding optimal dual point (as seen shortly, the value of θ_0 is irrelevant, as we will only need to know $\gamma_0 = -\mathbf{1}^T\theta_0$). We now scale the point θ_0 to make it feasible for $\mathcal{P}_{\text{hi}}(\lambda)$, where λ ($0 \leq \lambda \leq \lambda_0$) is given. The scaled dual point is obtained as $\theta = s\theta_0$, with s solution to (8). We obtain the optimal scaling $s = \lambda/\lambda_0$, and since $\gamma_0 = -\mathbf{1}^T\theta_0$, the corresponding bound is

$$\gamma(\lambda) = -\mathbf{1}^T(s\theta_0) = s\gamma_0 = \gamma_0\frac{\lambda}{\lambda_0}.$$

Our test takes the form

$$\lambda > \max\left(P_{\text{hi}}(\gamma(\lambda), x), P_{\text{hi}}(\gamma(\lambda), -x)\right).$$

Let us look at the condition $\lambda > P_{\text{hi}}(\gamma(\lambda), -x)$:

$$\exists\, \mu \geq 0,\ \nu \ : \ \lambda > -\gamma(\lambda)\mu + \sum_{i=1}^m (\mu + \nu y_i + x_i)_+,$$

which is equivalent to:

$$\lambda > \min_{\mu\geq 0,\nu} \frac{\displaystyle\sum_{i=1}^m (\mu + \nu y_i + x_i)_+}{1 + (\gamma_0/\lambda_0)\mu}.$$

The problem of minimizing the above objective function over variable ν has a closed-form solution. In appendix B.2, we show that for any vectors $x^\pm \in \mathbb{R}^{m_\pm}$, we have

$$\Phi(x^+, x^-) := \min_{\nu} \sum_{i=1}^{m_+} (x_i^+ + \nu)_+ + \sum_{i=1}^{m_-} (x_i^- - \nu)_+ = \sum_{i=1}^m (x_{[i]}^+ + x_{[i]}^-)_+,$$

with $x_{[j]}$ the j-th largest element in a vector x. Thus, the test becomes

$$\lambda > \min_{\mu \geq 0} \frac{\sum_{i=1}^{m} (2\mu + x_{[i]}^{+} + x_{[i]}^{-})_{+}}{1 + (\gamma_0/\lambda_0)\mu}.$$

Setting $\kappa = \lambda_0/(\lambda_0 + \gamma_0\mu)$, we obtain the following formulation for our test:

$$\lambda > \min_{0 \leq \kappa \leq 1} \sum_{i=1}^{\underline{m}} ((1-\kappa)\frac{2\lambda_0}{\gamma_0} + \kappa(x_{[i]}^{+} + x_{[i]}^{-}))_{+} = \frac{2\lambda_0}{\gamma_0} G(\frac{\gamma_0}{2\lambda_0}\overline{x}), \tag{23}$$

where $\overline{x}_i := x_{[i]}^{+} + x_{[i]}^{-}$, $i = 1, \ldots, \underline{m}$, and for $z \in \mathbb{R}^m$, we define

$$G(z) := \min_{0 \leq \kappa \leq 1} \sum_{i=1}^{m} (1 - \kappa + \kappa z_i)_{+}.$$

We show in appendix B.3 that $G(z)$ admits a closed-form expression, which can be computed in $O(d \log d)$, where d is the number of non-zero elements in vector z. By construction, the test removes all the features if we set $\lambda_0 = \lambda_{\max}$, $\gamma_0 = \gamma_{\max}$, and when $\lambda > \lambda_{\max}$.

Theorem 3 (SAFE-SVM) *Consider the SVM problem $\mathcal{P}_{\text{hi}}(\lambda)$ in (17). Denote by x_k the k-th row of the matrix $[y_1 z_1, \ldots, y_m z_m]$, and let $\mathcal{I}_{\pm} := \{i \,:\, y_i = \pm 1\}$, $m_{\pm} := |\mathcal{I}_{\pm}|$, $\underline{m} := \min(m_+, m_-)$, and $\gamma_{\max} := 2\underline{m}$. Let $\lambda_0 \geq \lambda$ be a value for which the optimal value $\gamma_0 \in [0, \gamma_{\max}]$ of $\mathcal{P}_{\text{sq}}(\lambda_0)$ is known. The following condition allows to remove the k-th feature vector x_k:*

$$\lambda > \frac{2\lambda_0}{\gamma_0} \max\left(G(\frac{\gamma_0}{2\lambda_0}\overline{x}_k), G(\frac{\gamma_0}{2\lambda_0}\underline{x}_k) \right), \tag{24}$$

where $(\overline{x}_k)_i := (x_k)_{[i]}^{+} + (x_k)_{[i]}^{-}$, $(\underline{x}_k)_i := (-x_k)_{[i]}^{+} + (-x_k)_{[i]}^{-}$, $i = 1, \ldots, \underline{m}$, and for $z \in \mathbb{R}^m$:

$$G(z) = \min_{z} \frac{1}{1-z} \sum_{i=1}^{p} (z_i - z)_{+} \;\; : \;\; z \in \{-\infty, 0, (z_j)_{j \,:\, z_j < 0}\}$$

A specific choice for λ_0 is $\overline{\lambda}_{\max}$ given by (22), with corresponding optimal value $\gamma_0 = \gamma_{\max}$.

6. Logistic Loss

We now consider the sparse logistic regression problem:

$$\mathcal{P}_{\text{lo}}(\lambda) \;\; : \;\; \phi_{\text{lo}}(\lambda) := \min_{w,v} \sum_{i=1}^{m} \log\left(1 + \exp(-y_i(z_i^T w + v))\right) + \lambda \|w\|_1, \tag{25}$$

with the same notation as in section 5. The dual problem takes the form

$$\phi_{\text{hi}}(\lambda) := \max_{\theta} \sum_{i=1}^{m} \left(\theta_i \log(-\theta_i) - (1+\theta_i)^T \log(1+\theta_i)\right) \;\; : \;\; \begin{array}{l} -\mathbf{1} \leq \theta \leq 0, \;\; \theta^T y = 0, \\ |\theta^T x_k| \leq \lambda, \;\; k = 1, \ldots, n. \end{array} \tag{26}$$

6.1 Test, γ given

Assume that we know a lower bound on the problem, $\gamma \leq \phi(\lambda)$. Since $0 \leq \phi(\lambda) \leq m \log 2$, we may assume that $\gamma \in [0, m \log 2]$ without loss of generality. We proceed to formulate problem (7). For given $x \in \mathbb{R}^m$, and $\gamma \in \mathbb{R}$, we have

$$P_{\log}(\gamma, x) \quad = \quad \min_{\mu>0,\, \nu} \; -\gamma\mu + \mu \sum_{i=1}^{m} f_{\log}\left(\frac{x_i + y_i\nu}{\mu}\right), \tag{27}$$

which can be computed in $O(m)$ by two-dimensional search, or by the dual interior-point method described in appendix. (As mentioned before, an alternative, resulting in a more conservative test, is to fix ν, for example $\nu = 0$.) Our test to eliminate the k-th feature takes the form

$$\lambda > T_{\log}(\gamma, x_k) := \max(P_{\log}(\gamma, x_k), P_{\log}(\gamma, -x_k)).$$

If γ is known, the complexity of running this test through all the features is $O(nm)$. (In fact, the terms in the objective function that correspond to zero elements of x are of two types, involving $f_{\log}(\pm\nu/\mu)$. This means that the effective dimension of problem (27) is the cardinality d of vector x, which in many applications is much smaller than m.)

6.2 Obtaining a dual feasible point

We can construct dual feasible points based on scaling one obtained by choice of a primal point (classifier weight) w_0. This in turn leads to other possible choices for the bound γ.

For $w_0 \in \mathbb{R}^n$ given, we solve the one-dimensional, convex problem

$$v_0 := \arg\min_b \sum_{i=1}^{m} f_{\log}(y_i x_i^T w_0 + y_i b).$$

This problem can be solved by bisection in $O(m)$ time Kim et al. (2007). At optimum, the derivative of the objective is zero, hence $y^T\theta_0 = 0$, where

$$\theta_0(i) := -\frac{1}{1 + \exp(y_i x_i^T w_0 + y_i v_0)}, \;\; i = 1, \ldots, m.$$

Now apply the scaling method seen before, and set γ by solving problem (8).

6.3 A specific example of a dual point

A convenient, specific choice in the above construction is to set $w_0 = 0$. Then, the intercept v_0 can be explicitly computed, as $v_0 = \log(m_+/m_-)$, where $m_\pm = |\{i : y_i = \pm 1\}|$ are the class cardinalities. The corresponding dual point θ_0 is

$$\theta_0(i) = \begin{cases} -\dfrac{m_-}{m} & (y_i = +1) \\ -\dfrac{m_+}{m} & (y_i = -1), \end{cases} \qquad i = 1, \ldots, m. \tag{28}$$

The corresponding value of λ_0 is (see Kim et al. (2007)):

$$\lambda_0 := \|A^T\theta_0\|_\infty = \max_{1 \leq k \leq n} |\theta_0^T x_k|.$$

We now compute $\gamma(\lambda)$ by solving problem (8), which expresses as

$$\gamma(\lambda) = \max_{|s| \leq \lambda/\lambda_0} G_{\log}(s\theta_0) = \max_{|s| \leq \lambda/\lambda_0} \; -m_+ f_{\log}^*(-s\frac{m_-}{m}) - m_- f_{\log}^*(-s\frac{m_+}{m}). \tag{29}$$

The above can be solved analytically: it can be shown that $s = \lambda/\lambda_0$ is optimal.

6.4 Solving the bisection problem

In this section, we are given $c \in \mathbb{R}^m$, $\gamma \in (0, m \log 2)$, and we consider the problem

$$F^* := \min_{\mu > 0} \; F(\mu) \quad := \quad -\gamma\mu + \mu \sum_{i=1}^{m} f_{\log}(c(i)/\mu). \tag{30}$$

Problem (30) corresponds to the problem (27), with ν set to a fixed value, and $c(i) = y_i x_i$, $i = 1, \ldots, m$. We assume that $c(i) \neq 0$ for every i, and that $\kappa := m \log 2 - \gamma > 0$. Observe that $F^* \leq F_0 := \lim_{\mu \to 0^+} \; F(\mu) = \mathbf{1}^T c_+$, where c_+ is the positive part of vector c.

To solve this problem via bisection, we initialize the interval of confidence to be $[0, \mu_u]$, with μ_u set as follows. Using the inequality $\log(1 + e^{-x}) \geq \log 2 - (1/2)x_+$, which is valid for every x, we obtain that for every $\mu > 0$:

$$F(\mu) \geq -\gamma\mu + \mu \sum_{i=1}^{m} \left(\log 2 - \frac{(c(i))_+}{2\mu} \right) = \kappa\mu - \frac{1}{2}\mathbf{1}^T c_+.$$

We can now identify a value μ_u such that for every $\mu \geq \mu_u$, we have $F(\mu) \geq F_0$: it suffices to ensure $\kappa\mu - (1/2)\mathbf{1}^T c_+ \geq F_0$, that is,

$$\mu \geq \mu_u := \frac{(1/2)\mathbf{1}^T c_+ + F_0}{\kappa} = \frac{3}{2} \frac{\mathbf{1}^T c_+}{m \log 2 - \gamma}.$$

6.5 Algorithm summary

An algorithm to check if a given feature can be removed from a sparse logistic regression problem works as follows.

Given: λ, k $(1 \leq k \leq n)$, $f_{\log}(x) = \log(1 + e^{-x})$, $f^*_{\log}(\vartheta) = (-\vartheta)\log(-\vartheta) + (\vartheta + 1)\log(\vartheta + 1)$.

1. Set $\lambda_0 = \max\limits_{1 \leq k \leq n} |\theta_0^T x_k|$, where $\theta_0(i) = -m_-/m$ $(y_i = +1)$, $\theta_0(i) = -m_+/m$ $(y_i = -1)$, $i = 1, \ldots, m$.

2. Set
$$\gamma(\lambda) := -m_+ f^*_{\log}(-\frac{\lambda}{\lambda_0}\frac{m_-}{m}) - m_- f^*_{\log}(-\frac{\lambda}{\lambda_0}\frac{m_+}{m}).$$

3. Solve via bisection a pair of one-dimensional convex optimization problems
$$P_\epsilon = \min_{\mu > 0} \; -\gamma(\lambda)\mu + \mu \sum_{i=1}^{m} f_{\log}(\epsilon y_i (x_k)_i / \mu) \;\; (\epsilon = \pm 1),$$
each with initial interval $[0, \mu_u]$, with
$$\mu_u = \frac{3}{2} \frac{\sum_{i=1}^{m} (\epsilon y_i (x_k)_i)_+}{m \log 2 - \gamma}.$$

4. If $\lambda > \max(P_+, P_-)$, the k-th feature can be safely removed.

7. Numerical results

In this section we report experiments[1] where we used SAFE prior to several sparse classification methods. We report on two kinds of experiments, which corresponds to the two main benefits of SAFE. One kind, in our opinion the most important, shows how memory limitations can be reduced, by allowing to treat larger data sets. The other seeks to measure how much computational time is saved.

In both cases, we have used a variety of available algorithms for sparse classification problems. We will use acronyms to refer to the following methods: IPM-LASSO stands for the Interior-Point Method for LASSO described in Kim et al. (2007); IPM-Logistic to that proposed in Koh et al. (2007) for sparse logistic regression; GLMNET corresponds to the Generalized Linear Model algorithm described in Friedman et al. (2010).

Since some methods (such as IPM ones) do not deliver exact zero coefficients, the issue arises as to how to evaluate the cardinality of classifier vectors. In appendix D, we discuss some issue related to thresholding the coefficients in IPM methods.

In our experiments, we have focused on LASSO problems for text classification. We use the acronym SAFE1 to refer to the basic safe test of theorem 1, and SAFE2 for safe test of theorem 2.

7.1 PubMed abstracts data

We have applied safe feature elimination procedure to a large data set which cannot be loaded into Matlab on our machine. This data set consists of medical journal abstracts represented in a bag-of-words format, where stop words have been eliminated, and capitalization removed. The number of features is $n = 127,025$ and the number of documents is $m = 1,000,000$. There is a total of $82,209,586$ non-zeros in the matrix, with an average of about 645 non-zeros per feature.

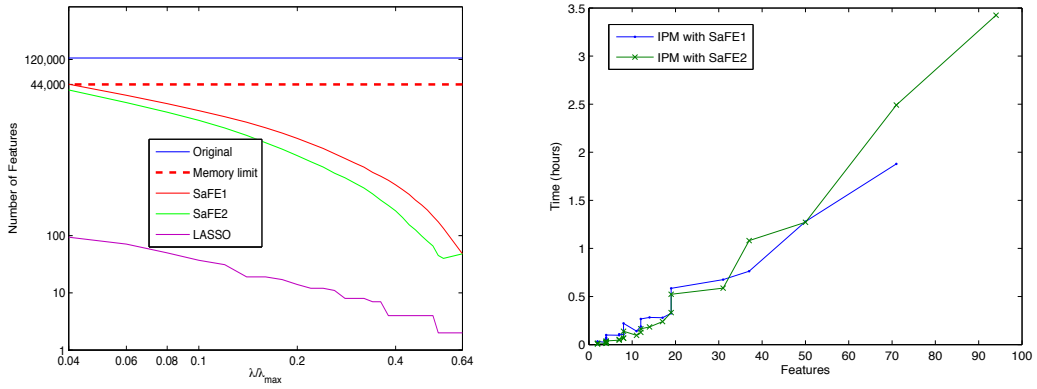

Figure 3: *Left panel:* In blue, the original number of features. In dashed-red, the number of features that can be loaded into matlab wihtout memory problems. In purple, the number of features obtained via IPM-LASSO, as a function of the parameter λ. In red, the number of features not eliminated by SAFE1. In green, the number of features not eliminated by SAFE2. *Right panel:* computational time needed to solve the IPM-LASSO after SAFE1 and SAFE2, as a function of the number of active features at the optimum

In some applications such as Gawalt et al. (2010), the goal is to learn a short list of terms that are predictive of the appearance of a given query term (say, "lung") in an abstract, producing a

1. In our experiments, we have used an Apple Mac Pro 64-bit workstation, with two 2.26 GHz Quad-Core Intel Xeon processors, 8 MB on-chip shared L3 cache per processor, with 6 GB SDRAM, operating at 1066 MHz.

summarization of the query term across many documents. To be manageable by a human reader, the list of predictive terms should be very short (say with at most 100 terms) with respect to the size of the dictionary (which is in our case about $130,000$). To produce such a short list, we use sparse supervised learning with large values of the penalty parameter λ.

The LASSO problem for the full-sized matrix can not be solved using IPM-LASSO or any other method because of memory limitations. However, for large values of the penalty parameter λ, the SAFE methods eliminate enough features for the data matrix to be loaded and the LASSO problem to be solved.

Figure 3 shows that SAFE methods remove enough features to solve the LASSO problem. The left panel shows how many features we eliminate as a function of the ratio $\lambda/\lambda_{\max}$, over a range such that the data matrix can be loaded. SAFE1 reaches the memory limit at $\lambda/\lambda_{\max} = 0.04$ The right panel shows the computational time needed to solve the IPM-LASSO, for the data matrix obtained after SAFE1 and SAFE2. Using SAFE2 we obtain 94 features at the optimum for $\lambda/\lambda_{\max} = 0.04$.

7.2 New York Times headlines data

This data set involves headlines from *The New York Times,* spanning a period of about 20 years (from 1985 to 2007). The raw text has been processed into numerical form using a bag-of-words representation. The number of features is $n = 159,943$ and the number of documents is $m = 3,241,260$. There is a total of $14,083,676$ non-zeros in the matrix, with an average of about 90 non-zeros per feature (word).

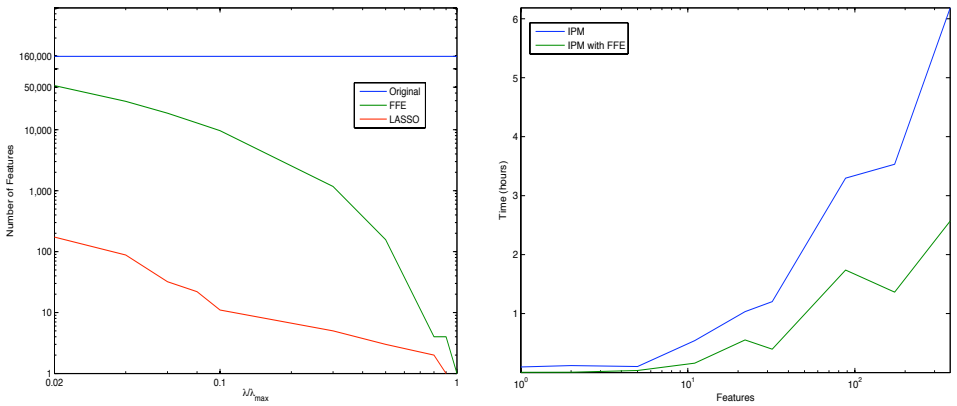

Figure 4: *Left panel:* In blue, the original number of features. In red, the number of features obtained via IPM-LASSO, as a function of of the ratio $\lambda/\lambda_{\max}$. In green, the number of features not eliminated by SAFE1. *Right panel:* computational time for solving the IPM-LASSO before and after SAFE1 as function of the number of active features at the optimum

Results on the whole data set. Figure 4 shows that SAFE1 is effective at removing a very large number of features. The left panel shows how many features we eliminate as a function of the ratio $\lambda/\lambda_{\max}$, over a range such that LASSO ends up with about 375 features or less; we end up reducing the number of features by a third or more. For 50 features the size reduction factor is one order of magnitude. The right panel shows how much computational time is saved as a result: we compare the time needed to solve the IPM-LASSO, for the full-sized data matrix, and for the matrix obtained after SAFE1 (the timing results include the time required to perform the test, which is negligible). For 375 features, we slash the time needed to solve the LASSO problem by a factor of 3; that ratio averages to about 2, and peaks to about 30 when 5 features are required.

Results on multiple subsets. In Gawalt et al. (2010), the objective was to exhibit a list of N features depicting the image of some topic (such as "China" for instance) in news media (typical values of N are 20, 50 and 100). Our aim was here to assess whether the pruning technique might be useful in this context. Given a data set - that is a response variable, or topic, and a matrix of features - several methods can be used to exhibit the desired list of N relevant features. First, we can solve a sparse regression model on a predefined grid of λ values, of the form $[\lambda_{\min}, \ldots, \lambda_{\max}]$, where $\lambda_{\min}$ expresses as a fraction of $\lambda_{\max}$ (typical values are $\lambda_{\min} = \lambda_{\max}/100$ or $\lambda_{\min} = \lambda_{\max}/1000$). This grid describes a schedule of λ values. Another approach, that can be computationally much faster, relies on a binary search: given values of $\lambda_{\max}$ and $\lambda_{\min}$ (where $\lambda_{\min}$ has to be such that the model obtained by solving the problem with $\lambda = \lambda_{\min}$ returns more than N non-zero coefficients), we solve the problem for $\lambda = \lambda_1 := (\lambda_{\min} + \lambda_{\max})/2$. If the corresponding number of non-zero coefficients is greater than N, then we solve the problem for $\lambda = \lambda_2' = (\lambda_1 + \lambda_{\max})/2$, otherwise we solve the problem for $\lambda = \lambda_2'' = (\lambda_1 + \lambda_{\min})/2$, and so on. We generally exit the loop when the number of non-zero coefficients lies between, say, $N - 2$ and $N + 2$. The choice of $\lambda_{\min}$ is crucial for computation times matters: the smaller $\lambda_{\min}$, the longer the whole process (since computation time of most ℓ_1-penalized methods dramatically increase as λ approaches 0). In addition to this "standard" binary search, we also considered a customized binary search: starting from $\lambda^{(1)} = \lambda_{\max}$, for $k = 2, \ldots, K$, we solve the problem for $\lambda^{(k)} = \lambda^{(k-1)}/\tau$, for some $\tau > 1$, until a model with more than 20 non-zero coefficient is obtained. We then set $\lambda_{\min} = \lambda^{(K)}$ and perform the standard binary search on the range $[\lambda_{\min}, \lambda_{\max}]$. In our experiments, we set $\tau = 5$. Note that the $K-1$ steps needed to get the value $\lambda^{(K)}$ are generally cheap and, if τ is small enough, $\lambda^{(K)}$ is a sharp lower-bound on the desired λ value. For these problems involving several values of λ, we have used the method outlined in section 3.5, where we take advantage of previously computed bounds as we proceed in our λ-search.

Rather than assessing the pruning method once on the whole dataset, we decided here to split this data set and analyze each year separately. This resulted in 23 subsets over which we compared the computational times needed to solve (*i*) IPM-LASSO, (*ii*) IPM-LASSO after SAFE1 and (*iii*) IPM-LASSO after SAFE2. These three methods were compared under the three settings described above: a given schedule of λ, the standard binary search and the customized binary search. Summary of the results obtained over the 23 subsets for topic "China", are given in Figures 5 and 6. When appropriate, $\lambda_{\min}$ was set to $\lambda_{\max}/1000$. Lasso needed, on average, one hour and a half. to compute the 50 problems corresponding to values of λ on the grid of 50 equally-spaced values (on a log-scale) $[\lambda_{\min}, \ldots, \lambda_{\max}]$. As for the standard binary search, it needed 6 minutes, 11 minutes, and 16 minutes for reaching a model with 20, 50 and 100 features respectively. For the customized binary search, these values reduced up to 4 minutes, 7 minutes, and 13 minutes. respectively.

Overall, we observed dramatic computational time savings when using SAFE2 (and, to a lower extent, SAFE1): in most situations, the saving correspond to 30 to 60% of the computational time needed for plain IPM-LASSO.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (under grants SES-0835531, CMMI-0969923) and Google. Useful comments from Bin Yu and Jinzhu Jia have greatly improved this report.

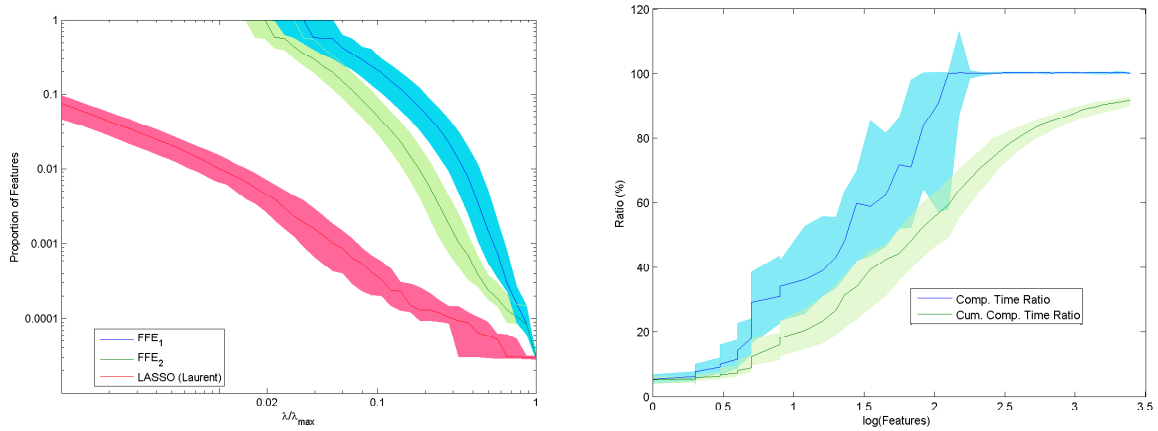

Figure 5: *Left panel:* In red, the proportion of features (as a fraction of the total number of features) obtained via LASSO, as a function of the parameter $\lambda/\lambda_{\max}$. In blue, the number of features not eliminated by SAFE1. In green, the number of features not eliminated by SAFE2. Solid lines represent the means while shaded area represent the ranges obtained on the 23 subsets. *Right panel:* computational savings obtained by solving IPM-LASSO after SAFE2, as a function of the number of active features at the optimum: percent ratio of the computational time needed to solve IPM-LASSO after SAFE2 to the computational time needed to solve IPM-LASSO. In blue, computational time for each number of active features at optimum. In green, cumulative computational time up to each number of active features at optimum.

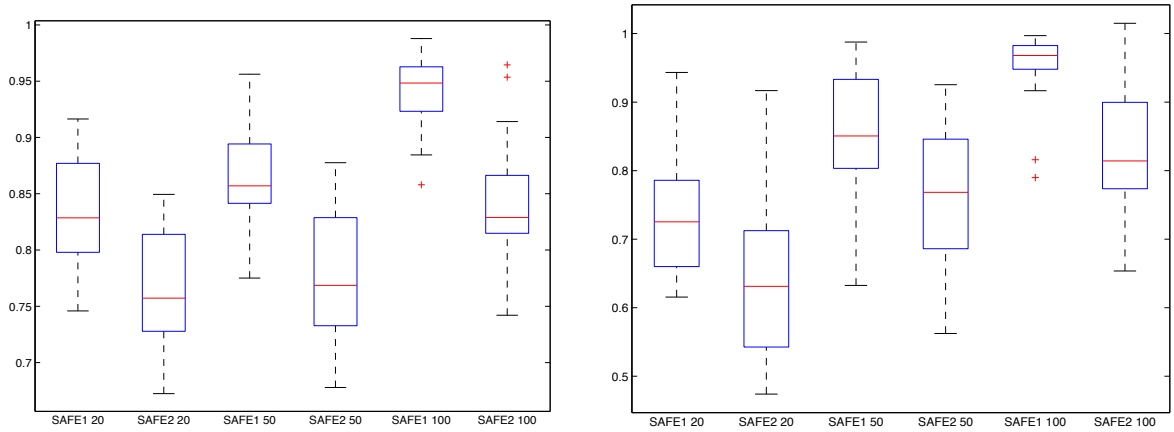

Figure 6: Computational time savings when solving the standard binary search (*left panel*) and the customized binary search (*right panel*). Distribution of the ratio of the time needed for lasso to the time needed for IPM-LASSO after SAFE1 and IPM-LASSO after SAFE2, when the objective is to obtain $N = 20, 50$ or 100 active features at optimum.

Appendix A. Expression of $P(\gamma, x)$

We show that the quantity $P(\gamma, x)$ defined in (6) can be expressed in dual form (7). This is a simple consequence of duality:

$$\begin{aligned} P(\gamma,x) &= \max_{\theta} \theta^T x \ : \ G(\theta) \geq \gamma, \ \ \theta^T b = 0 \\ &= \max_{\theta} \min_{\mu>0,\,\nu} \theta^T x + \mu(G(\theta) - \gamma) - \nu\theta^T b \\ &= \min_{\mu>0,\,\nu} \max_{\theta} \theta^T x + \mu(-y^T\theta - \sum_{i=1}^{m} f^*(\theta(i)) - \gamma) - \nu\theta^T b \\ &= \min_{\mu>0,\,\nu} -\gamma\mu + \max_{\theta} \theta^T(x - \mu y - \nu z) - \mu\sum_{i=1}^{m} f^*(\theta(i)) \end{aligned}$$

Appendix B. SAFE test for SVM

In this section, we examine various optimization problems involving polyhedral functions in one or two variables, which arise in section 5.1 for the computation of $P_{\rm hi}(\gamma, x)$ as well as in the SAFE-SVM theorem of section 5.2.

B.1 Computing $P_{\rm hi}(\gamma, x)$

We first focus on the specific problem of computing the quantity defined in (20). To simplify notation, we will consider the problem of computing $P_{\rm hi}(\gamma, -x)$, that is:

$$P_{\rm hi}(\gamma, -x) = \min_{\mu \geq 0,\, \nu} \; -\gamma\mu + \sum_{i=1}^{m} (\mu + \nu y_i + x_i)_+, \tag{31}$$

where $y \in \{-1, 1\}^m$, $x \in \mathbb{R}^m$ and γ are given, with $0 \leq \gamma \leq \gamma_0 := 2\min(m_+, m_-)$. Here, $\mathcal{I}_\pm := \{i : y_i = \pm 1\}$, and $x^+ = (x_i)_{i \in \mathcal{I}_+}$, $x^- = (x_i)_{i \in \mathcal{I}_-}$, $m_\pm = |\mathcal{I}_\pm|$, and $\underline{m} = \min(m_+, m_-)$. Without loss of generality, we assume that both x^+, x^- are both sorted in descending order: $x_1^\pm \geq \ldots \geq x_{m_\pm}^\pm$.

Using $\alpha = \mu + \nu$, $\beta = \mu - \nu$, we have

$$\begin{aligned}
P_{\rm hi}(\gamma, -x) &= \min_{\alpha+\beta \geq 0} \; -\frac{\gamma}{2}(\alpha+\beta) + \sum_{i=1}^{m_+} (x_i^+ + \alpha)_+ + \sum_{i=1}^{m_-} (x_i^- + \beta)_+ \\
&= \min_{\alpha,\,\beta} \max_{t \geq 0} \; -\frac{\gamma}{2}(\alpha+\beta) + \sum_{i=1}^{m_+} (x_i^+ + \alpha)_+ + \sum_{i=1}^{m_-} (x_i^- + \beta)_+ - t(\alpha+\beta) \\
&= \max_{t \geq 0} \min_{\alpha,\,\beta} \; -(\frac{\gamma}{2} + t)(\alpha+\beta) + \sum_{i=1}^{m_+} (x_i^+ + \alpha)_+ + \sum_{i=1}^{m_-} (x_i^- + \beta)_+ \\
&= \max_{t \geq 0} \; F(\frac{\gamma}{2} + t, x^+) + F(\frac{\gamma}{2} + t, x^-),
\end{aligned} \tag{32}$$

where, for $h \in \mathbb{R}$ and $x \in \mathbb{R}^p$, $x_1 \geq \ldots \geq x_p$, we set

$$F(h, x) := \min_z \; -hz + \sum_{i=1}^{p} (z + x_i)_+, \tag{33}$$

Expression of the function F. If $h > p$, then with $z \to +\infty$ we obtain $F(h, x) = -\infty$. Similarly, if $h < 0$, then $z \to -\infty$ yields $F(h, x) = -\infty$. When $0 \leq h \leq p$, we proceed by expressing F in dual form:

$$F(h, x) = \max_u \; u^T x \;:\; 0 \leq u \leq \mathbf{1}, \;\; u^T \mathbf{1} = h.$$

If $h = p$, then the only feasible point is $u = \mathbf{1}$, so that $F(p, x) = \mathbf{1}^T x$. If $0 \leq h < 1$, choosing $u_1 = h$, $u_2 = \ldots = u_p = 0$, we obtain the lower bound $F(h, x) \geq h x_1$, which is attained with $z = -x_1$.

Assume now that $1 \leq h < p$. Let $h = q + r$, with $q = \lfloor h \rfloor$ the integer part of h, and $0 \leq r < 1$. Choosing $u_1 = \ldots = u_q = 1$, $u_{q+1} = r$, we obtain the lower bound

$$F(h, x) \geq \sum_{j=1}^{q} x_j + r x_{q+1},$$

which is attained by choosing $z = -x_{q+1}$ in the expression (33).

To summarize:

$$F(h,x) = \begin{cases} hx_1 & \text{if } 0 \le h < 1, \\ \sum_{j=1}^{\lfloor h \rfloor} x_j + (h - \lfloor h \rfloor) x_{\lfloor h \rfloor + 1} & \text{if } 1 \le h < p, \\ \sum_{j=1}^{p} x_j & \text{if } h = p, \\ -\infty & \text{otherwise.} \end{cases} \tag{34}$$

A more compact expression, valid for $0 \le h \le p$ if we set $x_{p+1} = x_p$ and assume that a sum over an empty index sets is zero, is

$$F(h,x) = \sum_{j=1}^{\lfloor h \rfloor} x_j + (h - \lfloor h \rfloor) x_{\lfloor h \rfloor + 1}, \;\; 0 \le h \le p.$$

Note that $F(\cdot, x)$ is the piece-wise linear function that interpolates the sum of the h largest elements of x at the integer break points $h = 0, \ldots, p$.

Expression of $P_{\rm hi}(\gamma, -x)$. We start with the expression found in (32):

$$P_{\rm hi}(\gamma, -x) = \max_{t \ge 0} \; F(\frac{\gamma}{2} + t, x^+) + F(\frac{\gamma}{2} + t, x^-).$$

Since the domain of $F(\cdot, x^+) + F(\cdot, x^-)$ is $[0, \underline{m}]$, and with $0 \le \gamma/2 \le \gamma_0/2 = \underline{m}$, we get

$$P_{\rm hi}(\gamma, -x) = \max_{\gamma/2 \le h \le \underline{m}} G(h, x^+, x^-) := F(h, x^+) + F(h, x^-).$$

Since $F(\cdot, x)$ with $x \in \mathbb{R}^p$ is a piece-wise linear function with break points at $0, \ldots, p$, a maximizer of $G(\cdot, x^+, x^-)$ over $[\gamma/2, \underline{m}]$ lies in $\{\gamma/2, \lfloor \gamma/2 \rfloor + 1, \ldots, \underline{m}\}$. Thus,

$$P_{\rm hi}(\gamma, -x) = \max\left(G(\frac{\gamma}{2}, x^+, x^-), \max_{h \in \{\lfloor \gamma/2 \rfloor + 1, \ldots, \underline{m}\}} G(h, x^+, x^-) \right).$$

Let us examine the second term, and introduce the notation $\bar{x}_j := x_j^+ + x_j^-$, $j = 1, \ldots, \underline{m}$:

$$\begin{aligned} \max_{h \in \{\lfloor \gamma/2 \rfloor + 1, \ldots, \underline{m}\}} G(h, x^+, x^-) &= \max_{h \in \{\lfloor \gamma/2 \rfloor + 1, \ldots, \underline{m}\}} \sum_{j=1}^{h} (x_j^+ + x_j^-) \\ &= \sum_{j=1}^{\lfloor \gamma/2 \rfloor + 1} \bar{x}_j + \sum_{j=\lfloor \gamma/2 \rfloor + 2}^{\underline{m}} (\bar{x}_j)_+, \end{aligned}$$

with the convention that sums over empty index sets are zero. Since

$$G(\frac{\gamma}{2}, x^+, x^-) = \sum_{j=1}^{\lfloor \gamma/2 \rfloor} \bar{x}_j + (\frac{\gamma}{2} - \lfloor \frac{\gamma}{2} \rfloor) \bar{x}_{\lfloor \gamma/2 \rfloor + 1},$$

we obtain

$$P_{\rm hi}(\gamma, -x) = \sum_{j=1}^{\lfloor \gamma/2 \rfloor} \bar{x}_j + \max\left((\frac{\gamma}{2} - \lfloor \frac{\gamma}{2} \rfloor) \bar{x}_{\lfloor \gamma/2 \rfloor + 1}, \bar{x}_{\lfloor \gamma/2 \rfloor + 1} + \sum_{j=\lfloor \gamma/2 \rfloor + 2}^{\underline{m}} (\bar{x}_j)_+ \right).$$

An equivalent expression is:

$$\begin{aligned} P_{\rm hi}(\gamma,-x) &= \sum_{j=1}^{\lfloor\gamma/2\rfloor}\bar{x}_j - (\frac{\gamma}{2}-\lfloor\frac{\gamma}{2}\rfloor)(-\bar{x}_{\lfloor\gamma/2\rfloor+1})_+ + \sum_{j=\lfloor\gamma/2\rfloor+1}^{\underline{m}}(\bar{x}_j)_+,\ \ 0\le\gamma\le 2\underline{m},\\ &\bar{x}_j := x_j^+ + x_j^-,\ \ j=1,\ldots,\underline{m}. \end{aligned}$$

The function $P_{\rm hi}(\cdot,-x)$ linearly interpolates the values obtained for $\gamma = 2q$ with q integer in $\{0,\ldots,\underline{m}\}$:

$$P_{\rm hi}(2q,-x) = \sum_{j=1}^{q}\bar{x}_j + \sum_{j=q+1}^{\underline{m}}(\bar{x}_j)_+.$$

B.2 Computing $\Phi(x^+,x^-)$

Let us consider the problem of computing

$$\Phi(x^+,x^-) := \min_{\nu}\ \sum_{i=1}^{m_+}(x_i^+ + \nu)_+ + \sum_{i=1}^{m_-}(x_i^- - \nu)_+,$$

with $x^\pm\in\mathbb{R}^{m_\pm}$, $x_1^\pm\ge\ldots\ge x_{m_\pm}^\pm$, given. We can express $\Phi(x^+,x^-)$ in terms of the function F defined in (33):

$$\begin{aligned} \Phi(x^+,x^-) &= \min_{\nu_+,\nu_-}\ \sum_{i\in\mathcal{I}_+}(x_i^+ + \nu^+)_+ + \sum_{i\in\mathcal{I}_-}(x_i^- - \nu^-)_+\ :\ \nu^+=\nu^-\\ &= \max_h\ \min_{\nu^+,\nu^-}\ -h(\nu^+-\nu^-) + \sum_{i\in\mathcal{I}_+}(x_i^+ + \nu^+)_+ + \sum_{i\in\mathcal{I}_-}(x_i^- - \nu^-)_+\\ &= \max_h\ \min_{\nu^+,\nu^-}\ -h\nu^+ + \sum_{i\in\mathcal{I}_+}(x_i^+ + \nu^+)_+ + h\nu^- + \sum_{i\in\mathcal{I}_-}(x_i^- - \nu^-)_+\\ &= \max_h\ \left(\min_{\nu}\ -h\nu + \sum_{i\in\mathcal{I}_+}(x_i^+ + \nu)_+\right) + \left(\min_{\nu}\ -h\nu + \sum_{i\in\mathcal{I}_-}(x_i^- + \nu)_+\right)\ \ (\nu_+=-\nu_-=\nu)\\ &= \max_h\ F(h,x^+) + F(h,x^-)\\ &= \max_{0\le h\le\underline{m}}\ F(h,x^+) + F(h,x^-)\\ &= \max(A,B,C), \end{aligned}$$

where F is defined in (33), and

$$A = \max_{0\le h<1}\ F(h,x^+)+F(h,x^-),\ \ B := \max_{1\le h<\underline{m}}\ F(h,x^+)+F(h,x^-)),\ \ C = F(\underline{m},x^+)+F(\underline{m},x^-).$$

We have

$$A := \max_{0\le h<1}\ F(h,x^+)+F(h,x^-) = \max_{0\le h<1}\ h(x_1^+ + x_1^-) = (x_1^+ + x_1^-)_+.$$

Next:

$$\begin{aligned} B &= \max_{1\le h<\underline{m}}\ F(h,x^+)+F(h,x^-)\\ &= \max_{q\in\{1,\ldots,\underline{m}-1\},r\in[0,1[}\ \sum_{i=1}^{q}(x_i^+ + x_i^-) + r(x_{q+1}^+ + x_{q+1}^-)\\ &= \max_{q\in\{1,\ldots,\underline{m}-1\}}\ \sum_{i=1}^{q}(x_i^+ + x_i^-) + (x_{q+1}^+ + x_{q+1}^-)_+\\ &= (x_1^+ + x_1^-) + \sum_{i=2}^{\underline{m}}(x_i^+ + x_i^-)_+. \end{aligned}$$

Observe that

$$B \geq C = \sum_{i=1}^{\underline{m}} (x_i^+ + x_i^-).$$

Moreover, if $(x_1^+ + x_1^-) \geq 0$, then $B = \sum_{i=1}^{\underline{m}} (x_i^+ + x_i^-)_+ \geq A$. On the other hand, if $x_1^+ + x_1^- \leq 0$, then $x_i^+ + x_i^- \leq 0$ for $2 \leq j \leq \underline{m}$, and $A = \sum_{i=1}^{\underline{m}} (x_i^+ + x_i^-)_+ \geq x_1^+ + x_1^- = B$. In all cases,

$$\Phi(x^+, x^-) = \max(A, B, C) = \sum_{i=1}^{\underline{m}} (x_i^+ + x_i^-)_+.$$

B.3 SAFE-SVM test

Now we consider the problem that arises in the SAFE-SVM test (23):

$$G(z) := \min_{0 \leq \kappa \leq 1} \sum_{i=1}^{p} (1 - \kappa + \kappa z_i)_+,$$

where $z \in \mathbb{R}^p$ is given. (The SAFE-SVM condition (23) involves $z_i = \gamma_0/(2\lambda_0)(x_{[i]}^+ + x_{[i]}^-)$, $i = 1, \ldots, p := \underline{m}$.) We develop an algorithm to compute the quantity $G(z)$, the complexity of which grows as $O(d \log d)$, where d is (less than) the number of non-zero elements in z.

Define $\mathcal{I}_\pm = \{i \,:\, \pm z_i > 0\}$, $k := |\mathcal{I}_+|$, $h := |\mathcal{I}_-|$, $l = \mathcal{I}_0$, $l := |\mathcal{I}_0|$.

If $k = 0$, $\mathcal{I}_+$ is empty, and $\kappa = 1$ achieves the lower bound of 0 for $G(z)$. If $k > 0$ and $h = 0$, that is, $k + l = p$, then $\mathcal{I}_-$ is empty, and an optimal κ is attained in $\{0, 1\}$. In both cases ($\mathcal{I}_+$ or $\mathcal{I}_-$ empty), we can write

$$G(z) = \min_{\kappa \in \{0,1\}} \sum_{i=1}^{p} (1 - \kappa + \kappa z_i)_+ = \min(p, S_+), \quad S_+ := \sum_{i \in \mathcal{I}_+} z_i,$$

with the convention that a sum over an empty index set is zero.

Next we proceed with the assumption that $k \neq 0$ and $h \neq 0$. Let us re-order the elements of $\mathcal{I}_-$ in decreasing fashion, so that $z_i > 0 = z_{k+1} = \ldots = z_{k+l} > z_{k+l+1} \geq \ldots \geq z_p$, for every $i \in \mathcal{I}_+$. (The case when $\mathcal{I}_0$ is empty will be handled simply by setting $l = 0$ in our formulae.) We have

$$G(z) = k + l + \min_{0 \leq \kappa \leq 1} \left\{ \kappa\alpha + \sum_{i=k+l+1}^{p} (1 - \kappa + \kappa z_i)_+ \right\},$$

where, $\alpha := S_+ - k - l$. The minimum in the above is attained at $\kappa = 0, 1$ or one of the break points $1/(1 - z_j) \in (0, 1)$, where $j \in \{k + l + 1, \ldots, p\}$. At $\kappa = 0, 1$, the objective function of the original problem takes the values S_+, p, respectively. The value of the same objective function at the break

point $\kappa = 1/(1-z_j)$, $j = k+l+1, \ldots, p$, is $k+l+G_j(z)$, where

$$\begin{aligned} G_j(z) &:= \frac{\alpha}{1-z_j} + \sum_{i=k+l+1}^{p} \left(\frac{z_i - z_j}{1-z_j}\right)_+ \\ &= \frac{\alpha}{1-z_j} + \frac{1}{1-z_j} \sum_{i=k+l+1}^{j-1} (z_i - z_j) \\ &= \frac{1}{1-z_j} \left(\alpha - (j-k-l-1)z_j + \sum_{i=k+l+1}^{j-1} z_i\right) \\ &= \frac{1}{1-z_j} \left(S_+ - (j-1)z_j - (k+l)(1-z_j) + \sum_{i=k+l+1}^{j-1} z_i\right) \\ &= -(k+l) + \frac{1}{1-z_j} \left(\sum_{i=1}^{j-1} z_i - (j-1)z_j\right). \end{aligned}$$

This allows us to write

$$G(z) = \min\left(p, \sum_{i=1}^{k} z_i, \min_{j\in\{k+l+1,\ldots,p\}} \frac{1}{1-z_j}\left(\sum_{i=1}^{j-1} z_i - (j-1)z_j\right)\right).$$

The expression is valid when $k+l=p$ ($h=0$, $\mathcal{I}_-$ is empty), $l=0$ ($\mathcal{I}_0$ is empty), or $k=0$ ($\mathcal{I}_+$ is empty) with the convention that the sum (resp. minimum) over an empty index set is 0 (resp. $+\infty$).

We can summarize the result with the compact formula:

$$G(z) = \min_z \frac{1}{1-z} \sum_{i=1}^{p} (z_i - z)_+ \ : \ z \in \{-\infty, 0, (z_j)_{j\,:\,z_j<0}\}.$$

Let us detail an algorithm for computing $G(z)$. Assume $h > 0$. The quantity

$$\underline{G}(z) := \min_{k+l+1\le j\le p} (G_j(z))$$

can be evaluated in less than $O(h)$, via the following recursion:

$$\begin{aligned} G_{j+1}(z) &= \frac{1-z_j}{1-z_{j+1}} G_j(z) - j\frac{z_{j+1}-z_j}{1-z_{j+1}} \quad , \quad j = k+l+1, \ldots, p, \\ \underline{G}_{j+1}(z) &= \min(\underline{G}_j(z), G_{j+1}(z)) \end{aligned} \tag{35}$$

with initial values

$$G_{k+l+1}(z) = \underline{G}_{k+l+1}(z) = \frac{1}{1-z_{k+l+1}} \left(\sum_{i=1}^{k+l} z_i - (k+l)z_{k+l+1}\right).$$

On exit, $\underline{G}(z) = \underline{G}_p$.

Our algorithm is as follows.

Algorithm for the evaluation of $G(z)$.

1. Find the index sets $\mathcal{I}_+$, $\mathcal{I}_-$, $\mathcal{I}_0$, and their respective cardinalities k, h, l.
2. If $k=0$, set $G(z)=0$ and exit.

3. Set $S_+ = \sum_{i=1}^{k} z_i$.

4. If $h = 0$, set $G(z) = \min(p, S_+)$, and exit.

5. If $h > 0$, order the negative elements of z, and evaluate $\underline{G}(z)$ by the recursion (35). Set $G(z) = \min(p, S_+, \underline{G}(z))$ and exit.

The complexity of evaluating $G(z)$ thus grows in $O(k + h \log h)$, which is less than $O(d \log d)$, where $d = k + h$ is the number of non-zero elements in z.

Appendix C. Computing $P_{\log}(\gamma, x)$ via an interior-point method

We consider the problem (27) which arises with the logistic loss. We can use a generic interior-point method Boyd and Vandenberghe (2004), and exploit the decomposable structure of the dual function $G_{\log}$. The algorithm is based on solving, via a variant of Newton's method, a sequence of linearly constrained problems of the form

$$\min_{\theta} \ \tau x^T \theta + \log(G_{\log}(\theta) - \gamma) + \sum_{i=1}^{m} \log(-\theta - \theta^2) \ : \ z^T \theta = 0,$$

where $\tau > 0$ is a parameter that is increased as the algorithm progresses, and the last terms correspond to domain constraints $\theta \in [-1, 0]^m$. As an initial point, we can take the point θ generated by scaling, as explained in section 3. Each iteration of the algorithm involves solving a linear system in variable δ, of the form $H\delta = h$, with H is a rank-two modification to the Hessian of the objective function in the problem above. It is easily verified that the matrix H has a "diagonal plus rank-two" structure, that is, it can be written as $H = D - gg^T - vv^T$, where the $m \times m$ matrix D is diagonal and $g, v \in \mathbb{R}^m$ are computed in $O(m)$. The matrix H can be formed, as the associated linear system solved, in $O(m)$ time. Since the number of iterations for this problem with two constraints grows as $\log(1/\epsilon)O(1)$, the total complexity of the algorithm is $\log(1/\epsilon)O(m)$ (ϵ is the absolute accuracy at which the interior-point method computes the objective). We note that memory requirements for this method also grow as $O(m)$.

Appendix D. On thresholding methods for LASSO

Sparse classification algorithms may return a classifier vector w with many small, but not exactly zero, elements. This implies that we need to choose a thresholding rule to decide which elements to set to zero. In this section, we discuss an issue related to the thresholding rule originally proposed for the IPM-Logistic algorithm in Koh et al. (2007), and propose a new thresholding rule.

The KKT thresholding rule. Recall that the primal problem for LASSO is

$$\phi(\lambda) = \min_{w} \frac{1}{2} \|X^T w - y\|_2^2 + \lambda \|w\|_1. \tag{36}$$

Observing that the KKT conditions imply that, at optimum, $(X(X^T w - y))_k = \lambda \text{sign}(w_k)$, with the convention $\text{sign}(0) \in [-1, 1]$, and following the ideas of Koh et al. (2007), the following thresholding rule can be proposed: at optimum, set component w_k to 0 whenever

$$|(X(X^T w - y))_k| \leq 0.9999\lambda. \tag{37}$$

We refer to this rule as the "KKT" rule.

The IPM-LASSO algorithm takes as input a "duality gap" parameter ϵ, which controls the relative accuracy on the objective. When comparing the IPM code results with other algorithms

such as GLMNET, we observed chaotic behaviors when applying the KKT rule, especially when the duality gap parameter ϵ was not small enough. More surprisingly, when this parameter is not small enough, some components w_k with absolute values not close to 0 can be thresholded. This suggests that the KKT rule should only be used for problems solved with a small enough duality gap ϵ. However, setting the duality gap to a small value can dramatically slow down computations. In our experiments, changing the duality gap from $\epsilon = 10^{-4}$ to 10^{-6} (resp. 10^{-8}) increased the computational time by 30% to 40% (resp. 50 to 100%).

An alternative method. We propose an alternative thresholding rule, which is based on controlling the perturbation of the objective function that is induced by thresholding.

Assume that we have solved the LASSO problem above, with a given duality gap parameter ϵ. If we denote by w^* the classifier vector delivered by the IPM-LASSO algorithm, w^* is ϵ-sub-optimal, that is, achieves a value

$$\phi^* = \frac{1}{2}\|X^T w^* - y\|_2^2 + \lambda\|w^*\|_1,$$

with $0 \leq \phi^* - \phi(\lambda) \leq \epsilon\phi(\lambda)$.

For a given threshold $\tau > 0$, consider the thresholded vector $\tilde{w}(\tau)$ defined as

$$\tilde{w}_k(\tau) \quad = \quad \left\{ \begin{array}{ll} 0 & \text{if } |w_k^*| \leq \tau, \\ w_k^* & \text{otherwise,} \end{array} \right. \qquad k = 1, \ldots, n.$$

We have $\tilde{w}(\tau) = w^* + \delta(\tau)$ where the vector of perturbation $\delta(\tau)$ is such that

$$\delta_k(\tau) \quad = \quad \left\{ \begin{array}{ll} -w_k^* & \text{if } |w_k^*| \leq \tau, \\ 0 & \text{otherwise,} \end{array} \right. \qquad k = 1, \ldots, n.$$

Note that, by construction, we have $\|w^*\|_1 = \|w^* + \delta\|_1 + \|\delta\|_1$. Also note that if w^* is sparse, so is δ.

Let us now denote by ϕ_τ the LASSO objective that we obtain upon replacing the optimum classifier w^* with its thresholded version $\tilde{w}(\tau) = w^* + \delta(\tau)$:

$$\phi_\tau \quad := \quad \frac{1}{2}\|X^T(w^* + \delta(\tau)) - y\|_2^2 + \lambda\|w^* + \delta(\tau)\|_1.$$

Since $w(\tau)$ is (trivially) feasible for the primal problem, we have $\phi_\tau \geq \phi(\lambda)$. On the other hand,

$$\begin{array}{rcl} \phi_\tau & = & \frac{1}{2}\|X^T w^* - y\|_2^2 + \lambda\|w^* + \delta(\tau)\|_1 + \frac{1}{2}\|X^T\delta(\tau)\|_2^2 + \delta(\tau)^T X(X^T w^* - y) \\ & \leq & \frac{1}{2}\|X^T w^* - y\|_2^2 + \lambda\|w^*\|_1 + \frac{1}{2}\|X^T\delta(\tau)\|_2^2 + \delta(\tau)^T X(X^T w^* - y). \end{array}$$

For a given $\alpha > 1$, the condition

$$\mathcal{C}(\tau) := \frac{1}{2}\|X^T\delta(\tau)\|_2 + \delta(\tau)^T X(X^T w^* - y) \leq \kappa\phi^*, \;\; \kappa := \frac{1 + \alpha\epsilon}{1 + \epsilon} - 1 \geq 0, \tag{38}$$

allows to write

$$\phi(\lambda) \leq \phi_\tau \leq \;\; (1 + \alpha\epsilon)\phi(\lambda).$$

The condition (38) then implies that the thresholded classifier is sub-optimal, with relative accuracy $\alpha\epsilon$.

Our proposed thresholding rule is based on the condition (38). Precisely, we choose the parameter $\alpha > 0$, then we set the threshold level τ by solving, via line search, the largest threshold τ allowed by condition (38):

$$\tau_\alpha = \arg\max_{\tau \geq 0} \left\{ \tau \;\; : \;\; \|X^T\delta(\tau)\|_2 \leq \left(\sqrt{\frac{1 + \alpha\epsilon}{1 + \epsilon}} - 1 \right) \|X^T w^* - y\|_2 \right\}.$$

The larger α is, the more elements the rule allows to set to zero; at the same time, the more degradation in the objective will be observed: precisely, the new relative accuracy is bounded by $\alpha\epsilon$. The rule also depends on the duality gap parameter ϵ. We refer to the thresholding rule as TR(α) in the sequel. In practice, we observe that the value $\alpha = 2$ works well, in a sense made more precise below.

The complexity of the rule is $O(mn)$. More precisely, the optimal dual variable $\theta^* = X^T w^* - y$ is returned by IPM-LASSO. The matrix $X\theta^* = X(X^T w^* - y)$ is computed once for all in $O(mn)$. We then sort the optimal vector w^* so that $|w^*_{(1)}| \leq \ldots \leq |w^*_{(n)}|$, and set $\tau = \tau_0 = |w^*_{(n)}|$, so that $\delta_k(\tau_0) = -w^*_k$ and $\tilde{w}_k(\tau_0) = 0$ for all $k = 1, \ldots, n$. The product $X^T\delta(\tau_0)$ is computed in $O(mn)$, while the product $\delta(\tau_0)^T(X\theta^*)$ is computed in $O(n)$. If the quantity $\mathcal{C}(\tau_0) = \frac{1}{2}\|X^T\delta(\tau_0)\|_2 + \delta(\tau_0)^T(X\theta^*)$ is greater than $\kappa\phi^\star$, then we set $\tau = \tau_1 = |w^*_{(n-1)}|$. We have $\delta_k(\tau_1) = \delta_k(\tau_0)$ for any $k \neq (n)$ and $\delta_{(n)}(\tau_1) = 0$. Therefore, $\mathcal{C}(\tau_1)$ can be deduced from $\mathcal{C}(\tau_0)$ in $O(n)$. We proceed by successively setting $\tau_k = |w^*_{(n-k)}|$ until we reach a threshold τ_k such that $\mathcal{C}(\tau_k) \leq \kappa\phi^*$.

Simulation study. We conducted a simple simulation study to evaluate our proposal and compare it to the KKT thresholding rule. Both methods were further compared to the results returned by the `glmnet R` package. The latter algorithm returns hard zeros in the classifier coefficients, and we have chosen the corresponding sparsity pattern as the "ground truth", which the IPM-LASSO should recover.

We first experimented with synthetic data. We generated samples of the pair (X, y) for various values of (m, n). We present the results for $(m, n) = (5000, 2500)$ and $(m, n) = (100, 500)$. The number s of relevant features was set to $\min(m, n/2)$. Features were drawn from independent $\mathcal{N}(0, 1)$ distributions and y was computed as $y = X^T w + \xi$, where $\xi \sim \mathcal{N}(0, 0.2)$ and w is a vector of $\mathbb{R}^n$ with first s components equal to $0.1 + 1/s$ and remaining $n - s$ components set to 0. Because `glmnet` includes an unpenalized intercept while Boyd's method does not, both y and X were centered before applying either methods to make their results comparable.

Results are presented on Figures 7. First, the KKT thresholding rule was observed to be very chaotic when the duality gap was set to $\epsilon = 10^{-4}$ (we recall here that the default value for the duality gap in Boyd's `MATLAB` implementation is $\epsilon = 10^{-3}$), while it was way better when duality gap was set to $\epsilon = 10^{-8}$ (somehow justifying our choice of considering the sparsity pattern returned by `glmnet` as the ground truth). Therefore, for applications where computational time is not critical, running Boyd's IPM method and applying Boyd's thresholding rule should yield appropriate results. However, when computational time matters, passing the duality gap from, say, 10^{-4} to 10^{-8}, is not a viable option. Next, regarding our proposal, we observed that it was significantly better than Boyd's thresholding rule when the duality gap was set to 10^{-4} and equivalent to Boyd's thresholding rule for a duality gap of 10^{-8}. Interestingly, setting $\alpha = 1.5$ in (38) generally enabled to achieved very good results for low values of λ, but lead to irregular results for higher values of λ (in the case $m = 100$, results were unstable for the whole range of λ values we considered). Overall, the choices $\alpha = 2$, 3 and 4 lead to acceptable results. A little irregularity remained with $\alpha = 2$ for high values of λ, but this choice of α performed the best for lower values of λ. As for choices $\alpha = 3$ and $\alpha = 4$, it is noteworthy that the results were all the better as the dimension n was low.

D.1 Real data examples

We also applied our proposal and compared it to KKT rule (37) on real data sets arising in text classification. More precisely, we used the New York Times headlines data set presented in the Numerical results Section. For illustration, we present here results we obtained for the topic "China" and the year 1985. We successively ran IPM-LASSO method with duality gap set to 10^{-4} and 10^{-8} and compare the number of active features returned after applying KKT thresholding rule (37) and TR (1.5), TR (2), TR (3) and TR (4). Results are presented on Figure 8. Because we could not applied `glmnet` on this data set, the ground truth was considered as the result of KKT rule, when

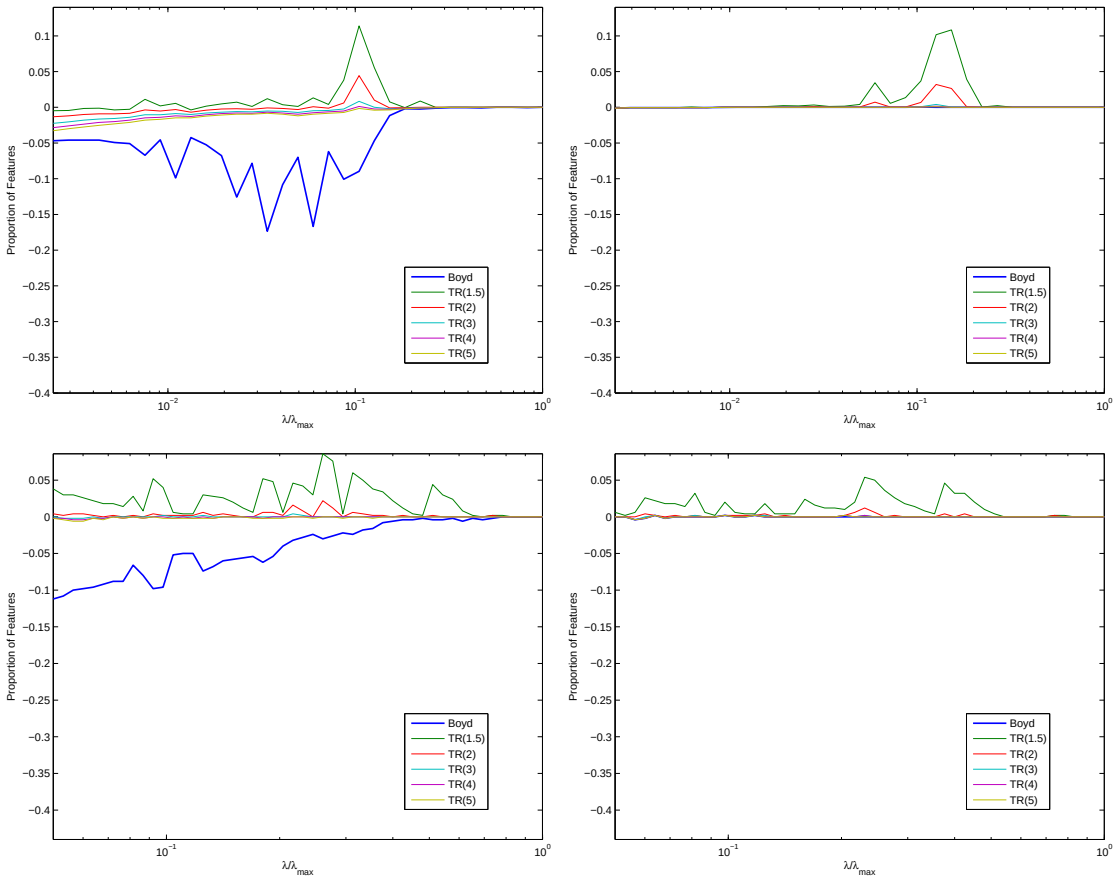

Figure 7: Comparison of several thresholding rules on synthetic data: the case $m = 5000$, $n = 100$ (*top panel*) and $m = 100$, $n = 500$ (*bottom panel*) with duality gap in Boyd's IPM method set to (i) 10^{-4} (*left panel*) and (iii) 10^{-8} (*right panel*). The curves represent the differences between the number of active features returned after each thresholding method and the one returned by `glmnet` (this difference is further divided by the total number of features n). The graphs present the results attached to six thresholding rules: the one proposed by Koh et al. (2007) and five versions of our proposal, corresponding to setting α in (38) to 1.5, 2, 3, 4 and 5 respectively. Overall, these results suggest that by setting $\alpha \in (2, 5)$, our rule is less sensitive to the value of the duality gap parameter in IPM-LASSO than is the rule proposed by Koh et al. (2007).

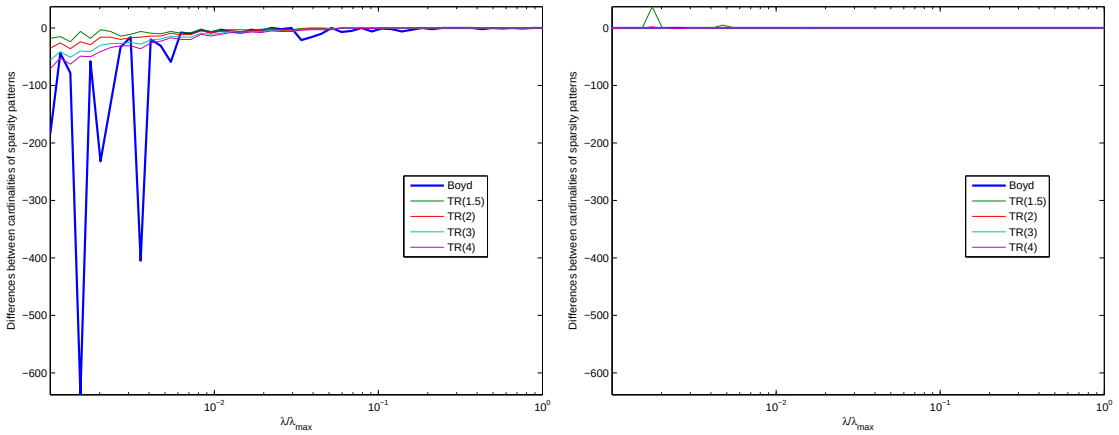

Figure 8: Comparison of several thresholding rules on the NYT headlines data set for the topic "China" and year 1985. Duality gap in IPM-LASSO was successively set to 10^{-4} (*left panel*) and 10^{-8} (*right panel*). The curves represent the differences between the number of active features returned after each thresholding method and the one returned by the KKT rule when duality gap was set to 10^{-10}. The graphs present the results attached to five thresholding rules: the KKT rule and four versions of our rule, corresponding to setting α in (38) to 1.5, 2, 3 and 4 respectively. Results obtained following our proposal appear to be less sensitive to the value of the duality gap used in IPM-LASSO. For instance, for the value $\lambda = \lambda_{\max}/1000$, the KKT rule returns 1758 active feature when the duality gap is set to 10^{-4} while it returns 2357 features for a duality gap of 10^{-8}.

applied to the model returned by IPM-LASSO ran with duality gap set to 10^{-10}. Applying KKT rule on the model built with a duality gap of 10^{-4} lead to very misleading results again, especially for low values of λ. In this very high-dimensional setting ($n = 38377$ here), our rule generally resulted in a slight "underestimation" of the true number of active features for the lowest values of λ when the duality gap was set to 10^{-4}. This suggests that the "optimal" α for our rule might depend on both n and λ when the duality gap is not small enough. However, we still observed that our proposal significantly improved upon KKT rule when the duality gap was set to 10^{-4}.

References

Jinbo Bi, Kristin P. Bennet, Mark Embrechts, Curt M. Breneman, and Minghu Song. Dimensionality reduction via sparse support vector machines. *J. Mach. Learn. Res.*, 3:1229–1243, 2003.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787.

Emmanuel Candès and Justin Romberg. l_1-magic: A collection of matlab routines for solving the convex optimization programs central to compressive sampling. Available: http://www.acm.caltech.edu/l1magic/, 2006.

David L. Donoho and Yaakov Tsaig. Fast solution of l_1-norm minimization problems when the solution may be sparse. *IEEE Trans. Inform. Theory*, 54(11):4789–4812, 2008.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression (with discussion). *Ann. Statist.*, 32:407–499, 2004.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *J. Roy. Statist. Soc. Ser. B*, 70(5):849–911, 2008.

Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statist. Sinica*, 20:101–148, 2010.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

George Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3:1289–1305, 2003.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Statist.*, 1(2):302–332, 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. URL `http://www.jstatsoft.org/v33/i01/`.

Glenn M. Fung and O. L. Mangasarian. A feature selection Newton method for support vector machine classification. *Comput. Optim. Appl.*, 28(2):185–202, 2004.

Brian Gawalt, Jinzhu Jia, Luke Miratrix, Laurent El Ghaoui, Bin Yu, and Sophie Clavier. Discovering word associations in news media via feature selection and sparse classification. In *MIR '10: Proceedings of the international conference on Multimedia information retrieval*, pages 211–220, 2010.

Alexander Genkin, David D. Lewis, and David Madigan. Large-scale Bayesian logistic regression for text categorization. *Technometrics*, 49(3):291–304, 2007.

Joshua Goodman. Exponential priors for maximum entropy models. *Proc. of the Annual Meetings of the Association for Computational Linguistics*, 2004.

Seung-Jean Kim, Kwangmoo Koh, Michael Lustig, Stephen Boyd, and Dimitry Gorinevsky. An interior-point method for large-scale l_1-regularized least squares. *IEEE J. Select. Top. Sign. Process.*, 1(4):606–617, 2007.

Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale l_1-regularized logistic regression. *JMLR*, 8:1519–1555, 2007.

Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient l_1-regularized logistic regression. *Proc. of the 21st National Conference on Artificial Intelligence (AAAI-06)*, 2006.

Mee Young Park and Trevor Hastie. L_1-regularization path algorithm for generalized linear models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 69(4):659–677, 2007.

Volker Roth. The generalized LASSO. *IEEE Transactions on Neural Networks*, 15(1):16–28, 2004.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *NIPS*, 2003.