HAL
archives-ouvertes.fr

Screening Data Points in Empirical Risk Minimization via Ellipsoidal Regions and Safe Loss Functions

Grégoire Mialon, Alexandre d'Aspremont, Julien Mairal

▶ To cite this version:

Grégoire Mialon, Alexandre d'Aspremont, Julien Mairal. Screening Data Points in Empirical Risk Minimization via Ellipsoidal Regions and Safe Loss Functions. 2019. hal-02395624

HAL Id: hal-02395624

https://hal.archives-ouvertes.fr/hal-02395624

Submitted on 5 Dec 2019

Screening Data Points in Empirical Risk Minimization via Ellipsoidal Regions and Safe Loss Functions

Grégoire Mialon
Inria[*][†]
gregoire.mialon@inria.fr

Alexandre d'Aspremont
CNRS - ENS[†]
aspremon@ens.fr

Julien Mairal
Inria[*]
julien.mairal@inria.fr

December 5, 2019

Abstract

We design simple screening tests to automatically discard data samples in empirical risk minimization without losing optimization guarantees. We derive loss functions that produce dual objectives with a sparse solution. We also show how to regularize convex losses to ensure such a dual sparsity-inducing property, and propose a general method to design screening tests for classification or regression based on ellipsoidal approximations of the optimal set. In addition to producing computational gains, our approach also allows us to compress a dataset into a subset of representative points.

1   Introduction

Let us consider a collection of n pairs $(a_i, b_i)_{i=1,\dots,n}$, where each vector a_i in $\mathbb{R}^p$ describes a data point and b_i is its label. For regression, b_i is real-valued, and we address the convex optimization problem

$$\min_{x \in \mathbb{R}^p, t \in \mathbb{R}^n} f(t) + \lambda R(x) \quad \text{s.t.} \quad t = Ax - b, \tag{$\mathcal{P}_1$}$$

where $A = [a_1, \dots, a_n]^\top$ in $\mathbb{R}^{n \times p}$ carries the feature vectors, and $b = [b_1, \dots, b_n]$ carries the labels. The function f is a convex loss and measures the fit between data points and the model, and R is a convex regularization function. For classification, the scalars b_i are binary labels in $\{-1, +1\}$, and we consider instead of (($\mathcal{P}_1$)) margin-based loss functions, where our problem becomes

$$\min_{x \in \mathbb{R}^p, t \in \mathbb{R}^n} f(t) + \lambda R(x) \quad \text{s.t.} \quad t = \mathbf{diag}(b) Ax, \tag{$\mathcal{P}_2$}$$

The above problems cover a wide variety of formulations such as Lasso [21] and its variants [26], logistic regression, support vector machines [10], and many more.

When R is the ℓ_1-norm, the solution is encouraged to be sparse [1], which can be exploited to speed-up optimization procedures.

A recent line of work has focused on screening tests that seek to automatically discard variables before running an optimization algorithm. For example, [7] derive a screening rule from Karush-Kuhn-Tucker conditions, noting that if a dual optimal variable satisfies a given inequality constraint, the corresponding primal optimal variable must be zero. Checking this condition on a set that is known to contain the optimal dual variable ensures that the corresponding primal variable can be safely removed. This prunes out irrelevant features *before* solving the problem. This is called a *safe* rule if it discards variables that are guaranteed to be useless; but it is possible to relax the "safety" of the rules [22] without losing too much accuracy in practice.

---

[*]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France.
[†]D.I., UMR 8548, École Normale Supérieure, Paris, France.

The seminal approach by [7] has led to a series of works proposing refined tests [6, 24] or dynamic rules [9] for the Lasso, where screening is performed as the optimization algorithm proceeds, significantly speeding up convergence. Other papers have proposed screening rules for sparse logistic regression [23] or other linear models.

Whereas the goal of these previous methods is to remove *variables*, our goal is to design screening tests for *data points* in order to remove observations that do not contribute to the final model. The problem is important when there is a large amount of "trivial" observations that are useless for learning. This typically occurs in *tracking or anomaly detection* applications, where a classical heuristic seeks to mine the data to find difficult examples [8].

A few of such screening tests for data points have been proposed in the literature. Some are problem-specific (*e.g.* [17] for SVM), others are making strong assumptions on the objective. For instance, the most general rule of [19] for classification requires strong convexity and the ability to compute a duality gap in closed form.

The goal of our paper is to provide a more generic approach for screening data samples, both for regression and classification. Such screening tests may be designed for loss functions that induce a sparse dual solution. We describe this class of loss functions and investigate a regularization mechanism that ensures that the loss enjoys such a property.

Our contributions can be summarized as follows:

- We revisit the Ellipsoid method [3] to design screening test for samples, when the objective is convex and its dual admits a sparse solution.

- We propose a new regularization mechanism to design regression or classification losses that induce sparsity in the dual. This allows us to recover existing loss functions and to discover new ones with sparsity-inducing properties in the dual.

- Originally designed for linear models, we extend our screening rules to kernel methods. Unlike the existing literature, our method also works for non strongly convex objectives.

- We demonstrate the benefits of our screening rules in various numerical experiments on large-scale classification problems and regression.

2 Preliminaries

We now present the key concepts used in our paper.

2.1 Fenchel Conjugacy

Definition 2.1 (Fenchel conjugate). *Let $f : \mathbb{R}^p \to \mathbb{R} \cup \{-\infty, +\infty\}$ be an extended real-valued function. The Fenchel conjugate of f is defined by*

$$f^*(y) = \max_{t \in \mathbb{R}^p} \langle t, y \rangle - f(t).$$

The biconjugate of f is naturally the conjugate of f^* and is denoted by f^{**}. The Fenchel-Moreau theorem [11] states that if f is proper, lower semi-continuous and convex, then it is equal to its biconjugate f^{**}. Finally, Fenchel-Young's inequality gives for all pair (t, y)

$$f(t) + f^*(y) \geq \langle t, y \rangle,$$

with an equality case iff $y \in \partial f(t)$.

Suppose now that for such a function f, we add a convex term Ω to f^* in the definition of the biconjugate. We get a modified biconjugate f_μ, written

$$f_\mu(t) = \max_{y \in \mathbb{R}^p} \langle y, t \rangle - f^*(y) - \mu\Omega(y)$$

$$= \max_{y \in \mathbb{R}^p} \langle y, t \rangle + \min_{z \in \mathbb{R}^p} \{-\langle z, y \rangle + f(z)\} - \mu\Omega(y).$$

The inner objective function is continuous, concave in y and convex in z, such that we can switch min and max according to Von Neumann's minimax theorem to get

$$f_\mu(t) = \min_{z \in \mathbb{R}^p} f(z) + \max_{y \in \mathbb{R}^p} \{\langle t - z, y \rangle - \mu\Omega(y)\}$$
$$= \min_{z \in \mathbb{R}^p} f(z) + \mu\Omega^*\left(\frac{t - z}{\mu}\right).$$

Definition 2.2 (Infimum convolution). *f_μ is called the infimum convolution of f and Ω^*, which may be written as $f \square \Omega^*$.*

Note that f_μ is convex as the minimum of a convex function in (t, z). We recover the Moreau-Yosida smoothing [14, 25] and its generalization when Ω is respectively a quadratic term or a strongly-convex term [16].

2.2 Empirical Risk Minimization and Duality

Let us consider the convex ERM problem

$$\min_{x \in \mathbb{R}^p} P(x) = \frac{1}{n} \sum_{i=1}^n f_i(a_i^\top x) + \lambda R(x), \tag{1}$$

which covers both $(\mathcal{P}_1)$ and $(\mathcal{P}_2)$ by using the appropriate definition of function f_i. We consider the dual problem (obtained from Lagrange duality)

$$\max_{\nu \in \mathbb{R}^n} D(\nu) = \frac{1}{n} \sum_{i=1}^n -f_i^*(\nu_i) - \lambda R^*\left(-\frac{A^T \nu}{\lambda n}\right). \tag{2}$$

We always have $P(x) \geq D(\nu)$. Since there exists a pair (x, t) such that $Ax = t$ (Slater's conditions), we have $P(x^*) = D(\nu^*)$ and $x^\star = -\frac{A^\top \nu^\star}{\lambda n}$ at the optimum.

2.3 Safe Loss Functions and Sparsity in the Dual of ERM Formulations

A key feature of our losses is to encourage sparsity of dual solutions, which typically emerge from loss functions with a flat region. We call such functions "safe losses" since they will allow us to design safe screening tests.

Definition 2.3 (Safe loss function). *Let $\phi : \mathbb{R} \to \mathbb{R}$ be a continuous convex loss function such that $\inf_{t \in \mathbb{R}} \phi(t) = 0$. We say that ϕ is a safe loss if there exists a non-singleton and non-empty interval $\mathcal{I} \subset \mathbb{R}$ such that*

$$t \in \mathcal{I} \implies \phi(t) = 0.$$

Lemma 2.4 (Safe loss and dual sparsity). *Consider the problem (1) where R is a convex penalty. Denoting by $x^\star$ and $\nu^\star$ the optimal primal and dual variables respectively, we have for all $i = 1, \ldots, n$,*

$$\nu_i^* \in \partial f_i(a_i^\top x^\star).$$

A consequence of this lemma is that for both classification and regression, the sparsity of the dual solution is related to loss functions that have "flat" regions—that is, such that $0 \in \partial f_i'(t)$. This is the case for safe loss functions defined above. Note that the relation between flat losses and sparse dual solutions is classical, see [20, 4].

3 Safe rules for screening data points

In this section, we derive screening rules in the spirit of SAFE [7] to select data points in regression or classification problems with safe losses.

3.1 Principle of SAFE Rules for Data Points

We recall that our goal is to safely delete data points prior to optimization, that is, we want to train the model on a subset of the original dataset while still getting the same optimal solution as a model trained on the whole dataset. This amounts to identifying beforehand which dual variables are zero at the optimum. Indeed, as discussed in Section 2.2, the optimal primal variable $x^\star = -\frac{A^\top \nu^\star}{\lambda n}$ only relies on non-zero entries of $\nu^\star$. To that effect, we make the following assumption:

Assumption 3.1 (Safe loss assumption)**.** *We consider problem (1), where each f_i is a safe loss function. Specifically, we assume that $f_i(a_i^\top x) = \phi(a_i^\top x - b_i)$ for regression, or $f_i(a_i^\top x) = \phi(b_i a_i^\top x)$ for classification, where ϕ satisfies Definition 2.3 on some interval $\mathcal{I}$. For simplicity, we assume that there exists $\mu > 0$ such that $\mathcal{I} = [-\mu, \mu]$ for regression losses and $\mathcal{I} = [\mu, +\infty)$ for classification, which covers most useful cases.*

We may now state the basic safe rule for screening.

Lemma 3.2 (SAFE rule)**.** *Under Assumption 3.1, consider a subset $\mathcal{X}$ containing the optimal solution $x^\star$. If, for a given data point (a_i, b_i), $a_i^\top x - b_i \in \mathring{\mathcal{I}}$ for all x in $\mathcal{X}$, (resp. $b_i a_i^\top x \in \mathring{\mathcal{I}}$), where $\mathring{\mathcal{I}}$ is the interior of I, then this data point can be discarded from the dataset.*

Proof. From the definition of safe loss functions, f_i is differentiable at $a_i^\top x^\star$ with $\nu_i^\star = f_i'(a_i^\top x) = 0$. ∎

We see now how the safe screening rule can be interpreted in terms of discrepancy between the model prediction $a_i^\top x$ and the true label b_i. If, for a set $\mathcal{X}$ containing the optimal solution x^* and a given data point (a_i, b_i), the prediction always lies in $\mathring{\mathcal{I}}$, then the data point can be discarded from the dataset. The data point screening procedure therefore consists in maximizing linear forms, $a_i^\top x - b_i$ and $-a_i^\top x + b_i$ in regression (resp. minimizing $b_i a_i^\top x$ in classification), over a set $\mathcal{X}$ containing x^* and check whether they are lower (resp. greater) than the threshold μ. The smaller $\mathcal{X}$, the lower the maximum (resp. the higher the minimum) hence the more data points we can hope to safely delete. Finding a good test region $\mathcal{X}$ is critical however. We show how to do this in the next section.

3.2 Building the Test Region $\mathcal{X}$

Screening rules aim at sparing computing resources, testing a data point should therefore be easy. As in [7] for screening variables, if $\mathcal{X}$ is an ellipsoid, the optimization problem detailed above admits a closed-form solution. Furthermore, it is possible to get a smaller set $\mathcal{X}$ by adding a first order optimality condition with a subgradient g of the objective evaluated in the center z of this ellipsoid. This linear constraint cuts the final ellipsoid roughly in half thus reducing its volume.

Lemma 3.3 (Closed-form screening test)**.** *Consider the optimization problem*

$$
\begin{aligned}
& maximize && a_i^\top x - b_i \\
& subject\ to && (x - z)^T E^{-1}(x - z) \leq 1 \\
& && g^T(x - z) \leq 0
\end{aligned}
\tag{3}
$$

in the variable x in $\mathbb{R}^p$ with E defining an ellipsoid with center z and g is in $\mathbb{R}^p$. Then the maximum is

$$
\begin{cases}
a_i^\top z + (a_i^\top E a_i)^{\frac{1}{2}} - b_i \ if\ g^T E a_i < 0 \\
a_i^\top \left(z + \frac{1}{2\gamma} E(a_i - \nu g) \right) - b_i \ otherwise,
\end{cases}
$$

with $\nu = \frac{g^T E a_i}{g^T E g}$ and $\gamma = \left(\frac{1}{2}(a_i - \nu g)^\top E(a_i - \nu g) \right)^{\frac{1}{2}}$.

The proof can be found in Appendix A.2 and it is easy to modify it for minimizing $b_i a_i^\top x$. We can obtain both E and z by using a few steps of *the ellipsoid method* [15, 3]. The method starts from an initial ellipsoid containing the solution x^* to a given convex problem. It iteratively computes a subgradient in the center of
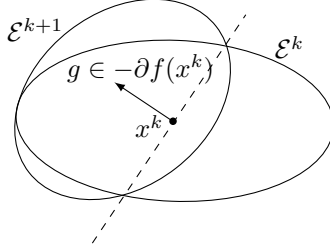
4

Figure 1: One step of the ellipsoid method.

the current ellipsoid, selects the half-ellipsoid containing x^*, and computes the ellipsoid with minimal volume containing the previous half-ellipsoid before starting all over again. Such a method, presented in Algorithm 1, performs closed-form updates of the ellipsoid.

Note that the ellipsoid update formula was also used to screen primal variables for the Lasso problem [6], although not iterating over ellipsoids in order to get smaller volumes.

Algorithm 1 Building ellipsoidal test regions

1: **initialization:** Given $\mathcal{E}^0(x_0, E_0)$ containing x^*;
2: **while** $k < nb_{\text{steps}}$ **do**
3:      • Compute a subgradient g of (1) in x_k;
4:      • $\tilde{g} \leftarrow g/\sqrt{g^T E_k g}$;
5:      • $x_{k+1} \leftarrow x_k - \frac{1}{p+1} E_k \tilde{g}$;
6:      • $E_{k+1} \leftarrow \frac{p^2}{p^2-1}(E_k - \frac{2}{p+1} E_k \tilde{g}\tilde{g}^T E_k)$;
7: For regression problems:
8: **for** each sample a_i in A **do**
9:      **if** $\max|a_i^\top x - b_i| \leq \mu$ for $x \in \mathcal{E}^{nb_{\text{steps}}}$ **then**
10:          Discard a_i from A.
11: For classification, replace condition $|a_i^\top x - b_i| \leq \mu$ by $b_i a_i^\top x \geq \mu$ in the above expression.

Initialization. The algorithm requires an initial ellipsoid $\mathcal{E}^0(x_0, E_0)$ that contains the solution. This is typically achieved by defining the center x_0 as an approximate solution of the problem, which can be obtained in various ways. For instance, one may run a few steps of a solver on the whole dataset, or one may consider the solution obtained previously for a different regularization parameter when computing a regularization path, or the solution obtained for slightly different data, *e.g.*, for tracking applications where an optimization problem has to be solved at every time step t, with slight modifications from time $t - 1$.

Once the center x_0 is defined, there are many cases where the initial ellipsoid can be safely assumed to be a sphere. For instance, if the objective—let us call it F—is κ-strongly convex, we have the basic inequality $\frac{\kappa}{2}\|x_0 - x^\star\|^2 \leq F(x_0) - F^\star$, which can often be upper-bounded by several quantities, *e.g.*, a duality gap [19] or simply $F(x_0)$ if F is non-negative as in typical ERM problems. Otherwise, other strategies can be used depending on the problem at hand, as done for the Lasso by [7, 9] for example.

Efficient implementation. Since each update of the ellipsoid matrix E is rank one, it is possible to parametrize E_k at step k as

$$E_k = s_k \mathrm{I} - L_k D_k L_k^T,$$

with I the identity matrix, L_k is in $\mathbb{R}^{p \times k}$ and D_k in $\mathbb{R}^{k \times k}$ is a diagonal matrix. Hence, we only have to update D and L while the algorithm proceeds.

Complexity of our screening rules. For each step of Algorithm 1, we compute a subgradient g in $O(np)$ operations. The ellipsoids are modified using rank one updates that can be stored. As a consequence, the computations at this stage are dominated by the computation of Eg, which is $O(pk)$. As a result, k steps cost $O(k^2p + npk)$.

Once we have the test set $\mathcal{X}$, we have to compute the closed forms from Lemma 3.3 for each data point. This computation are dominated by the matrix-vector multiplications with E, which cost $O(kp)$ using the structure of E. Hence, testing the whole dataset costs $O(npk)$. Since we typically have $n \gg k$, the cost of the overall screening procedure is therefore $O(npk)$.

In constrast, solving the ERM problem without screening would cost $O(npT)$ where T is the number of passes over the data, with $T \gg k$. With screening, the complexity becomes $O(nsT + npk)$, where s is the number of data points accepted by the screening procedure.

3.3 Extension to Kernel Methods

It is relatively easy to adapt our safe rules to kernel methods. Consider for example $(\mathcal{P}_1)$, where A has been replaced by $\phi(A) = [\phi(a_1), \ldots, \phi(a_n)]^\top$ in $\mathcal{H}^n$, with $\mathcal{H}$ a RKHS and ϕ its mapping function $\mathbb{R}^p \to \mathcal{H}$. The prediction function $g \colon \mathbb{R}^p \to \mathbb{R}$ lives in the RKHS, thus it can be written $g(a) = \langle g, \phi(a) \rangle$, $\forall a \in \mathbb{R}^p$. In the setting of ERM, the representer theorem ensures $g(a) = \sum_{i=1}^n \alpha_i K(a_i, a)$ with $\alpha_i \in \mathbb{R}$ and K the kernel associated to $\mathcal{H}$. The problem becomes:

$$\min_{\alpha \in \mathbb{R}^n, t \in \mathbb{R}^n} f(t) + \lambda \sum_{i,j=1}^n \alpha_i \alpha_j K(a_i, a_j) \quad \text{s.t.} \quad t = \mathbf{K}\alpha - b, \tag{4}$$

with $\mathbf{K}$ the Gram matrix. The constraint is linear in α (thus satisfying to Lemma 4.1) while yielding non-linear prediction functions. The screening test becomes maximizing the linear forms $[\mathbf{K}]_i\alpha - b_i$ and $-[\mathbf{K}]_i\alpha + b_i$ over an ellipsoid $\mathcal{X}$ containing α^*. When the problem is convex (it depends on K), $\mathcal{X}$ can still be found using the ellipsoid method.

We now have an algorithm for selecting data points in regression or classification problems with linear or kernel models. As detailed above, the rules require a sparse dual, which is not the case in general except in particular instances such as support vector machines. We now explain how to induce sparsity in the dual.

4 Constructing safe losses

In this section, we introduce a way to induce sparsity in the dual of empirical risk minimization problems.

4.1 Inducing Sparsity in the Dual of ERM

When the ERM problem does not admit a sparse dual solution, safe screening is not possible. To fix this issue, consider the ERM problem $(\mathcal{P}_1)$ and replace f by f_μ defined in Section 2:

$$\min_{x \in \mathbb{R}^p, t \in \mathbb{R}^n} f_\mu(t) + \lambda R(x) \quad \text{s.t.} \quad t = Ax - b, \tag{$\mathcal{P}_1'$}$$

We have the following result connecting the dual of $(\mathcal{P}_1)$ with that of $(\mathcal{P}_1')$.

Lemma 4.1 (Regularized dual for regression)**.** *The dual of $(\mathcal{P}_1')$ is*

$$\max_{\nu \in \mathbb{R}^n} -\langle b, \nu \rangle - f^*(\nu) - \lambda R^* \left(-\frac{A^T \nu}{\lambda} \right) - \mu \Omega(\nu), \tag{5}$$

and the dual of $(\mathcal{P}_1)$ is obtained by setting $\mu = 0$.

Before we prove this lemma, we remark that is possible, in many cases, to induce sparsity in the dual if Ω is the ℓ_1-norm, or another sparsity-inducing penalty. This is notably true if the unregularized dual is smooth with bounded gradients. In such a case, it is possible to show that the optimal dual solution would be $\nu^\star = 0$ as soon as μ is large enough [1].

Proof. We can write $\mathcal{P}'_1$ as

$$
\begin{aligned}
\text{minimize} \quad & \tilde{f}(\tilde{x}) + \lambda \tilde{R}(\tilde{x}) \\
\text{subject to} \quad & \tilde{A}\tilde{x} = -b
\end{aligned}
\tag{6}
$$

in the variable $\tilde{x} = (t, x) \in \mathbb{R}^{n+p}$ with $\tilde{f} \colon \tilde{x} \mapsto f_\mu(t)$ and $\tilde{R} \colon \tilde{x} \mapsto R(x)$ and $\tilde{A} \in \mathbb{R}^{n \times (n+p)} = (\mathrm{Id}, -A)$. Since the constraints are linear, we can directly express the dual of this problem in terms of the Fenchel conjugate of the objective (see *e.g.* [5], 5.1.6). Let us note $f_0 = \tilde{f} + \lambda \tilde{R}$. For all $y \in \mathbb{R}^{n+p}$, we have

$$
\begin{aligned}
f_0^*(y) &= \sup_{x \in \mathbb{R}^{n+p}} \langle x, y \rangle - \tilde{f}(x) - \lambda \tilde{R}(x) \\
&= \sup_{x_1 \in \mathbb{R}^n, x_2 \in \mathbb{R}^p} \langle x_1, y_1 \rangle + \langle x_2, y_2 \rangle - f(x_1) - \lambda R(x_2) \\
&= f_\mu^*(y_1) + \lambda R^* \left(\frac{y_2}{\lambda} \right).
\end{aligned}
$$

It is known from [2] that $f_\mu = f \,\square\, \Omega_\mu^* = (f^* + \Omega_\mu^{**})^*$ with $\Omega_\mu^* = \mu \Omega^*(\div)$. Clearly, $\Omega_\mu^{**} = \mu \Omega$. If Ω is proper, convex and lower semicontinuous, then $\Omega = \Omega^{**}$. As a consequence, $f_\mu^* = (f^* + \mu\Omega)^{**}$. If $f^* + \mu\Omega$ is proper, convex and lower semicontinuous, then $f_\mu^* = f^* + \mu\Omega$, hence

$$
f_0^*(y) = f^*(y_1) + \lambda R^* \left(\frac{y_2}{\lambda} \right) + \mu\Omega(y_1).
$$

Now we can form the dual of $\mathcal{P}'_1$ by writing

$$
\text{maximize} \quad -\langle -b, \nu \rangle - f_0^*(-\tilde{A}^T \nu)
\tag{7}
$$

in the variable $\nu \in \mathbb{R}^n$. Since $-\tilde{A}^T \nu = (-\nu, A^T \nu)$ with $\nu \in \mathbb{R}^n$ the dual variable associated to the equality constraints,

$$
f_0^*(-\tilde{A}^T \nu) = f^*(-\nu) + \lambda R^* \left(\frac{A^T \nu}{\lambda} \right) + \mu\Omega(-\nu).
$$

Injecting f_0^* in the problem and setting ν instead of $-\nu$ (we optimize in $\mathbb{R}$) concludes the proof. $\blacksquare$

We consider now the classification problem $(\mathcal{P}_2)$ and show that the previous remarks about sparsity-inducing regularization for the dual of regression problems also hold in this new context.

Lemma 4.2 (Regularized dual for classification). *Consider now the modified classification problem*

$$
\min_{x \in \mathbb{R}^p, t \in \mathbb{R}^n} f_\mu(t) + \lambda R(x) \quad s.t. \quad t = \mathbf{diag}(b) A x.
\tag{$\mathcal{P}'_2$}
$$

The dual of $\mathcal{P}'_2$ is

$$
\max_{\nu \in \mathbb{R}^n} -f^*(-\nu) - \lambda R^* \left(\frac{A^T \mathbf{diag}(b) \nu}{\lambda} \right) - \mu\Omega(-\nu).
\tag{8}
$$

Proof. We proceed as above with a linear constraint $\tilde{A}\tilde{x} = 0$ and $\tilde{A} = (Id, -\mathbf{diag}(b)A)$. $\blacksquare$

Note that the formula directly provides the dual of regression and classification ERM problems with a linear model such as the Lasso and SVM.

4.2 Link Between the Original and Regularized Problems

Lemma 4.3 (Smoothness of f_μ). *If $f^* + \Omega$ is strongly convex, then f_μ is smooth.*

Proof. The lemma follows directly from the fact that $f_\mu = (f^* + \mu\Omega)^*$ (see the proof of Lemma 4.1). The conjugate of a closed, proper, strongly convex function is indeed smooth. ∎

Lemma 4.4 (Bounding f_μ). *If $\mu \geq 0$ and Ω is a norm then*

$$f(t) - \delta(t) \leq f_\mu(t) \leq f(t), \quad \text{for all } t \in \text{dom} f$$

with $\delta(t) = \max_{\|\frac{u}{\mu}\|^* \leq 1} g^T u$ and $g \in \partial f(t)$.

Proof. If Ω is a norm, then $\Omega(0) = 0$ and Ω^* is the indicator function of the dual norm of Ω hence non-negative. Moreover, if $\mu > 0$ then, $\forall z \in \text{dom} f$ and $\forall t \in \mathbb{R}^n$,

$$f_\mu(t) \leq f(z) + \mu\Omega^* \left(\frac{t-z}{\mu} \right).$$

In particular, we can take $t = z$ hence the right-hand inequality. On the other hand,

$$f_\mu(t) - f(t) = \min_z f(z) + \mu I_{\|\frac{z-t}{\mu}\|^* \leq 1} - f(t)$$
$$= \min_{\|\frac{u}{\mu}\|^* \leq 1} f(t+u) - f(t).$$

Since f is convex,

$$f(t+u) - f(t) \geq g^T u \text{ with } g \in \partial f(t).$$

As a consequence,

$$f_\mu(t) - f(t) \geq \min_{\|\frac{u}{\mu}\|^* \leq 1} g^T u.$$

∎

Corollary 4.5 (Bounding the value of $\mathcal{P}_1$). *Let us denote the optimum objectives of $\mathcal{P}_1$, $\mathcal{P}_1'$ by P_λ, $P_{\lambda,\mu}$. If Ω is a norm, we have the following inequalities:*

$$P_\lambda - \delta^* \leq P_{\lambda,\mu} \leq P_\lambda,$$

with δ^* the value of δ at the optimum of $P_\lambda(t) - \delta(t)$.

Proof. The proof is trivial given the inequalities in Lemma 4.4. ∎

4.3 Effect of Regularization and Examples

We start by recalling that the infimum convolution is traditionally used for smoothing an objective when Ω is strongly convex, and then we discuss the use of sparsity-inducing regularization in the dual.

Euclidean distance to a closed convex set. It is known that convolving the indicator function of a closed convex set $\mathcal{C}$ with a quadratic term Ω (the Fenchel conjugate of a quadratic term is itself) yields the euclidean distance to $\mathcal{C}$

$$f_\mu(t) = \min_{z \in \mathbb{R}^n} I_\mathcal{C}(z) + \frac{1}{2\mu} \|t - z\|_2^2 = \min_{z \in \mathcal{C}} \frac{1}{2\mu} \|t - z\|_2^2.$$

Huber loss. The ℓ_1-loss is more robust to outliers than the ℓ_2-loss, but is not differentiable in zero which may induce difficulties during the optimization. A natural solution consists in smoothing it: [2] for example show that applying the Moreau-Yosida smoothing, *i.e* convolving $|t|$ with a quadratic term $\frac{1}{2}t^2$ yields the well-known Huber loss, which is both smooth and robust:

$$f_\mu(t) = \begin{cases} \frac{t^2}{2\mu} & \text{if } |t| \leq \mu, \\ |t| - \frac{\mu}{2} & \text{otherwise.} \end{cases}$$

Now, we present examples where Ω has a sparsity-inducing effect.

Hinge loss. Instead of the quadratic loss in the previous example, choose a robust loss $f \colon t \mapsto \|1 - t\|_1$. By using the same function Ω, we obtain the classical hinge loss of support vector machines

$$f_\mu(t) = \sum_{i=1}^{n} \frac{1}{2}[1 - t_i - \mu, 0]_+.$$

We see that the effect of convolving with the constraint $\mathbf{1}_{x \preceq 0}$ is to turn a regression loss (*e.g.*, square loss) into a classification loss. The effect of the ℓ_1-norm is to encourage the loss to be flat (when μ grows, $[1 - t_i - \mu, 0]_+$ is equal to zero for a larger range of values t_i), which corresponds to the sparsity-inducing effect in the dual that we will exploit for screening data points.

Squared hinge loss. Let us consider a problem with a quadratic loss $f \colon t \mapsto \|1 - t\|_2^2/2$ designed for a classification problem, and consider $\Omega(x) = \|x\|_1 + \mathbf{1}_{x \preceq 0}$. We have $\Omega^*(y) = \mathbf{1}_{y \succeq -1}$, and

$$f_\mu(t) = \sum_{i=1}^{n}[1 - t_i - \mu, 0]_+^2,$$

which is a squared Hinge Loss with a threshold parameter μ and $[.]_+ = \max(0, .)$.

Screening-friendly regression. Consider now the quadratic loss $f \colon t \mapsto \|t\|^2/2$ and $\Omega(x) = \|x\|_1$. Then $\Omega^\star(y) = \mathbf{1}_{\|y\|_\infty \leq 1}$ (see *e.g.* [1]), and

$$f_\mu(t) = \sum_{i=1}^{n} \frac{1}{2}[|t_i| - \mu]_+^2. \tag{9}$$

A proof can be found in Appendix A. As before, the parameter μ encourages the loss to be flat (it is exactly 0 when $\|t\|_\infty \leq \mu$).

Screening-friendly logistic regression. Let us now consider the logistic loss $f(t) = \log\left(1 + e^{-t}\right)$, which we define only with one dimension for simplicity here. It is easy to show that the infimum convolution with the ℓ_1-norm does not induce any sparsity in the dual, because the dual of the logistic loss has unbounded gradients, making classical sparsity-inducing penalties ineffective. However, we may consider instead another penalty to fix this issue: $\Omega(x) = -x \log(-x) + \mu|x|$ for $x \in [-1, 0]$. We have $\Omega^*(y) = -e^{y+\mu-1}$. Convolving Ω^* with f yields

$$f_\mu(x) = \begin{cases} e^{x+\mu-1} - (x + \mu) & \text{if } x + \mu - 1 \leq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Note that this loss is asymptotically robust. Moreover, the entropic part of Ω makes this penalty strongly convex hence f_μ is smooth [16]. Finally, the ℓ_1 penalty ensures that the dual is sparse thus making the screening usable. Our regularization mechanism thus builds a smooth, robust classification loss akin to the logistic loss on which we can use screening rules. The effect of regularization parameter in a few previous cases are illustrated in Figure 2.

In summary, regularizing the dual with the ℓ_1 norm induces a flat region in the loss, which induces sparsity in the dual. The geometry is preserved elsewhere.

9

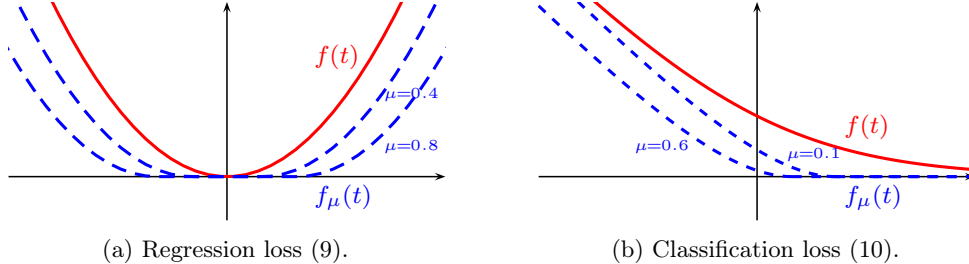(a) Regression loss (9). (b) Classification loss (10).

Figure 2: Effect of the dual sparsity-inducing regularization on the quadratic loss (left) and logistic loss (right). After regularization, the loss functions have flat areas. Note that both of them are smooth.

5 Experiments

We now present experimental results demonstrating the effectiveness of the data screening procedure.

Datasets. We consider three real datasets, SVHN, MNIST, RCV-1, and a synthetic one. *MNIST* ($n = 60000$) and *SVHN* ($n = 604388$) both represent digits, which we encode by using the output of a two-layer convolutional kernel network [13] leading to feature dimensions $p = 2304$. *RCV-1* ($n = 781265$) represents sparse TF-IDF vectors of categorized newswire stories ($p = 47236$). For classification, we consider a binary problem consisting of discriminating digit 9 for MNIST vs. all other digits (resp. digit 1 vs rest for SVHN, 1st category vs rest for RCV-1). For regression, we also consider a synthetic dataset, where data is generated by $b = Ax + \epsilon$, where x is a random, sparse ground truth, $A \in \mathbb{R}^{n \times p}$ a data matrix whose coefficients are in $[-1, 1]$ and $\epsilon \sim \mathcal{N}(0, \sigma)$ with $\sigma = 0.01$. Implementation details are provided in the appendix. We fit usual models using scikit-learn [18].

5.1 Safe Screening

Here, we consider problems that are naturally admit a sparse dual solution, which allows safe screening.

Interval regression. We first illustrate the practical use of the screening-friendly regression loss (9) derived above. It corresponds indeed to a particular case of a supervised learning task called interval regression [12], which is widely used in fields such as economics. In interval regression, one does not have scalar labels but intervals $\mathcal{S}_i$ containing the true labels $\tilde{b}_i$, which are unknown. The loss is written

$$\ell(x) = \sum_{i=1}^{n} \inf_{b_i \in \mathcal{S}_i} (a_i^\top x - b_i)^2, \tag{11}$$

where $\mathcal{S}_i$ contains the true label $\tilde{b}_i$. For a given data point, the model only needs to predict a value inside the interval in order not to be penalized. When the intervals $\mathcal{S}_i$ have the same width and we are given their centers b_i, (11) is exactly (9). Since we proved (9) to yield a sparse dual, we can apply our rules to safely discard intervals that are assured to be matched by the optimal solution. We use an ℓ_1 penalty along with the loss. As an illustration, the experiment was done using a toy synthetic dataset ($n = 20, p = 2$), the signal to recover being generated by one feature only. The intervals can be visualized in Figure 3. The "difficult" intervals (red) were kept in the training set. The predictions hardly fit these intervals. The "easy" intervals (blue) were discarded from the training set: the safe rules certify that the optimal solution will fit these intervals. Our screening algorithm was run for 20 iterations of the Ellipsoid method. Most of the intervals can be ruled out afterwards while the remaining intervals yield the same optimal solution as a model trained on all the intervals.
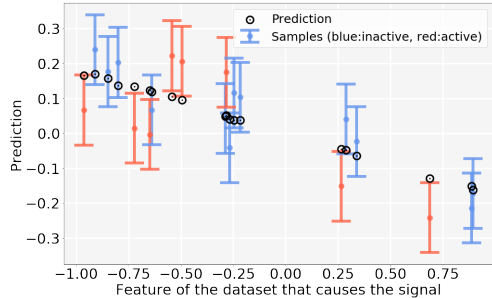
10

Figure 3: Safe interval regression on synthetic dataset. Most "easy" samples (in blue) can be screened (*i.e.* thrown away from the dataset) while the "difficult" ones (in red) are kept.
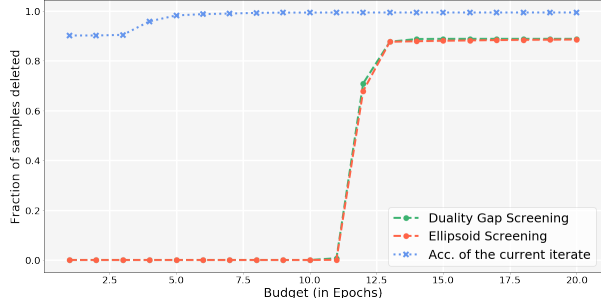

Figure 4: Fraction of samples screened vs Epochs done for two screening strategies along with test accuracy of the current iterate (Squared Hinge $+ \ell_2$ trained on MNIST).

Epochs	20			30		
λ	MNIST	SVHN	RCV-1	MNIST	SVHN	RCV-1
10^{-3}	0	0	1	0	2	12
10^{-4}	0.3	0.01	8	27	17	42
10^{-5}	35	12	45	65	54	75

Table 1: Percentage of samples screened (*i.e* that can be thrown away) in an ℓ_1 penalized Safe Logistic loss given the epochs made at initialization. The radius is initialized respectively at 10 and 1 for MNIST and SVHN at Epochs 20 and 30, and at 1 and 0.1 for RCV-1.

Classification. Common sample screening methods such as [19] require a strongly convex objective. When it is not the case, there is, to the best of our knowledge, no baseline for this case. Thus, when considering classification using the non strongly convex safe logistic loss derived in Section 4 along with an ℓ_1 penalty, our algorithm is still able to screen samples, as shown in Table 1. The algorithm is initialized using an approximate solution to the problem, and the radius of the initial ball is chosen depending on the number of epochs (100 for 10 epochs, 10 for 20 and 1 for 30 epochs), which is valid in practice.

As established in Lemma 2.4, the hinge loss and squared hinge loss allow for safe screening, Combined with an ℓ_2 penalty, the resulting ERM problem is strongly convex. We can therefore compare our Ellipsoid algorithm to the baseline introduced by [19], where the safe region is a ball centered in the current iterate of the solution and whose radius is $\frac{2\Delta}{\lambda}$ with Δ a duality gap of the ERM problem. Both methods are initialized by running the default solver of scikit-learn with a certain number of epochs. The resulting approximate solution and duality gap are subsequently fed into our algorithm for initialization. Then, we perform one more epoch of the duality gap screening algorithm on the one hand, and the corresponding number of ellipsoid steps computed on a subset of the dataset on the other hand, so as to get a fair comparison in terms of data access. The results can be seen in Table 2. While being more general (our approach is neither restricted to classification, nor requires strong convexity), our method performs similarly to the baseline. Figure 4 highlights the trade-off between optimizing and evaluating the gap (Duality Gap Screening) versus performing one step of Ellipsoid Screening. Key observations here is that both methods start screening after a correct iterate (i.e. with good test accuracy) is obtained by the solver (blue curve) thus underlining the fact that screening methods would rather be of practical use when computing a regularization path, or when the computing budget is less constrained (e.g. tracking or anomaly detection) which is the object of next paragraph.

Computational gains As demonstrated in Figure 5, computational gains can indeed be obtained in a regularization path setting (MNIST features, Squared Hinge Loss and L2 penalty). Each point of both curves

11

Epochs	20		30	
λ	MNIST	SVHN	MNIST	SVHN
1.0	89 / 89	87 / 87	89 / 89	87 / 87
10^{-1}	95 / 95	11 / 47	95 / 95	91 / 91
10^{-2}	16 / 84	0 / 0	98 / 98	90 / 92
10^{-3}	0 / 0	0 / 0	34 / 50	0 / 0

Table 2: Percentage of samples screened in an ℓ_2 penalized SVM with Squared Hinge loss (Ellipsoid (ours) / Duality Gap) given the epochs made at initialization.

Epochs	10	20
$\lambda = 1$	7 / 84	85 / 85
$\lambda = 10$	80 / 80	80 / 80
$\lambda = 100$	68 / 68	68 / 68

Table 3: RCV-1 : Percentage of samples screened in an ℓ_2 penalized SVM with Squared Hinge loss (Ellipsoid (ours) / Duality Gap) given the epochs made at initialization.

represents an estimator fitted for a given lambda against the corresponding cost (in epochs). Each estimator is initialized with the solution to the previous parameter lambda. On the orange curve, the previous solution is also used to initialize a screening. In this case, the estimator is fit on the remaining samples which further accelerates the path computation.

5.2   Dataset Compression

We now consider the problem of dataset compression, where the goal is to maintain a good accuracy while using less examples from a dataset. This section should be seen as a proof of concept. A natural scheme consists in choosing the samples that have a higher margin since those will carry more information than samples that are easy to fit. In this setting, our screening algorithm can be used for compression by using the scores of the screening test as a way of ranking the samples. In our experiments, and for a given model, we progressively delete data points according to their score in the screening test for this model, before fitting the model on the remaining subsets. We compare those methods to random deletions in the dataset and to a margin computed on early approximations of the solution when the loss admits a flat area.

Lasso regression.   The Lasso objective combines an ℓ_2 loss with an ℓ_1 penalty. Since its dual is not sparse, we will instead apply the safe rules offered by the screening-friendly regression loss (9) derived in Section 4.3 and illustrated in Figure 2, combined with an ℓ_1 penalty. We can draw an interesting parallel with the SVM, which is naturally sparse in data points. At the optimum, the solution of the SVM can be expressed in terms of data points (the so-called support vectors) that are close to the classification boundary, that is the points that are *the most difficult* to classify. Our screening rule yields the analog for regression: the points that are easy to predict, i.e. that are close to the regression curve, are less informative than the points that are harder to predict. In our experiments on *synthetic data* ($n = 100$), this does consistently better than random subsampling as can be seen in Figure 6.

Classification.   Our compression scheme is also valid for classification as can be seen in Figure 7.
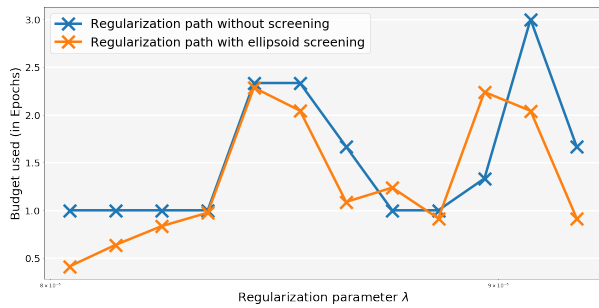
Acknowledgments

Figure 5: Regularization path of a Squared Hinge SVM trained on MNIST. The screening enables computational gains compared to a classical regularization path.
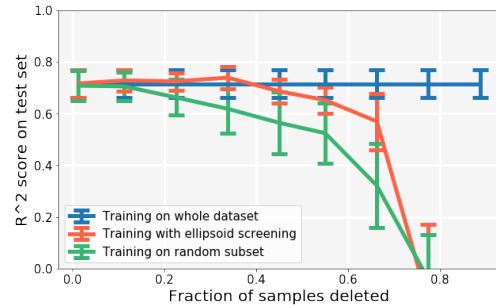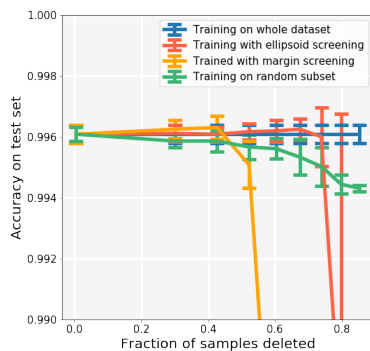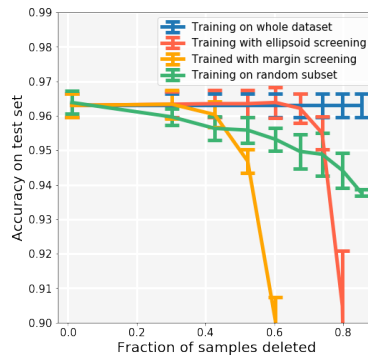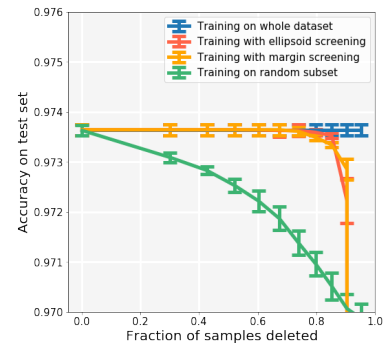

Figure 6: Dataset compression for the Lasso trained in a synthetic dataset. The scores given by the screening yield a ranking that is better than random subsampling.
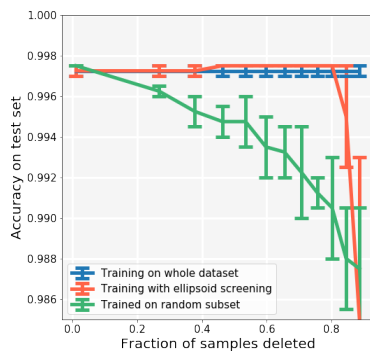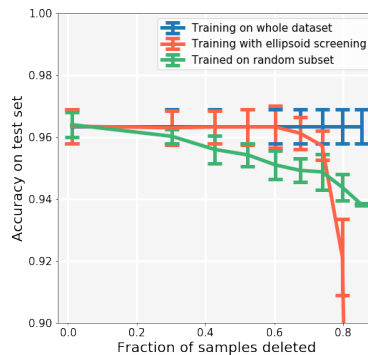

(a) MNIST and ℓ_2 Squared Hinge
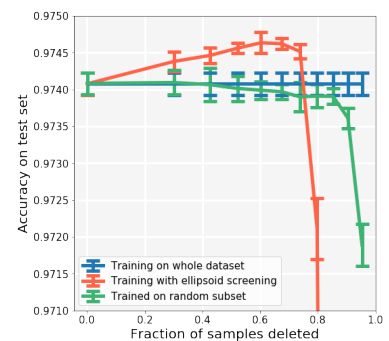

(b) SVHN and ℓ_2 Squared Hinge

(c) RCV-1 and ℓ_2 Squared Hinge

(d) MNIST and ℓ_1 Safe Logistic

(e) SVHN and ℓ_1 Safe Logistic

(f) RCV-1 and ℓ_1 Safe Logistic

Figure 7: Dataset compression in classification.

13

References

[1] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[2] A. Beck and M. Teboulle. Smoothing and first order methods: a unified framework. *SIAM J. Optim Vol. 22, No. 2*, 2012.

[3] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: A survey. *Operation Research*, 29, 1981.

[4] M. Blondel, A. F. T. Martins, and V. Niculae. Learning classifiers with fenchel-young losses: Generalized entropies, margins, and algorithms. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[6] L. Dai and K. Pelckmans. An ellipsoid based, two-stage screening test for bpdn. *European Signal Processing Conference*, pages 654–658, 01 2012.

[7] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe Feature Elimination for the LASSO and Sparse Supervised Learning Problems. *arXiv e-prints*, page arXiv:1009.4219, Sep 2010.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[9] O. Fercoq, A. Gramfort, and J. Salmon. Mind the duality gap: safer rules for the Lasso. In *International Conference on Machine Learning (ICML)*, 2015.

[10] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*. Springer series in statistics New York, 2001.

[11] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer, 1993.

[12] T. Hocking, G. Rigaill, J.-P. Vert, and F. Bach. Learning sparse penalties for change-point detection using max margin interval regression. In *International Conference on Machine Learning (ICML)*, 2013.

[13] J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advance in Neural Information Processing Systems (NIPS)*, 2016.

[14] J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A MAth*, 1962.

[15] A. Nemirovskii and D. Yudin. Problem complexity and method efficiency in optimization. *Nauka (published in English by John Wiley, Chichester, 1983)*, 1979.

[16] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[17] K. Ogawa, Y. Suzuki, and I. Takeuchi. Safe screening of non-support vectors in pathwise svm computation. In *International Conference on Machine Learning (ICML)*, 2013.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[19] A. Shibagaki, M. Karasuyama, K. Hatano, and I. Takeuchi. Simultaneous Safe Screening of Features and Samples in Doubly Sparse Modeling. In *International Conference on Machine Learning (ICML)*, 2016.

[20] I. Steinwart. Sparseness of support vector machines—some asymptotically sharp bounds. In *Advances in Neural Information Processing Systems*, 2004.

[21] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

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

[23] J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye. A safe screening rule for sparse logistic regression. In *Advance in Neural Information Processing Systems (NIPS)*, 2014.

[24] J. Wang, J. Zhou, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. In *Advance in Neural Information Processing Systems (NIPS)*, 2013.

[25] K. Yosida. Functional analysis. *Berlin-Heidelberg*, 1980.

[26] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

A Proofs.

A.1 Proof of Lemma 2.4

Proof. At the optimum,

$$P(x^*) - D(\nu^*) = \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^\top x) + f_i^*(\nu_i) + \lambda R(x) + \lambda R^* \left(-\frac{A^T \nu}{\lambda n} \right) = 0.$$

Adding the null term $\langle x, -\frac{A^\top \nu}{n} \rangle - \langle x, -\frac{A^\top \nu}{n} \rangle$ gives

$$\frac{1}{n} \sum_{i=1}^{n} \underbrace{f_i(a_i^\top x) + f_i^*(\nu_i) - a_i^\top x \nu_i}_{\geq 0} + \lambda \underbrace{\left(R(x) + R^* \left(-\frac{A^\top \nu}{\lambda n} \right) - \left\langle x, -\frac{A^\top \nu}{\lambda n} \right\rangle \right)}_{\geq 0} = 0,$$

since Fenchel-Young's inequality states that each term is greater or equal to zero. We have a null sum of non-negative terms; hence, each one of them is equal to zero. We therefore have for each $i = 1 \ldots n$:

$$f(a_i^\top x) + f^*(\nu_i) = a_i^\top x \nu_i,$$

which corresponds to the equality case in Fenchel-Young's relation, which is equivalent to $\nu_i^* \in \partial f_i(a_i^\top x^*)$. ∎

A.2 Proof of Lemma 3.3

Proof. The Lagrangian of the problem writes:

$$L(x, \nu, \gamma) = a_i^\top x - b_i + \nu \left(1 - (x - z)^T E^{-1}(x - z) \right) - \gamma g^T(x - z),$$

with $\nu, \gamma \geq 0$. When maximizing in x, we get:

$$\frac{\partial L}{\partial x} = a_i + 2\nu(E^{-1}z - E^{-1}x) - \gamma = 0.$$

We have $\nu > 0$ since the opposite leads to a contradiction. This yields $x = z + \frac{1}{2\nu}(Ea_i - \gamma Eg)$ and $(x - z)^T E^{-1}(x - z) = 1$ at the optimum which gives $\nu = \frac{1}{2} \sqrt{(a_i - \gamma)^T E(a_i - \gamma)}$.

Now, we have to minimize

$$g(\nu, \gamma) = a_i \left(z + \frac{1}{2\nu}(Ea_i - \gamma Eg) \right) - \gamma^\top \left(\frac{1}{2\nu}(Ea_i - \gamma Eg) \right).$$

To do that, we consider the optimality condition

$$\frac{\partial g}{\partial \gamma} = -\frac{1}{2\nu} a_i Eg - \frac{1}{2\nu} g^T Ea_i + \frac{\gamma}{\nu} g^T Eg = 0,$$

which yields $\gamma = \frac{g^T Ea_i}{g^T Eg}$. If $g^T Ea_i < 0$ then $\gamma = 0$ in order to avoid a contradiction.

In summary, either $g^T Ea_i \leq 0$ hence the maximum is attained in $x = z + \frac{1}{2\nu} Ea_i$ and is equal to $a_i z + \sqrt{a_i^T Ea_i} - y_i$, or $g^T Ea_i > 0$ and the maximum is attained in $x = z + \frac{1}{2\nu} E(a_i - \gamma Eg)$ and is equal to $a_i \left(z + \frac{1}{2\nu} E(a_i - \gamma g) \right) - b_i$ with $\nu = \frac{1}{2} \sqrt{(a_i - \gamma)^T E(a_i - \gamma)}$ and $\gamma = \frac{g^T Ea_i}{g^T Eg}$. ∎

A.3   Proof of Example 4.3

Proof. The Fenchel conjugate of a norm is the indicator function of the unit ball of its dual norm, the ℓ_∞ ball here. Hence the infimum convolution to solve

$$f_\mu(x) = \min_{z \in \mathbb{R}^n} \left\{ f(z) + \mathbf{1}_{\|x-z\|_\infty \leq \mu} \right\} \tag{12}$$

Since $f(x) = \frac{1}{2n}\|x\|_2^2$,

$$f_\mu(x) = \min_{z \in \mathbb{R}^n} \frac{1}{2n} z^T z + \mathbf{1}_{\|x-z\|_\infty \leq \mu}.$$

If we consider the change of variable $t = x - z$, we get:

$$f_\mu(x) = \min_{t \in \mathbb{R}^n} \frac{1}{2n} \|x - t\|_2^2 + \mathbf{1}_{\|t\|_\infty \leq \mu}.$$

The solution t^* to this problem is exactly the proximal operator for the indicator function of the infinity ball applied to x. It has a closed form

$$
\begin{aligned}
t^* &= \operatorname{prox}_{\mathbf{1}_{\|\cdot\|_\infty \leq \mu}}(x) \\
&= x - \operatorname{prox}_{(\mathbf{1}_{\|\cdot\|_\infty \leq \mu})^*}(x), \quad \text{using Moreau decomposition} \\
&= x - \operatorname{prox}_{\mu\|\cdot\|_1}(x).
\end{aligned}
$$

Hence,

$$f_\mu(x) = \frac{1}{2n}\|x - t^*\|_2^2 = \frac{1}{2n}\|\operatorname{prox}_{\mu\|\cdot\|_1}(x)\|_2^2.$$

But, $\operatorname{prox}_{\mu\|\cdot\|_1}(t) = \operatorname{sgn}(t) \times [|t| - \mu]_+$ for $t \in \mathbb{R}$, where $[x]_+ = \max(x, 0)$. ∎

B   Additional experimental results.

Experimental protocol and reproducibility.   The data sets did not require any pre-processing except *MNIST* and *SVHN* on which exhaustive details can be found in [13]. For both regression and classification, the examples were allocated to train and test sets using scikit-learn's *train-test-split*. The experiments were run three to ten times (depending on the cost of the computations) and our error bars reflect the standard deviation. For each fraction of points deleted, we fit three to five estimators on the screened dataset and the random subset before averaging the corresponding scores. The optimal parameters for the linear models were found using a simple grid-search.

Accuracy of our safe logistic loss.   The accuracies of the Safe Logistic loss we build is similar to the accuracies obtained with the Squared Hinge and the Logistic losses on the datasets we use in this paper thus making it a realistic loss function.

Dataset	MNIST	SVHN	RCV-1
Logistic + ℓ_1	0.997 (0.01)	0.99 (0.0003)	0.975 (1.0)
Logistic + ℓ_2	0.997 (0.001)	0.99 (0.0003)	0.975 (1.0)
Safelog + ℓ_1	0.996 (0.0)	0.989 (0.0)	0.974 (1e-05)
Safelog + ℓ_2	0.996 (0.0)	0.989 (0.0)	0.975 (1e-05)
Squared Hinge + ℓ_1	0.997 (0.03)	0.99 (0.03)	0.975 (1.0)
Squared Hinge + ℓ_2	0.997 (0.003)	0.99 (0.003)	0.974 (1.0)

Table 4: Averaged best accuracies on test set (best λ in a Logarithmic grid from $\lambda = 0.00001$ to 1.0).

Exemplar selection.   Here we generate respectively 100 and 400 redundant examples of *synthetic data* ($n = 100$) and *diabetes* ($n = 442$, $p = 10$, in scikit-learn) by forming convex combinations of existing data points and adding gaussian noise with zero mean. As in ranking data points for the Lasso, we apply our screening rules to iteratively discard examples that are redundant and fit a Lasso on the remaining dataset. This method greatly outperforms random subsets as can be seen in Figure 8.
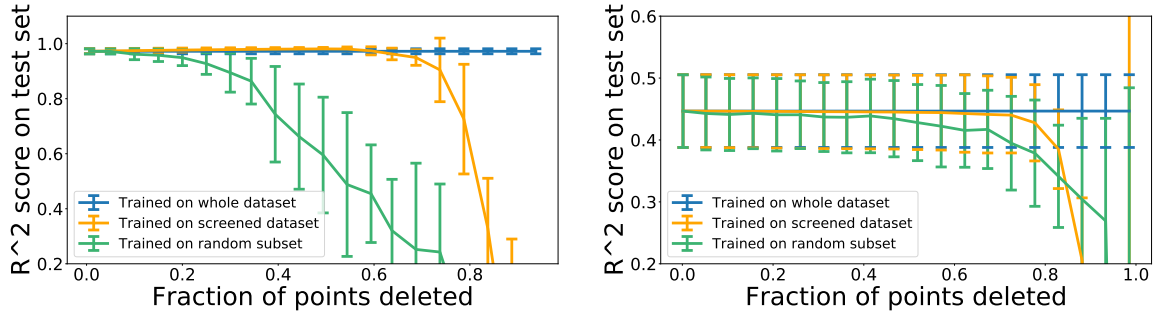

Figure 8: *Left*: Exemplar selection with 60 synthetic data points and 100 redundant, noisy data points. Our screening rule discards 60% of the dataset which corresponds to ~ 100 data points. *Right*: Diabetes. Exemplar selection with 400 real data points and 400 redundant, noisy data points. Our screening rule matches the accuracy of the original model while keeping 30% of the dataset only.