Robust sketching for multiple square-root LASSO problems

Vu Pham and Laurent El Ghaoui
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
\{vu,elghaoui\}@eecs.berkeley.edu

Abstract

Many learning tasks, such as cross-validation, parameter search, or leave-one-out analysis, involve multiple instances of similar problems, each instance sharing a large part of learning data with the others. We introduce a robust framework for solving multiple square-root LASSO problems, based on a sketch of the learning data that uses low-rank approximations. Our approach allows a dramatic reduction in computational effort, in effect reducing the number of observations from \(m\) (the number of observations to start with) to \(k\) (the number of singular values retained in the low-rank model), while not sacrificing—sometimes even improving—the statistical performance. Theoretical analysis, as well as numerical experiments on both synthetic and real data, illustrate the efficiency of the method in large scale applications.

1 Introduction

In many practical applications, learning tasks arise not in isolation, but as multiple instances of similar problems. A typical instance is when the same problem has to be solved, but with many different values of a regularization parameter. Cross-validation also involves a set of learning problems where the different “design matrices” are very close to each other, all being a low-rank perturbation of the same data matrix. Other examples of such multiple instances arise in sparse inverse covariance estimation with the LASSO (Friedman et al. (2008)), or in robust subspace clustering (Soltanolkotabi et al. (2014)). In such applications, it makes sense to spend processing time on the common part of the problems, in order to compress it in certain way, and speed up the overall computation.

In this paper we propose an approach to multiple-instance square root LASSO based on “robust sketching”, where the data matrix of an optimization problem is approximated by a sketch, that is, a simpler matrix that preserves some property of interest, and on which computations can be performed much faster than with the original. Our focus is a square-root LASSO problem:

\[
\min_{w \in \mathbb{R}^n} \| X^T w - y \|_2 + \lambda \| w \|_1
\]

where \(X \in \mathbb{R}^{n \times m}\) and \(y \in \mathbb{R}^m\). Square-root LASSO has pivotal recovery properties; also, solving a square-root LASSO problem is as fast as solving an equivalent LASSO problem with both first-order and second order methods (Belloni et al. (2011)). We chose the square-root version of the LASSO to make the derivations simpler; these derivations can also be adapted to the original LASSO problem, in which the loss function is squared.

In real-life data sets, the number of features \(n\) and the number of observations \(m\) can be both very large. A key observation is that real data often has structure that we can exploit. Figure 1 shows that real-life text data sets are often low-rank, or can be well-approximated by low-rank structures.

Contribution. Our objective is to solve multiple instances of square-root LASSO fast, each instance being a small modification to the same design matrix. Our approach is to first spend computational efforts in finding a low-rank sketch of the full data. With this sketch, we propose a robust model that takes into account the approximation error, and explain how to solve that approximate problem one order of magnitude faster, in effect reducing the number of observations from \(m\) (the number of observations to start with) to \(k\) (the number of singular values retained in the low-rank model). Together with our proposed model, we can perform cross validation, for example, an order of magnitude faster.
Robust sketching for multiple square-root LASSO problems

Solving multiple learning problems fast. We now turn to a fast solution to the low-rank elastic net problem (3), with algorithm-independent model for solving multiple LASSO problems fast. Our model can therefore be implemented with any generic convex solver, providing theoretical guarantees in computational savings while not sacrificing statistical performance.

Organization. The structure of our paper is as follows. In Section 2 we propose the robust square-root LASSO with sketched data. In Section 3 we present a simple method to reduce the dimension of the problem. Section 4 studies the effects of a non-robust framework. Section 5 provides a theoretical complexity analysis and we conclude our paper with numerical experiments in Section 6.

2 Robust Square-root LASSO

Low-rank elastic net. Assume we are given $\hat{X}$ as a sketch to the data matrix $X$, the robust square-root LASSO is defined as follows:

$$\phi_{\epsilon,\lambda}(\hat{X}) := \min_{w \in \mathbb{R}^n} \max_{X : \|X - \hat{X}\|_2 \leq \epsilon} \|X^T w - y\|_2 + \lambda \|w\|_1$$

where $X, \hat{X} \in \mathbb{R}^{n \times m}$, $y \in \mathbb{R}^m$, $y \neq 0$ and both $\lambda \geq 0, \epsilon \geq 0$ are given as our parameters. El Ghaoui and Lebret (1997) has shown that

$$\max_{\|\Delta\|_{\leq \epsilon}} \| (\hat{X} + \Delta)^T w - y\|_2 \leq \|X^T w - y\|_2 + \epsilon \|w\|_2$$

and the equality holds with the choice of $\Delta$ as

$$\Delta := \epsilon w^T$$

$$u := \begin{cases} X^T w - y \|X^T w - y\|_2 & \text{if } \hat{X}^T w \neq y \\ \text{any unit-norm vector} & \text{otherwise} \end{cases}$$

$$v := \begin{cases} w \|w\|_2 & \text{if } w \neq 0 \\ \text{any unit-norm vector} & \text{otherwise} \end{cases}$$

We also have $\text{rank}(\Delta) = 1$ and $\|\Delta\|_F = \|\Delta\|_2 = \epsilon$, which implies $\Delta$ is the worst-case perturbation for both the Frobenius and maximum singular value norm. Problem (2) can therefore be rewritten as:

$$\phi_{\epsilon,\lambda}(\hat{X}) = \min_{w \in \mathbb{R}^n} \|X^T w - y\|_2 + \epsilon \|w\|_2 + \lambda \|w\|_1$$

(3)

Note the presence of an “elastic net” term (Zou and Hastie (2005)), directly imputable to the error on the design matrix.

In our model, we employ the low-rank approximation of the original data matrix from any sketching algorithm: $\hat{X} = PQ^T$, where $P \in \mathbb{R}^{n \times k}$, $Q \in \mathbb{R}^{m \times k}$, $P$ and $Q$ have full rank $k$ with $k \ll \min\{m, n\}$. When the full data matrix $X \in \mathbb{R}^{n \times m}$ is approximated by $X \simeq PQ^T$, for leave-one-out analysis, the low rank approximation of the “design matrix” can be quickly computed by: $X_{\setminus i} \simeq PQ_{\setminus i}^T$ where $\setminus i$ means leaving out the $i$-th observation.

Solving the problem fast. We now turn to a fast solution to the low-rank elastic net problem (3), with
\( \hat{X} = PQ^T \). This “primal” problem is convex, and its dual is:

\[
\phi_{\lambda, \epsilon} = \min_{w, z \in \mathbb{R}^n} \|Qz - y\|_2 + \epsilon \|w\|_2 + \lambda \|w\|_1 : z = P^Tw
\]

\[
= \min_{w, z \in \mathbb{R}^n} \max_{u \in \mathbb{R}^k} \|Qz - y\|_2 + \epsilon \|w\|_2 + \lambda \|w\|_1 \\
+ u^T(z - P^Tw)
\]

\[
= \max_{u \in \mathbb{R}^k} \min_{w, z \in \mathbb{R}^n} \|Qz - y\|_2 + u^Tz + \\
\epsilon \|w\|_2 + \lambda \|w\|_1 - (Pu)^T w
\]

\[
= \max_{u \in \mathbb{R}^k} f_1(u) + f_2(u),
\]

where \( f_1(u) := \min_{z \in \mathbb{R}^n} \|Qz - y\|_2 + u^Tz \) and \( f_2(u) := \min_{w \in \mathbb{R}^n} \epsilon \|w\|_2 + \lambda \|w\|_1 - (Pu)^T w \).

**First subproblem.** Consider the first term in \( f_1(u) \):

\[
\|Qz - y\|_2^2 = z^TQ^TQz - 2y^TQz + y^Ty
\]

\[
= z^T \tilde{z} - 2c^T \tilde{z} + y^Ty
\]

where \( \tilde{z} := (Q^TQ)^{1/2}z \in \mathbb{R}^n \) and \( c := (Q^TQ)^{-1/2}Q^Ty \in \mathbb{R}^k \)

\[
= \|\tilde{z} - c\|^2_2 + y^Ty - c^Tc.
\]

Note that \( c^Tc = y^TQ(Q^TQ)^{-1}Q^Ty \leq y^Ty \) since \( Q(Q^TQ)^{-1}Q^T \preceq I \) is the projection matrix onto \( R(Q) \).

Letting \( s := \sqrt{y^Ty - c^Tc} \geq 0 \) gives

\[
f_1(u) := \min_{\tilde{z}} \|Qz - y\|_2 + u^Tz
\]

\[
= \min_{\tilde{z}} \sqrt{\|\tilde{z} - c\|^2_2 + s^2 + u^T \tilde{z}}
\]

by letting \( \tilde{\tilde{u}} := (Q^TQ)^{-1/2}u \)

\[
= \tilde{\tilde{u}}^Tc + \min_{\tilde{z}} \sqrt{\|x\|^2_2 + s^2 - \tilde{\tilde{u}}^T x}
\]

by letting \( x := c - \tilde{\tilde{z}} \).

Now consider the second term \( \min_{\tilde{\tilde{z}}} \sqrt{\|x\|^2_2 + s^2 - \tilde{\tilde{u}}^T x} \).

The optimal \( x^* \) must be in the direction of \( \tilde{\tilde{u}} \). Letting \( x := \alpha \tilde{\tilde{u}}, \alpha \in \mathbb{R} \), we have the expression

\[
\min_{\alpha \in \mathbb{R}} \sqrt{\alpha^2\|\tilde{\tilde{u}}\|^2_2 + s^2 - \alpha \tilde{\tilde{u}}^T \tilde{\tilde{u}}}
\]

When \( \|\tilde{\tilde{u}}\|_2 \geq 1 \), the problem is unbounded below. When \( \|\tilde{\tilde{u}}\|_2 < 1 \), the optimal solution is \( \alpha^* = \frac{s}{\sqrt{1 - \|\tilde{\tilde{u}}\|^2_2}} \)

and the optimal value is thus \( s \sqrt{1 - \|\tilde{\tilde{u}}\|^2_2} \).

The closed-form expression for \( f_1(u) \) is therefore:

\[
f_1(u) = \tilde{\tilde{u}}^Tc + \min_{\tilde{\tilde{z}}} \sqrt{\|x\|^2_2 + s^2 - \tilde{\tilde{u}}^T x}
\]

\[
= \tilde{\tilde{u}}^Tc + \sqrt{1 - \|\tilde{\tilde{u}}\|^2_2}
\]

\[
= u^T(Q^TQ)^{-1/2}c + \sqrt{1 - u^T(Q^TQ)^{-1}u}
\]

\[
= u^T K^{-1/2}c + \sqrt{1 - u^TK^{-1}u}
\]

by letting \( K := Q^TQ \).

(4)

**Second subproblem.** Consider the function \( f_2(u) := \min_{w \in \mathbb{R}^n} \epsilon \|w\|_2 + \lambda \|w\|_1 - (Pu)^T w \). We observe that the objective function is homogeneous. Strong duality gives:

\[
f_2(u) = \min_{w, v, r} r^T w + v^T w - (Pu)^T w
\]

s.t. \( \|r\|_2 \leq \epsilon, \|v\|_\infty \leq \lambda \)

\[
= \max_{u, v, r} (r + v)^T w
\]

s.t. \( \|r\|_2 \leq \epsilon, \|v\|_\infty \leq \lambda \)

\[
= \max_{u, v, r} 0
\]

s.t. \( \|r\|_2 \leq \epsilon, \|v\|_\infty \leq \lambda, Pu = v + r \)

(5)

Hence \( f_2(u) = 0 \) if there exists \( v, r \in \mathbb{R}^n \) satisfying the constraints. Otherwise \( f_2(u) \) is unbounded below.

**Dual problem.** From (4) and (5), the dual problem to (3) can be derived as:

\[
\phi_{\lambda, \epsilon} = \max_{u \in \mathbb{R}^n, v, r, t, \lambda, \epsilon} u^T K^{-1/2}c + \sqrt{1 - u^TK^{-1}u}
\]

s.t. \( \|v\|_\infty \leq \lambda, \|r\|_2 \leq \epsilon, Pu = v + r \)

Letting \( R := PK^{1/2} \in \mathbb{R}^{n \times k} \) and replacing \( u \) by \( K^{-1/2}u \), we have

\[
\phi_{\lambda, \epsilon} = \max_{u, v, r, t, \lambda, \epsilon} u^TK^{-1/2}c + s\sqrt{1 - u^TK^{-1}u}
\]

s.t. \( \|v\|_\infty \leq \lambda, \|r\|_2 \leq \epsilon, Ru = v + r \)

\[
= \max_{u, v, r, t} u^TKc + s\sqrt{1 - u^TKu}
\]

s.t. \( \|v\|_\infty \leq \lambda, \|r\|_2 \leq \epsilon, Ru = v + r \)

(6)

**Bidual problem.** The bidual of (3) writes

\[
\phi_{\lambda, \epsilon} = \max_{w, v, r, t, \lambda, \epsilon} \left[ c - R^Tw \right]^T u + w^Tv + w^TRr
\]

s.t. \( \|u\|_2 \leq 1, \|v\|_\infty \leq \lambda, \|r\|_2 \leq \epsilon \)

\[
= \min_{w, v, r, t, \lambda, \epsilon} \left[ c - R^Tw \right]^T u + w^Tv + w^TRr
\]

s.t. \( \|u\|_2 \leq 1, \|v\|_\infty \leq \lambda, \|r\|_2 \leq \epsilon \)

Therefore,

\[
\phi_{\lambda, \epsilon} = \min_{w, v, r, t, \lambda, \epsilon} \left[ c - R^Tw \right]^T u + \epsilon \|w\|_2 + \lambda \|w\|_1
\]

(7)

where \( R \in \mathbb{R}^{n \times k}, c \in \mathbb{R}^k \) and \( s \in \mathbb{R} \). Note that problem (7) still involves \( n \) variables in \( w \), but now the size of the design matrix is only \( n \)-by-\( k \), instead of \( n \)-by-\( m \) as the original problem.

**Summary.** To solve problem (3) with \( X = PQ^T \) we first set \( c := (Q^TQ)^{-1/2}Q^Ty, s := \sqrt{y^Ty - c^Tc} \),
Robust sketching for multiple square-root LASSO problems

\[ R := P(Q^T Q)^{1/2}, \text{ then solve the problem (7) above.} \]

As discussed later, the worst-case complexity grows as \( O(kn^2 + k^3) \). This is in contrast with the original problem (3) when no structure is exploited, in which case the complexity grows as \( O(mn^2 + m^3) \).

3 Safe Feature Elimination

In this section we present a method to reduce the dimension of (7) without changing the optimal value, as proposed by El Ghaoui et al. (2012). Let us define \( A := \begin{bmatrix} R^T \\ 0 \end{bmatrix} \in \mathbb{R}^{(k+1) \times n} \) and \( b := \begin{bmatrix} c \\ s \end{bmatrix} \in \mathbb{R}^{k+1} \), problem (7) becomes:

\[
\phi_{\lambda, \epsilon} = \min_{w \in \mathbb{R}^n} \|Aw - b\|_2 + \epsilon \|w\|_2 + \lambda \|w\|_1 \quad (8)
\]

Problem (8) is equivalent to:

\[
\phi_{\lambda, \epsilon} = \min_{w \in \mathbb{R}^n} \max_{\alpha, \beta, \gamma} \alpha^T (Aw - b) + \beta^T w + \gamma^T w \\
\text{s.t.} \quad \|\alpha\|_2 \leq 1, \|\beta\|_2 \leq \epsilon, \|\gamma\|_\infty \leq \lambda \\
= \max_{\|\alpha\|_2 \leq 1} \min_{w \in \mathbb{R}^n} \alpha^T \left( A^T \alpha + \beta + \gamma \right) - \alpha^T b \\
= \max_{\|\alpha\|_2 \leq \epsilon, \|\beta\|_2 \leq \epsilon, \|\gamma\|_\infty \leq \lambda} - \alpha^T b \\
\text{s.t.} \quad \|\alpha\|_2 \leq 1, \|\beta\|_2 \leq \epsilon, \|\gamma\|_\infty \leq \lambda, \\
A^T \alpha + \beta + \gamma = 0
\]

where \( a_i \)'s are columns of \( A \). If \( \|a_i\|_2 \leq \lambda - \epsilon \), we always have \( |a_i^T \alpha| \leq \lambda \). In other words, we can safely discard the \( i \)-th feature without affecting our optimal value.

4 Non-robust square-root LASSO

In practice, a simple idea is to replace the data matrix by its low rank approximation in the model. We refer to this approach as the non-robust square-root LASSO:

\[
\min_{w \in \mathbb{R}^n} \|QP^T w - b\|_2 + \lambda \|w\|_1 \quad (9)
\]

For many learning applications, this approach first appears as an attractive heuristic to speed up the computation. Nevertheless, in problems with sparsity as the main emphasis, care must be taken in the presence of regularization involving the \( l_1 \)-norm. Figure 2(a) shows an example of a non-robust square-root LASSO with data replaced by its rank-1 approximation. The optimal solution then always has a cardinality at most 1, and the tuning parameter \( \lambda \) does not provide any sparsity control, unlike the robust low-rank model in Figure 2(b). In general, replacing our data with its low-rank approximation will result in the lost of the sparsity control from regularization parameters. We provide an insight for this absence of sparsity control in the following theorem.

**Theorem 1** For the non-robust square-root LASSO problem (9), with \( P \in \mathbb{R}^{n \times k} \) and \( Q \in \mathbb{R}^{m \times k} \) full rank where \( k \ll \min\{m, n\} \), there exists a LASSO solution with cardinality at most \( k \).

**Proof.** Uniquely decomposing \( b \) into \( b = Qz + u \) where \( u \perp \mathcal{R}(Q) \) gives

\[
\min_{w \in \mathbb{R}^n} \|QP^T w - z\|_2 + \lambda \|w\|_1 \\
= \min_{w \in \mathbb{R}^n} \sqrt{\|QP^T w - z\|_2^2 + \|u\|_2^2} + \lambda \|w\|_1 \\
= \min_{w \in \mathbb{R}^n} \|w\|_1 : P^T w = P^T w_0
\]

Let \( w_0 \) be any optimal solution to this problem, it suffices to show that the problem

\[
\min_{w \in \mathbb{R}^n} \|w\|_1 : P^T w = P^T w_0
\]

has an optimal solution with cardinality at most \( k \). We prove this in the following lemma:
Lemma 1 The problem
\[ \min_{x \in \mathbb{R}^n} \| x \|_1 : Ax = b \] (10)
with \( A \in \mathbb{R}^{k \times n} \) wide \((k < n)\) and \( b \in \mathcal{R}(A) \) has an optimal solution with cardinality at most \( k \).

Proof. Our proof is adapted from Tibshirani et al. (2013) on the existence and uniqueness of the solution. Let \( x \in \mathbb{R}^n \) be an optimal solution to (10). Without loss of generality, we can assume all components of \( x \) are non-zeros (if some components are zeros one can discard the corresponding columns of \( A \)).

If \( \text{card}(x) > k \), we provide a procedure to reduce the cardinality of \( x \), while keeping the same \( l_1 \)-norm and constraint feasibility. Let \( s \in \mathbb{R}^n \) be the (unique) subgradient of \( \| x \|_1 \); \( s_i := \text{sign}(x_i), i = 1, \ldots, n \). The optimality condition of (10) shows that \( \exists \mu \in \mathbb{R}^n \) such that \( A^T \mu = s \). Since all the columns \( A_i \)'s are linearly dependent, there exist \( i \) and \( c_i \) such that
\[
\begin{align*}
A_i &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j A_j, & \text{where } \mathcal{E} := \{1, \ldots, n\} \\
A_i^T \mu &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j A_j^T \mu \\
s_i \mu &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j s_j
\end{align*}
\]
Therefore \( 1 = s_i^2 = \sum_{j \in \mathcal{E} \setminus \{i\}} c_j s_j s_i \). Defining \( d_j := c_j s_j s_i \) gives \( \sum_{j \in \mathcal{E} \setminus \{i\}} d_j = 1 \) and
\[
\begin{align*}
s_i A_i &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j s_j A_j \\
&= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j s_j s_i s_j A_j & \text{since } s_j^2 = 1 \\
&= \sum_{j \in \mathcal{E} \setminus \{i\}} d_j s_j A_j
\end{align*}
\]
Let us define a direction vector \( \theta \in \mathbb{R}^n \) as follows:
\[
\theta_i := -s_i \text{ and } \theta_j := d_j s_j, j \in \mathcal{E} \setminus \{i\}. \quad \text{Then } A \theta = (-s_i A_i) + \sum_{j \in \mathcal{E} \setminus \{i\}} d_j s_j A_j = 0.
\]
Thus letting
\[
x^{(\rho)} := x + \rho \theta \text{ with } \rho > 0
\]
we have \( x^{(\rho)} \) feasible and its \( l_1 \) norm stays unchanged:
\[
\| x^{(\rho)} \|_1 = \| x \|_1 + \sum_{j \in \mathcal{E} \setminus \{i\}} \| x^{(\rho)} \|_1 = \| x \|_1 + \rho \sum_{j \in \mathcal{E} \setminus \{i\}} d_j = \| x \|_1 + \rho \| d \|_1
\]
Choosing \( \rho := \min(t \geq 0 : x_j + \rho \theta_j = 0 \text{ for some } j) \) we have one fewer non-zeros components in \( x^{(\rho)} \). Note that \( \rho \leq \| x \|_1 \). Therefore, repeating this process gives an optimal solution \( x \) of at most \( k \) non-zeros components. ■

Remark. An alternative proof is to formulate problem (10) as a linear program, and observe that the optimal solution is at a vertex of the constraint set. Our result is also consistent with the simple case when the design matrix \( A \) has more features than observations, there exists an optimal solution of the LASSO problem with cardinality at most the number of observations, as shown by many authors (Tibshirani et al. (2013)).

5 Theoretical Analysis

Our objective in this section is to analyze the theoretical complexity of solving problem (8):
\[ \min_{x \in \mathbb{R}^n} \| Ax - b \|_2 + \epsilon \| x \|_2 + \lambda \| x \|_1 \]
where \( A \in \mathbb{R}^{k \times n}, b \in \mathbb{R}^k \), and the optimization variable is now \( x \in \mathbb{R}^n \). We present an analysis for a standard second-order methods via log-barrier functions. We also remark that with a generic primal-dual interior point method, our robust model can effectively solve problems of \( 3 \times 10^5 \) observations and \( 10^5 \) features in just a few seconds. In practical applications, specialized interior-point methods for specific models with structures can give very high performance for large-scale problems, such as in Kim et al. (2007) or in Koh et al. (2007). Our paper, nevertheless, does not focus on developing a specific interior-point method for solving square-root LASSO; instead we focus on a generalized model and the analysis of multiple instances with a standard method.

5.1 Square-root LASSO

In second-order methods, the main cost at each iteration is from solving a linear system of equations involving the Hessian of the barrier function (Andersen et al. (2003)). Consider the original square-root LASSO problem:
\[
\min_{x \in \mathbb{R}^n} \| Ax - b \|_2 = \min_{x \in \mathbb{R}^n, s \in \mathbb{R}} s : \| Ax - b \|_2 \leq s
\]
The log-barrier function is \( \varphi_c(x, s) = C \varepsilon - \log \left( s^2 - \| Ax - b \|_2^2 \right) \). The cost is from evaluating the inverse Hessian of
\[ f := -\log \left( (s^2 - (Ax - b)^T(Ax - b)) \right) \]
Let \( g := -\log (s^2 - w^T w) \), we have \( \nabla g = \frac{2}{s} \begin{bmatrix} -w \\ 0 \end{bmatrix} \) and \( \nabla^2 g = \frac{2}{s} \begin{bmatrix} -I & 0 \\ 0 & 1 \end{bmatrix} + \nabla g \nabla g^T \). The Hessian \( \nabla^2 g \) is therefore a diagonal plus a dyad.

For (11), rearranging the variables as \( \tilde{x} = \begin{bmatrix} x \\ s \end{bmatrix} \) gives
\[
\begin{bmatrix} Ax - b \\ s \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} = \tilde{A} \tilde{x} - \tilde{b} \text{ where } \tilde{A} \in \mathbb{R}^{(k+1) \times n} \text{ and:}

\begin{align*}
\nabla^2 f &= \tilde{A}^T \left( \frac{2}{\tilde{A}} \begin{bmatrix} -I & 0 \\ 0 & 1 \end{bmatrix} + \frac{4}{\tilde{A}} \begin{bmatrix} -Ax + b \\ -Ax + b \end{bmatrix} \begin{bmatrix} -Ax + b \\ s \end{bmatrix} \right) \tilde{A} \\
&= \frac{2}{\tilde{A}} \tilde{A}^T \begin{bmatrix} -I & 0 \\ 0 & 1 \end{bmatrix} \tilde{A} \\
&+ \frac{4}{\tilde{A}^T} \begin{bmatrix} -Ax + b \\ s \end{bmatrix} \begin{bmatrix} -Ax + b \\ s \end{bmatrix}^T
\end{align*}
\]
(12)
Robust sketching for multiple square-root LASSO problems

The Hessian $\nabla^2 f$ is therefore simply a $(k+2)$-dyad.

5.2 Regularized square-root LASSO

For (8), decomposing $x = p - q$ with $p \geq 0, q \geq 0$ gives

$$\phi = \min_{w \in \mathbb{R}^n} \| A z - b \|_2 + \varepsilon \| x \|_2 + \lambda \| x \|_1$$

$$= \min_{p,q \in \mathbb{R}^n} \min_{s,t \in \mathbb{R}^n} \left( \begin{array}{c} s + ct + \lambda (1^T p + 1^T q) \\ s + bt + \lambda (1^T p + 1^T q) \end{array} \right)$$

$$\text{s.t. } \| p - q \|_2 \leq t, \| A(p - q) - b \|_2 \leq s, p \geq 0, q \geq 0$$

The log-barrier function is thus

$$\varphi_\gamma(p, q, s, t) = \gamma \left( s + ct + \lambda (1^T p + 1^T q) \right)$$

$$- \log \left( t^2 - \| p - q \|_2^2 \right)$$

$$- \log \left( s^2 - \| A(p - q) - b \|_2^2 \right)$$

$$- \sum_{i=1}^n \log(p_i) - \sum_{i=1}^n \log(q_i)$$

$$- \log(s) - \log(t).$$

First log term. Let $l_1 := - \log \left( t^2 - \| p - q \|_2^2 \right)$. Rearranging our variables as $x = [p_1, q_1, \ldots, p_n, q_n, t]^T$, we have

$$\nabla l_1 = -\frac{2}{l_1} \begin{bmatrix} p_1 - q_1, q_1 - p_1, \cdots, p_n - q_n, q_n - p_n, t \end{bmatrix}^T$$

$$\nabla^2 l_1 = -\frac{2}{l_1^2} \begin{bmatrix} B & \cdots & B \\ \vdots & \ddots & \vdots \\ B & \cdots & B \end{bmatrix} + \nabla l_1 (\nabla l_1)^T$$

where there are $n$ blocks of $B := \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}$ in the Hessian $\nabla^2 l_1$.

Second log term. Let $l_2 := - \log \left( s^2 - \| A(p - q) - b \|_2^2 \right)$. Keeping the same arrangement of variables $x = [p_1, q_1, \ldots, p_n, q_n, s]^T$, we have

$$\begin{bmatrix} A(p - q) - b \\ s \end{bmatrix} = A \hat{x}$$

where $A \in \mathbb{R}^{(k+1) \times (2n+1)}$. Following (12), we have the Hessian is a $(k+2)$-dyad.

Third log term. Let $l_3 := - \sum_{i=1}^n \log(p_i) - \sum_{i=1}^n \log(q_i) - \log(s) - \log(t)$. Every variable is decoupled; therefore the Hessian is simply diagonal.

Summary. The Hessian of the log barrier function $\varphi_\gamma(p, q, s, t)$ is a block diagonal plus a $(k+2)$-dyad. At each iteration of second-order method, inverting the Hessian following the matrix inversion lemma costs $O(kn^2)$. For the original square-root LASSO problem (1), using similar methods will cost $O(mn^2)$ at each iteration (Andersen et al. (2003)).

6 Numerical Results

In this section, we perform experiments on both synthetic data and real-life data sets on different learning tasks. The data sets\footnote{All data sets are available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/} are of varying sizes, ranging from small, medium and large scales (Table 1). To compare our robust model and the full model, we run all experiments on the same workstation at 2.3 GHz Intel core i7 and 8 GB memory. Both models have an implementation of the generic second-order algorithm from Mosek solver (Andersen and Andersen (2000)). For low-rank approximation, we use the simple power iteration methods. To make the comparison impartial, we do not use the safe feature elimination technique presented in Section 3 in our robust model.

<table>
<thead>
<tr>
<th>Data set</th>
<th>#train</th>
<th>#test</th>
<th>#features</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Newsgroups</td>
<td>15,935</td>
<td>3,993</td>
<td>62,961</td>
<td>sparse</td>
</tr>
<tr>
<td>RCV1.binary</td>
<td>20,242</td>
<td>677,399</td>
<td>47,236</td>
<td>sparse</td>
</tr>
<tr>
<td>SIAM 2007</td>
<td>21,519</td>
<td>7,077</td>
<td>30,438</td>
<td>sparse</td>
</tr>
<tr>
<td>Real-sim</td>
<td>72,309</td>
<td>1,000</td>
<td>5,000</td>
<td>dense</td>
</tr>
<tr>
<td>NIPS papers</td>
<td>1,500</td>
<td>N/A</td>
<td>12,419</td>
<td>sparse</td>
</tr>
<tr>
<td>NYTtimes</td>
<td>300,000</td>
<td>N/A</td>
<td>102,660</td>
<td>sparse</td>
</tr>
<tr>
<td>Random 1</td>
<td>500</td>
<td>N/A</td>
<td>100</td>
<td>dense</td>
</tr>
<tr>
<td>Random 2</td>
<td>625</td>
<td>N/A</td>
<td>125</td>
<td>dense</td>
</tr>
<tr>
<td>Random 3</td>
<td>750</td>
<td>N/A</td>
<td>150</td>
<td>dense</td>
</tr>
<tr>
<td>Random 19</td>
<td>2750</td>
<td>N/A</td>
<td>550</td>
<td>dense</td>
</tr>
<tr>
<td>Random 20</td>
<td>2875</td>
<td>N/A</td>
<td>575</td>
<td>dense</td>
</tr>
<tr>
<td>Random 21</td>
<td>3000</td>
<td>N/A</td>
<td>600</td>
<td>dense</td>
</tr>
</tbody>
</table>

6.1 Complexity on synthetic data

Our objective in this experiment is to compare the actual computational complexity with the theoretical analysis presented in Section 5. We generated dense and i.i.d. random data for $n = 100 \ldots 600$. At each $n$, a data set of size $5n$-by-$n$ is constructed. We keep $k$ fixed across all problem sizes, run the two models and compute the ratio between the running time of our model to that of the full model. The running time of our model is the total computational time of the data sketching phase and the training phase. The experiment is repeated 100 times at each problem size. As Figure 3 shows, the time ratio grows asymptotically as $O(1/n)$, a reduction of an order of magnitude in consistent with the theoretical result in Section 5.

6.2 Cross validation and leave-one-out

In this experiment, we focus on the classical 5-fold cross validation on real-life data sets. Figure 4 shows the running time (in CPU seconds) from $k = 1 \ldots 50$ for 5-fold cross validation on Gisette data, the hand-
written digit recognition data from NIPS 2003 challenge (Guyon et al. (2004)). It takes our framework less than 40 seconds, while it takes 22,082 seconds (500 times longer) for the full model to perform 5-fold cross validation. Furthermore, with leave-one-out analysis, the running time for the full model would require much more computations, becoming impractical while our model only needs a total of 7,684 seconds, even less than the time to carry out 5-fold cross validation on the original model. Table 2 reports the experimental results on other real-life data sets.

Table 2: Comparisons of 5-fold cross-validation on real data sets (in CPU time).

<table>
<thead>
<tr>
<th>Data set</th>
<th>Original model (seconds)</th>
<th>Our model (seconds)</th>
<th>Saving factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gisette</td>
<td>22,082</td>
<td>39</td>
<td>566</td>
</tr>
<tr>
<td>20 Newsgroups</td>
<td>17,731</td>
<td>65</td>
<td>272</td>
</tr>
<tr>
<td>RCV1.binary</td>
<td>17,776</td>
<td>70.8</td>
<td>251</td>
</tr>
<tr>
<td>SIAM 2007</td>
<td>9,025</td>
<td>67</td>
<td>134</td>
</tr>
<tr>
<td>Real-sim</td>
<td>73,764</td>
<td>56.3</td>
<td>1310</td>
</tr>
</tbody>
</table>

Figure 3: The ratio between the running time of our robust model and the original model.

Figure 4: 5-fold cross validation on Gisette data with robust approximation model.

6.3 Statistical performance

We further evaluate our model on statistical learning performance with binary classification task on both Gisette and RCV1 data sets. RCV1 is a sparse text corpus from Reuters while Gisette is a very dense pixel data. For evaluation metric, we use the F1-score on the testing sets. As Figure 5 and Figure 6 show, the classification performance is equivalent to the full model. As far as time is concerned, the full model requires 5,547.1 CPU seconds while our framework needs 18 seconds for \( k = 50 \) on RCV1 data set. For Gisette data, the full model requires 991 seconds for training and our framework takes less than 34 seconds.

6.4 Topic imaging

One application of solving multiple learning problems is topic imaging. Topic imaging is analogous in spirit to leave-one-out analysis. Instead of leave-one-observation-out, topic imaging removes a feature and runs a LASSO model on the remaining so as to explore the “neighborhood” (topic) of the query feature. Data sketching is computed only once for each data set and is shared to answer all queries in parallel.

We experiment our robust sketching model on two large text corpora: NIPS full papers and New York Times articles (Bache and Lichman (2013)). Table 3 and Table 4 show the results to sample queries on NIPS and NYTimes as well as the computational time our model takes to answer these queries. In both data sets, our model gives the result in just a few seconds. We can see the topic of Statistics, or Vision (Computer vision) with NIPS (Table 3) and the theme of Political and Research with NYTimes data (Table 4).

7 Concluding Remarks

We proposed in this paper a robust sketching model to approximate the task of solving multiple learning problems. We illustrate our approach with the square-root LASSO model given a low-rank sketch of the original data set. The numerical experiments suggest this framework is highly scalable, gaining one order of magnitude in computational complexity over the full model.

One interesting direction is to extend this model to a different data approximation, such as sparse plus low-rank (Chandrasekaran et al. (2011)), in order to capture more useful information while keeping the structures simple in our proposed framework. Our future works also include an analysis and implementation of this framework using first-order techniques for very large-scale problems.
Robust sketching for multiple square-root LASSO problems

Figure 5: Classification performance and running time on RCV1 data set.

(a) Performance on binary classification. (b) Training time of our model. The time to train the full model is 5547.1 seconds.

Figure 6: Classification performance and running time on Gisette data set.

(a) Performance on binary classification. (b) Training time of our model. The time to train the full model is 991 seconds.

Table 3: Topic imaging for 5 query words on NIPS papers.

<table>
<thead>
<tr>
<th>Query</th>
<th>LEARNING</th>
<th>STATISTIC</th>
<th>OPTIMIZATION</th>
<th>TEXT</th>
<th>VISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>3.15</td>
<td>2.89</td>
<td>3.11</td>
<td>3.92</td>
<td>3.15</td>
</tr>
<tr>
<td>1</td>
<td>error</td>
<td>data</td>
<td>algorithm</td>
<td>trained</td>
<td>object</td>
</tr>
<tr>
<td>2</td>
<td>action</td>
<td>distribution</td>
<td>data</td>
<td>error</td>
<td>image</td>
</tr>
<tr>
<td>3</td>
<td>algorithm</td>
<td>model</td>
<td>distribution</td>
<td>generalization</td>
<td>visiting</td>
</tr>
<tr>
<td>4</td>
<td>targeting</td>
<td>error</td>
<td>likelihood</td>
<td>wooter</td>
<td>images</td>
</tr>
<tr>
<td>5</td>
<td>weighed</td>
<td>algorithm</td>
<td>variable</td>
<td>classifier</td>
<td>unit</td>
</tr>
<tr>
<td>6</td>
<td>trained</td>
<td>parameter</td>
<td>network</td>
<td>student</td>
<td>recognition</td>
</tr>
<tr>
<td>7</td>
<td>uniqueness</td>
<td>trained</td>
<td>mixture</td>
<td>validating</td>
<td>representation</td>
</tr>
<tr>
<td>8</td>
<td>reinforced</td>
<td>likelihood</td>
<td>param</td>
<td>trainable</td>
<td>motion</td>
</tr>
<tr>
<td>9</td>
<td>control</td>
<td>gaussian</td>
<td>bound</td>
<td>hidden</td>
<td>view</td>
</tr>
<tr>
<td>10</td>
<td>policies</td>
<td>set</td>
<td>bayesian</td>
<td>hmm</td>
<td>field</td>
</tr>
</tbody>
</table>

Table 4: Topic imaging for 5 query words on New York Times articles.

<table>
<thead>
<tr>
<th>Query</th>
<th>HEALTH</th>
<th>TECHNOLOGY</th>
<th>POLITICAL</th>
<th>BUSINESS</th>
<th>RESEARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (s)</td>
<td>11.81</td>
<td>11.84</td>
<td>10.95</td>
<td>11.93</td>
<td>10.65</td>
</tr>
<tr>
<td>1</td>
<td>drug</td>
<td>weaving</td>
<td>campaign</td>
<td>companionship</td>
<td>drug</td>
</tr>
<tr>
<td>2</td>
<td>patience</td>
<td>companies</td>
<td>election</td>
<td>companias</td>
<td>patient</td>
</tr>
<tr>
<td>3</td>
<td>doctor</td>
<td>com</td>
<td>presidency</td>
<td>milling</td>
<td>cell</td>
</tr>
<tr>
<td>4</td>
<td>cell</td>
<td>computer</td>
<td>vortices</td>
<td>stmurray</td>
<td>doctor</td>
</tr>
<tr>
<td>5</td>
<td>perceiving</td>
<td>site</td>
<td>republic</td>
<td>marker</td>
<td>percent</td>
</tr>
<tr>
<td>6</td>
<td>disease</td>
<td>company</td>
<td>tawny</td>
<td>customary</td>
<td>disease</td>
</tr>
<tr>
<td>7</td>
<td>tawny</td>
<td>wwi</td>
<td>voted</td>
<td>weaving</td>
<td>program</td>
</tr>
<tr>
<td>8</td>
<td>medica</td>
<td>online</td>
<td>democratic</td>
<td>analyst</td>
<td>tessie</td>
</tr>
<tr>
<td>9</td>
<td>cancer</td>
<td>sites</td>
<td>presidentes</td>
<td>firing</td>
<td>medical</td>
</tr>
<tr>
<td>10</td>
<td>care</td>
<td>customer</td>
<td>leader</td>
<td>billion</td>
<td>study</td>
</tr>
</tbody>
</table>
References


