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:

$$\min_{w \in \mathbf{R}^n} \left\| X^T w - y \right\|_2 + \lambda \|w\|_1 \tag{1}$$

where $X \in \mathbf{R}^{n \times m}$ and $y \in \mathbf{R}^m$. 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
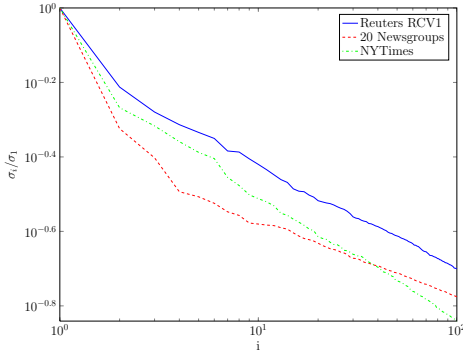
Figure 1: Graphs of the top 100 singular values from real-life text data sets.

faster than the traditional method, with the sketching computation included in our approach.

This paper employs low-rank sketching for data approximation phase, for which an extensive body of algorithmic knowledge exists, including power method, random projection and random sampling, or Nyström methods (Miranian and Gu (2003); Drineas and Mahoney (2005); Drineas et al. (2006); Halko et al. (2011); Mahoney (2011); Liberty (2013)). Our framework works with any approximation algorithms, thus provides flexibility when working with different types of sketching methods, and remains highly scalable in learning tasks.

Related work. Solving multiple learning problems has been widely studied in the literature, mostly in the problem of computing the regularization path (Park and Hastie (2007)). The main task in this problem is to compute the full solutions under different regularization parameters. The most popular approach includes the warm-start technique, which was first proposed in specific optimization algorithms (e.g. Yildirim and Wright (2002)), then applied in various statistical learning models, for example in (Kim et al. (2007); Koh et al. (2007); Garrigues and Ghaoui (2009)). Recent works (Tsai et al. (2014)) show strong interest in incremental and decremental training, and employ the same warm-start technique. These techniques are all very specific to the multiple learning task at hand, and require developing a specific algorithm for each case.

In our approach, we propose a generic, robust, and 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:

$$
\begin{aligned}
\phi_{\epsilon,\lambda}(\hat{X}) &:= \min_{w\in\mathbf{R}^n} \max_{X:\, \|X-\hat{X}\|_2\leq\epsilon} \|X^T w - y\|_2 + \lambda\|w\|_1 \\
&= \min_{w\in\mathbf{R}^n} \max_{\|\Delta\|_2\leq\epsilon} \|(\hat{X}+\Delta)^T w - y\|_2 + \lambda\|w\|_1
\end{aligned}
\tag{2}
$$

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

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

and the equality holds with the choice of Δ as

$$
\begin{aligned}
\Delta &:= \epsilon u v^T \\
u &:= \begin{cases} \frac{\hat{X}^T w - y}{\|\hat{X}^T w - y\|} & \text{if } \hat{X}^T w \neq y \\ \text{any unit-norm vector} & \text{otherwise} \end{cases} \\
v &:= \begin{cases} \frac{w}{\|w\|} & \text{if } w \neq 0 \\ \text{any unit-norm vector} & \text{otherwise} \end{cases}
\end{aligned}
$$

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

$$
\phi_{\epsilon,\lambda}(\hat{X}) = \min_{w\in\mathbf{R}^n} \left\|\hat{X}^T w - y\right\|_2 + \epsilon\|w\|_2 + \lambda\|w\|_1
\tag{3}
$$

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 $P \in \mathbf{R}^{n\times k}$, $Q \in \mathbf{R}^{m\times k}$, P and Q have full rank k with $k \ll \min\{m,n\}$. When the full data matrix $X \in \mathbf{R}^{n\times m}$ 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: $X_{\backslash i} \simeq PQ_{\backslash i}^T$ where $\backslash i$ 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:

$$\phi_{\lambda,\epsilon} =$$
$$\min_{w,z\in\mathbf{R}^n} \|Qz-y\|_2 + \epsilon\|w\|_2 + \lambda\|w\|_1 \; : \; z = P^T w$$
$$= \min_{w,z\in\mathbf{R}^n} \max_{u\in\mathbf{R}^k} \|Qz-y\|_2 + \epsilon\|w\|_2 + \lambda\|w\|_1$$
$$+ u^T(z - P^T w)$$
$$= \max_{u\in\mathbf{R}^k} \min_{w,z\in\mathbf{R}^n} \|Qz-y\|_2 + u^T z +$$
$$\epsilon\|w\|_2 + \lambda\|w\|_1 - (Pu)^T w$$
$$= \max_{u\in\mathbf{R}^k} f_1(u) + f_2(u),$$

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

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

$$\|Qz-y\|_2^2 = z^T Q^T Q z - 2y^T Q z + y^T y$$
$$= \bar{z}^T\bar{z} - 2c^T\bar{z} + y^T y$$
$$\text{where } \bar{z} := (Q^T Q)^{1/2}z \in \mathbf{R}^n$$
$$\text{and } c := (Q^T Q)^{-1/2}Q^T y \in \mathbf{R}^k$$
$$= \|\bar{z}-c\|_2^2 + y^T y - c^T c.$$

Note that $c^T c = y^T Q(Q^T Q)^{-1}Q^T y \le y^T y$ since $Q(Q^T Q)^{-1}Q^T \preceq I$ is the projection matrix onto $\mathcal{R}(Q)$. Letting $s := \sqrt{y^T y - c^T c} \ge 0$ gives

$$f_1(u) := \min_z \|Qz-y\|_2 + u^T z$$
$$= \min_z \sqrt{\|\bar{z}-c\|_2^2 + s^2} + \bar{u}^T\bar{z}$$
$$\text{by letting } \bar{u} := (Q^T Q)^{-1/2}u$$
$$= \bar{u}^T c + \min_x \sqrt{\|x\|_2^2 + s^2} - \bar{u}^T x$$
$$\text{by letting } x := c - \bar{z}.$$

Now consider the second term $\min_x \sqrt{\|x\|_2^2 + s^2} - \bar{u}^T x$. The optimal x^* must be in the direction of $\bar{u}$. Letting $x := \alpha\bar{u}, \alpha \in \mathbf{R}$, we have the expression

$$\min_{\alpha\in\mathbf{R}} \sqrt{\alpha^2\|\bar{u}\|_2^2 + s^2} - \alpha\|\bar{u}\|_2^2$$

When $\|\bar{u}\|_2 \ge 1$, the problem is unbounded below. When $\|\bar{u}\|_2 < 1$, the optimal solution is $\alpha^* = \frac{s}{\sqrt{1-\|\bar{u}\|_2^2}}$ and the optimal value is thus $s\sqrt{1 - \|\bar{u}\|_2^2}$. The closed-form expression for $f_1(u)$ is therefore:

$$f_1(u) = \bar{u}^T c + \min_x \sqrt{\|x\|_2^2 + s^2} - \bar{u}^T x$$
$$= \bar{u}^T c + s\sqrt{1 - \|\bar{u}\|_2^2}$$
$$= u^T(Q^T Q)^{-1/2}c + s\sqrt{1 - u^T(Q^T Q)^{-1}u}$$
$$= u^T K^{-1/2}c + s\sqrt{1 - u^T K^{-1}u}$$
$$\text{by letting } K := Q^T Q. \tag{4}$$

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

$$f_2(u) = \min_w \max_{v,r} r^T w + v^T w - (Pu)^T w$$
$$\text{s.t.} \quad \|r\|_2 \le \epsilon, \|v\|_\infty \le \lambda$$
$$= \max_{v,r} \min_w (r + v - Pu)^T w$$
$$\text{s.t.} \quad \|r\|_2 \le \epsilon, \|v\|_\infty \le \lambda$$
$$= \max_{v,r} 0$$
$$\text{s.t.} \quad \|r\|_2 \le \epsilon, \|v\|_\infty \le \lambda, Pu = v + r \tag{5}$$

Hence $f_2(u) = 0$ if there exists $v, r \in \mathbf{R}^n$ 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:

$$\phi_{\lambda,\epsilon} = \max_{\substack{u\in\mathbf{R}^k, \\ v,r\in\mathbf{R}^n}} u^T K^{-1/2}c + s\sqrt{1 - u^T K^{-1}u}$$
$$\text{s.t.} \quad \|v\|_\infty \le \lambda, \|r\|_2 \le \epsilon, Pu = v + r$$

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

$$\phi_{\lambda,\epsilon} = \max_{\substack{u\in\mathbf{R}^k, \\ v,r\in\mathbf{R}^n}} u^T c + s\sqrt{1 - u^T u}$$
$$\text{s.t.} \quad \|v\|_\infty \le \lambda, \|r\|_2 \le \epsilon, Ru = v + r$$
$$= \max_{u,v,r,t} u^T c + st$$
$$\text{s.t.} \quad \left\|\begin{bmatrix} u \\ t \end{bmatrix}\right\|_2 \le 1, \|v\|_\infty \le \lambda, \|r\|_2 \le \epsilon,$$
$$Ru = v + r \tag{6}$$

Bidual problem. The bidual of (3) writes

$$\phi_{\lambda,\epsilon} = \min_{w\in\mathbf{R}^n} \max_{u,v,r,t} u^T c + st + w^T(v + r - Ru)$$
$$\text{s.t.} \quad \left\|\begin{bmatrix} u \\ t \end{bmatrix}\right\|_2 \le 1, \|v\|_\infty \le \lambda, \|r\|_2 \le \epsilon$$
$$= \min_{w\in\mathbf{R}^n} \max_{u,v,r,t} \begin{bmatrix} c - R^T w \\ s \end{bmatrix}^T \begin{bmatrix} u \\ t \end{bmatrix} + w^T v + w^T r$$
$$\text{s.t.} \quad \left\|\begin{bmatrix} u \\ t \end{bmatrix}\right\|_2 \le 1, \|v\|_\infty \le \lambda, \|r\|_2 \le \epsilon$$

Therefore,

$$\phi_{\lambda,\epsilon} = \min_{w\in\mathbf{R}^n} \left\|\begin{bmatrix} c - R^T w \\ s \end{bmatrix}\right\|_2 + \epsilon\|w\|_2 + \lambda\|w\|_1 \tag{7}$$

where $R \in \mathbf{R}^{n\times k}$, $c \in \mathbf{R}^k$ and $s \in \mathbf{R}$. 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 $c := (Q^T Q)^{-1/2}Q^T y$, $s := \sqrt{y^T y - c^T c}$,

$R := P(Q^TQ)^{1/2}$, 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 $A := \begin{bmatrix} R^T \\ 0 \end{bmatrix} \in \mathbf{R}^{(k+1)\times n}$ and $b := \begin{bmatrix} c \\ s \end{bmatrix} \in \mathbf{R}^{k+1}$, problem (7) becomes:

$$\phi_{\lambda,\epsilon} = \min_{w\in\mathbf{R}^n} \|Aw - b\|_2 + \epsilon\|w\|_2 + \lambda\|w\|_1 \qquad (8)$$

Problem (8) is equivalent to:

$$
\begin{aligned}
\phi_{\lambda,\epsilon} &= \min_{w\in\mathbf{R}^n} \max_{\alpha,\beta,\gamma} \alpha^T(Aw - b) + \beta^T w + \gamma^T w \\
&\quad \text{s.t.} \quad \|\alpha\|_2 \le 1,\ \|\beta\|_2 \le \epsilon,\ \|\gamma\|_\infty \le \lambda \\
&= \max_{\substack{\|\alpha\|_2\le 1 \\ \|\beta\|_2\le\epsilon \\ \|\gamma\|_\infty\le\lambda}} \min_{w\in\mathbf{R}^n} w^T(A^T\alpha + \beta + \gamma) - \alpha^T b \\
&= \max_{\alpha,\beta,\gamma} -\alpha^T b \\
&\quad \text{s.t.} \quad \|\alpha\|_2 \le 1,\ \|\beta\|_2 \le \epsilon,\ \|\gamma\|_\infty \le \lambda, \\
&\qquad\quad A^T\alpha + \beta + \gamma = 0 \\
&= \max_{\alpha,\beta,\gamma} -\alpha^T b \\
&\quad \text{s.t.} \quad \|\alpha\|_2 \le 1,\ \|\beta\|_2 \le \epsilon, \\
&\qquad\quad |a_i^T\alpha + \beta_i| \le \lambda, \forall i = 1\dots n
\end{aligned}
$$

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

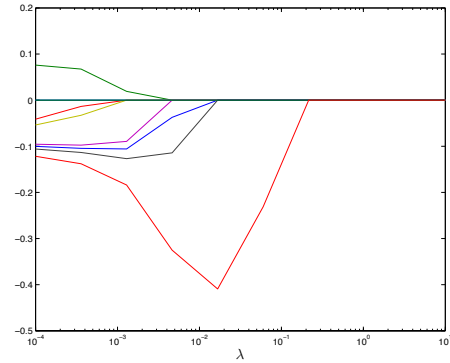## 4   Non-robust square-root LASSO

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:

$$\min_{w\in\mathbf{R}^n} \|QP^Tw - b\|_2 + \lambda\|w\|_1 \qquad (9)$$

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 λ does not provide any
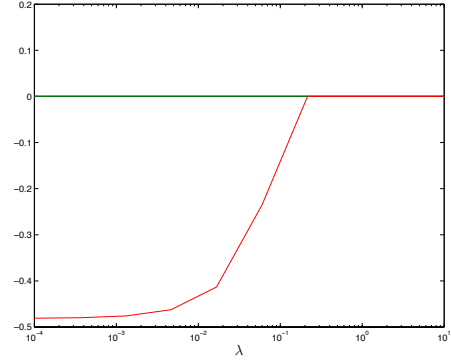
(a) Non-robust rank-1 square-root LASSO.

(b) Robust rank-1 square-root LASSO.

Figure 2:   Non-robust versus Robust square-root LASSO under rank-1 approximated data. The y-axis shows non-zero components of the optimal solution as λ varies.

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 $P \in \mathbf{R}^{n\times k}$ and $Q \in \mathbf{R}^{m\times k}$ 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

$$
\begin{aligned}
&\min_{w\in\mathbf{R}^n} \|Q(P^Tw - z) - u\|_2 + \lambda\|w\|_1 \\
={} &\min_{w\in\mathbf{R}^n} \sqrt{\|Q(P^Tw - z)\|_2^2 + \|u\|_2^2} + \lambda\|w\|_1
\end{aligned}
$$

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

$$\min_{w\in\mathbf{R}^n} \|w\|_1 \ : \ P^Tw = P^Tw_0$$

has an optimal solution with cardinality at most k. We prove this in the following lemma:

Lemma 1 *The problem*

$$\min_{x \in \mathbf{R}^n} \|x\|_1 \ : \ Ax = b \qquad (10)$$

with $A \in \mathbf{R}^{k \times n}$ 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 $x \in \mathbf{R}^n$ be an optimal solution to (10). Without loss of generality, we can assume all components of x_i are non-zeros (if some components are zeros one can discard the corresponding columns of A).

If $\mathbf{card}(x) > k$, we provide a procedure to reduce the cardinality of x, while keeping the same l_1-norm and constraint feasibility. Let $s \in \mathbf{R}^n$ be the (unique) sub-gradient of $\|x\|_1$: $s_i := \text{sign}(x_i), i = 1, \ldots, n$. The optimality condition of (10) shows that $\exists \mu \in \mathbf{R}^k$ such that $A^T \mu = s$. Since all the columns A_i's are linearly dependent, there exist i and c_j such that

$$\begin{aligned} A_i &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j A_j, \text{ where } \mathcal{E} := \{1, \ldots, n\} \\ A_i^T \mu &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j A_j^T \mu \\ s_i \mu &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j s_j \end{aligned}$$

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

$$\begin{aligned} s_i A_i &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j s_i A_j \\ &= \sum_{j \in \mathcal{E} \setminus \{i\}} c_j s_j s_i s_j A_j : \text{ since } s_j^2 = 1 \\ &= \sum_{j \in \mathcal{E} \setminus \{i\}} d_j s_j A_j \end{aligned}$$

Let us define a direction vector $\theta \in \mathbf{R}^n$ as follows: $\theta_i := -s_i$ and $\theta_j := d_j s_j, j \in \mathcal{E} \setminus \{i\}$. 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:

$$\begin{aligned} \|x^{(\rho)}\|_1 &= |x_i^{(\rho)}| + \sum_{j \in \mathcal{E} \setminus \{i\}} |x_j^{(\rho)}| \\ &= (|x_i| - \rho) + \sum_{j \in \mathcal{E} \setminus \{i\}} (x_j^{(\rho)} + \rho d_j) \\ &= \|x\|_1 + \rho \left(\sum_{j \in \mathcal{E} \setminus \{i\}} \rho d_j - 1 \right) \\ &= \|x\|_1 \end{aligned}$$

Choosing $\rho := \min\{t \geq 0 : x_j + t\theta_j = 0 \text{ for some j}\}$ we have one fewer non-zeros components in $x^{(\rho)}$. Note that $\rho \leq |x_i|$. 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 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):

$$\min_{x \in \mathbf{R}^n} \|Ax - b\|_2 + \epsilon\|x\|_2 + \lambda\|x\|_1$$

where $A \in \mathbf{R}^{k \times n}$, $b \in \mathbf{R}^k$, and the optimization variable is now $x \in \mathbf{R}^n$. 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 3×10^5 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:

$$\min_{x \in \mathbf{R}^n} \|Ax - b\|_2 = \min_{x \in \mathbf{R}^n, s \in \mathbf{R}} s \ : \ \|Ax - b\|_2 \leq s$$

The log-barrier function is $\varphi_\gamma(x, s) = \gamma s - \log\left(s^2 - \|Ax - b\|_2^2\right)$. The cost is from evaluating the inverse Hessian of

$$f := -\log\left(s^2 - (Ax - b)^T(Ax - b)\right) \qquad (11)$$

Let $g := -\log\left(s^2 - w^T w\right)$, we have $\nabla g = \frac{2}{-g}\begin{bmatrix} -w \\ s \end{bmatrix}$ and $\nabla^2 g = \frac{2}{-g}\begin{bmatrix} -I & 0 \\ 0 & 1 \end{bmatrix} + \nabla g(\nabla g)^T$. 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 $\begin{bmatrix} Ax - b \\ s \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ s \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} = \tilde{A}\tilde{x} - \tilde{b}$ where $\tilde{A} \in \mathbf{R}^{(k+1) \times n}$ and:

$$\begin{aligned} &\nabla^2 f \\ &= \tilde{A}^T \left(\frac{2}{-f}\begin{bmatrix} -I & 0 \\ 0 & 1 \end{bmatrix} + \frac{4}{f^2}\begin{bmatrix} -Ax + b \\ s \end{bmatrix}\begin{bmatrix} -Ax + b \\ s \end{bmatrix}^T \right) \tilde{A} \\ &= \frac{2}{-f}\tilde{A}^T\begin{bmatrix} -I & 0 \\ 0 & 1 \end{bmatrix}\tilde{A} \\ &\quad + \frac{4}{f^2}\left(\tilde{A}^T\begin{bmatrix} -(Ax - b) \\ s \end{bmatrix} \right)\left(\tilde{A}^T\begin{bmatrix} -(Ax - b) \\ s \end{bmatrix} \right)^T \end{aligned}$$

$$(12)$$

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

$$
\begin{aligned}
\phi &= \min_{w \in \mathbf{R}^n} \|Ax - b\|_2 + \epsilon\|x\|_2 + \lambda\|x\|_1 \\
&= \min_{\substack{p,q \in \mathbf{R}^n, \\ s,t \in \mathbf{R}}} s + \epsilon t + \lambda\left(\mathbf{1}^T p + \mathbf{1}^T q\right) \\
&\quad \text{s.t.} \quad \|p - q\|_2 \leq t, \ \|A(p-q) - b\|_2 \leq s, \\
&\qquad\quad p \geq 0, q \geq 0
\end{aligned}
$$

The log-barrier function is thus

$$
\begin{aligned}
\varphi_\gamma(p,q,s,t) = \ & \gamma\left(s + \epsilon t + \lambda\left(\mathbf{1}^T p + \mathbf{1}^T q\right)\right) \\
& - \log\left(t^2 - \|p - q\|_2^2\right) \\
& - \log\left(s^2 - \|A(p-q) - b\|_2^2\right) \\
& - \sum_{i=1}^n \log(p_i) - \sum_{i=1}^n \log(q_i) \\
& - \log(s) - \log(t).
\end{aligned}
$$

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

$$
\nabla l_1 = \frac{2}{-l_1}\left[p_1 - q_1, q_1 - p_1, \cdots p_n - q_n, q_n - p_n, t\right]^T
$$

$$
\nabla^2 l_1 = \frac{2}{-l_1}\begin{bmatrix} B & & & \\ & \ddots & & \\ & & B & \\ & & & 1 \end{bmatrix} + \nabla l_1 (\nabla l_1)^T
$$

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 $\tilde{x} = \left[p_1, q_1, \cdots p_n, q_n, s\right]^T$, we have

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

where $\tilde{A} \in \mathbf{R}^{(k+1) \times (2n+1)}$. 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)).

Table 1: Data sets used in numerical experiments.

Data set	#train	#test	#features	Type
Gisette	6,000	1,000	5,000	dense
20 Newsgroups	15,935	3,993	62,061	sparse
RCV1.binary	20,242	677,399	47,236	sparse
SIAM 2007	21,519	7,077	30,438	sparse
Real-sim	72,309	N/A	20,958	sparse
NIPS papers	1,500	N/A	12,419	sparse
NYTimes	300,000	N/A	102,660	sparse
Random 1	500	N/A	100	dense
Random 2	625	N/A	125	dense
Random 3	750	N/A	150	dense
. . .	. . .	. . .	. . .	. . .
Random 19	2750	N/A	550	dense
Random 20	2875	N/A	575	dense
Random 21	3000	N/A	600	dense

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 [1] 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 8GB 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.

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 theorical 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-

---

[1] All data sets are available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.
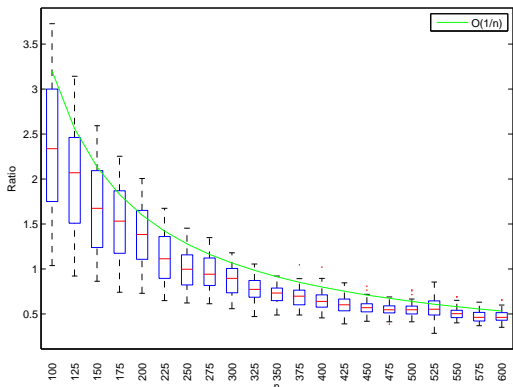
Figure 3: The ratio between the running time of our robust model and the original model.

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

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

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.
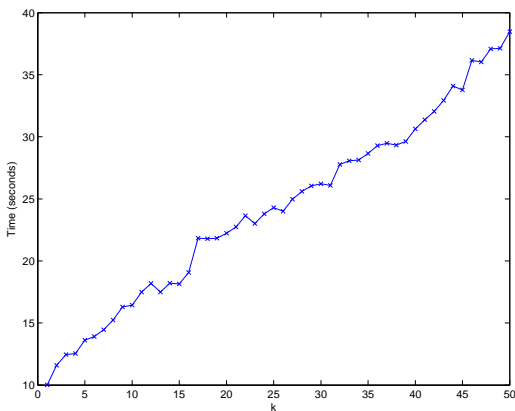

Figure 4: 5-fold cross validation on Gisette data with robust approximation model.

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.
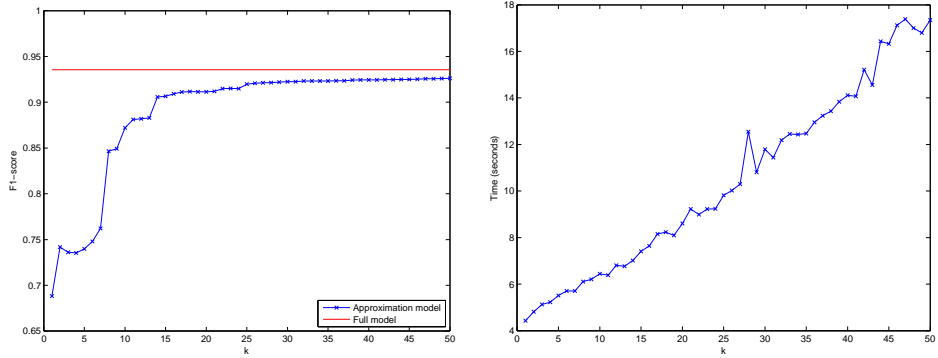
(a) Performance on binary classification.

(b) Training time of our model. The time to train the full model is 5547.1 seconds.

Figure 5: Classification performance and running time on RCV1 data set.
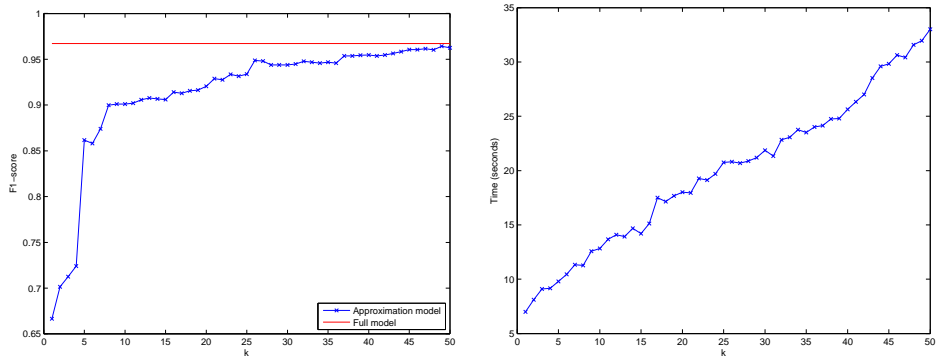
(a) Performance on binary classification.

(b) Training time of our model. The time to train the full model is 991 seconds.

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

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

Query	LEARNING	STATISTIC	OPTIMIZATION	TEXT	VISION
Time (s)	3.15	2.89	3.11	3.52	3.15
1	error	data	algorithm	trained	object
2	action	distribution	data	error	image
3	algorithm	model	distribution	generalization	visiting
4	targeting	error	likelihood	wooter	images
5	weighed	algorithm	variable	classifier	unit
6	trained	parameter	network	student	recognition
7	uniqueness	trained	mixture	validating	representation
8	reinforced	likelihood	parame	trainable	motion
9	control	gaussian	bound	hidden	view
10	policies	set	bayesian	hmm	field

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

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

References

E. D. Andersen and K. D. Andersen. The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm. pages 197–232, 2000.

E. D. Andersen, C. Roos, and T. Terlaky. On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2):249–277, 2003.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21 (2):572–596, 2011.

P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *The Journal of Machine Learning Research*, 6:2153–2175, 2005.

P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4): 1035–1064, 1997.

L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *Pacific Journal of Optimization*, 8(4):667–698, 2012.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

P. Garrigues and L. E. Ghaoui. An homotopy algorithm for the lasso with online observations. In *Advances in neural information processing systems*, pages 489–496, 2009.

I. Guyon, S. R. Gunn, A. Ben-Hur, and G. Dror. Result analysis of the nips 2003 feature selection challenge. In *NIPS*, volume 4, pages 545–552, 2004.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, May 2011.

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale l 1-regularized least squares. *Selected Topics in Signal Processing, IEEE Journal of*, 1(4):606–617, 2007.

K. Koh, S.-J. Kim, and S. P. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *Journal of Machine learning research*, 8(8): 1519–1555, 2007.

E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588. ACM, 2013.

M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends® in Machine Learning*, 3(2):123–224, 2011.

L. Miranian and M. Gu. Strong rank revealing lu factorizations. *Linear Algebra and its Applications*, 367 (0):1 – 16, 2003.

M. Y. Park and T. Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

M. Soltanolkotabi, E. Elhamifar, E. J. Candes, et al. Robust subspace clustering. *The Annals of Statistics*, 42(2):669–699, 2014.

R. J. Tibshirani et al. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7:1456–1490, 2013.

C.-H. Tsai, C.-Y. Lin, and C.-J. Lin. Incremental and decremental training for linear classification. 2014.

E. A. Yildirim and S. J. Wright. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12(3):782–810, 2002.

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.