# Robust Optimization and Data Approximation in Machine Learning

*Gia Vinh Anh Pham*

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-216
http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-216.html

December 1, 2015

# Robust Optimization and Data Approximation in Machine Learning

by

Gia Vinh Anh Pham

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy
in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Robust Optimization and Data Approximation in Machine Learning

# Abstract

Robust Optimization and Data Approximation in Machine Learning

by

Gia Vinh Anh Pham

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Laurent El Ghaoui, Chair

Modern learning problems in nature language processing, computer vision, computational biology, etc. often involve large-scale datasets with millions of samples and/or millions of features, thus are challenging to solve. Simply replacing the original data with a simpler approximation such as a sparse matrix or a low-rank matrix does allow a dramatic reduction in the computational effort required to solve the problem, however some information of the original data will be lost during the approximation process. In some cases, the solution obtained by directly solving the learning problem with approximated data might be infeasible for the original problem or might have undesired properties. In this thesis, we present a new approach that utilizes data approximation techniques and takes into account, via robust optimization the error made during the approximation process in order to obtain learning algorithms that could solve large-scale learning problems efficiently while preserving the learning quality.

In the first part of this thesis, we give a brief review of robust optimization and its appearance in machine learning literature. In the second part of this thesis, we examine two data approximation techniques, namely data thresholding and low-rank approximation, and then discuss their connection to robust optimization.

Professor Laurent El Ghaoui
Dissertation Committee Chair

Dedicated to my parents, Cuong Pham & Nguyet Nguyen, and my wife Dan-Quynh Tran for their continuous support and encouragement throughout my life.

# Contents

# List of Figures

# List of Tables

## Acknowledgements

First, I would like to thank my parents, my wife and my brother for their unconditional love and constant support. They have been the source of motivation for me to get through the ups and downs of graduate student's life.

With profound gratitude, I want to thank my advisor, Professor Laurent El Ghaoui for tolerating my ignorance during my time at Berkeley and encouraging me when I was at my most difficult moments. I have grown significantly since I met Laurent, both personally and academically, and I am extremely grateful for his guidance.

I would like to thank Professor Bin Yu and Professor Ming Gu for being on my thesis committee and for their valuable feedbacks and suggestions during my qualification exam.

Finally I want to thank my colleagues and friends at Berkeley, who have provided countless technical, intellectual, and personal help with various aspects of my work. These include, but are not limited to, Tarek Rabbani, Youwei Zhang, Andrew Godbehere, Mert Pilanci and other members past and present of StatNews group for their help and support.

# Chapter 1

# Introduction

Modern supervised learning algorithms such as support vector machine or logistic regression have become popular tools and widely used in many areas: natural language processing, computer vision, data mining, bioinformatics, etc.. In most of real-world learning problems, for instance text classification, the number of features can be very large - hundreds of thousands or millions, so is the number of training samples; thus, we have to face the daunting task of storing and operating on matrices with thousands to millions rows and columns. It is expensive to store the large-scale data in the memory and in many cases the data cannot even be loaded into the memory. More importantly, these learning algorithms often require us to solve a large convex optimization problem in which under large-scale setting, might be impossible at times.

In many machine learning applications, the problem data involved is closed to one having a simple structure (we will see in more details in Chapter 3 and 4), for example the input data could be closed to a sparse matrix or a low-rank matrix, etc.. If the data matrix has exactly simple structure then it is often possible to exploit that structure to decrease the computational time and the memory needed to solve the problem. What if, instead, the data matrix is only approximately simple, and we have at our disposal the simple approximation?

In this dissertation, we will introduce a new approach that uses data approximation under robust optimization scheme to deal with large-scale learning problems. The main idea is that instead of ignoring the error made by approximating the original data with a simpler structure, we consider such error as some sort of artificially introduced *data uncertainty* [1] and then use robust optimization to obtain modified learning algorithms that are capable of handling the approximation error even in worst case scenario. We show that the savings obtained using data approximation still hold for the *robust counterpart problem*[1]. In this sense, our approach takes advantage of the simpler problem structure on the computational level; in addition it allows to control the noise. The proposed approach has both advantage of having memory saving and computational speedups as well as having reliable performance.

The thesis is organized as follows: we first describe robust optimization and its important concepts such as data uncertainty and robust counterpart problem in Chapter 2. We will also briefly review robust optimization in machine learning literature in Chapter 2. We then introduce data thresholding technique for large-scale sparse linear classification in Chapter 3. In Chapter 4, we will introduce an efficient and scalable robust low-rank model for LASSO problem. We draw conclusions and point out some future research directions in Chapter 5.

[1]this concept will be described in details in Chapter 2

# Chapter 2

# Robust Optimization

## 2.1 General optimization problem

In general mathematical optimization, we often assume that the input data is precisely known, and we seek to optimize an objective function over a set of decision variables as following:

$$\begin{aligned} \min_{x} \quad & f_0(x, u_0) \\ \text{subject to} \quad & f_i(x, u_i) \leq 0, i = 1, \ldots, m \end{aligned} \tag{2.1.1}$$

where $x \in \mathbf{R}^d$ is a vector of decision variables and $\{u_i\}_{i=1}^m$ are **known** input parameters for functions $\{f_i\}_{i=1}^m$.

Many machine learning problems require us to solve an optimization problem - usually a convex optimization problem of the form (2.1.1). Following are some of such problems in which we will look at in the latter chapters of this thesis:

(a) **Linear classification:** given a set of $n$ training examples $\{(x_i, y_i)\}_{i=1}^n$ where each input $x_i \in \mathbf{R}^d$ is a $d$-dimensional vector and the output $y_i \in \{-1, +1\}$ is a binary variable, the two-class linear classification problem aims to find a classification rule from training data so that given new input $x \in \mathbf{R}^d$, we can "optimally" assign a class label $y \in \{-1, +1\}$ to $x$, for example $y = \mathbf{sign}(w^T x + b)$ for some weight vector $w \in \mathbf{R}^d$ and scalar $b \in \mathbf{R}$. In general, the linear classification problem seeks for the optimal solution of the following optimization problem:

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{n} \sum_{i=1}^{n} f\left(y_i(w^T x_i + b)\right) + R(w)$$

where $f : \mathbf{R} \to \mathbf{R}_+$ is a non-negative convex function, so called the loss function and $R : \mathbf{R}^d \to \mathbf{R}_+$ is the regularization function.

- In case of **Support Vector Machines (SVM)**, the loss functions is hinge loss: $f(z) = [1 - z]_+$ (we define $[z]_+ := \max(0, z)$)

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{n} \sum_{i=1}^{n} [1 - (y_i(w^T x_i + b)]_+ + R(w)$$

- For **logistic regression**, the loss function is logistic loss: $f(z) = \log(1 + e^{-z})$.

$$\min_{w \in \mathbf{R}^d, b \in \mathbf{R}} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp(-y_i(w^T x_i + b))\right) + R(w)$$

The regularization term is widely used to seeks a trade-off between fitting the observations and reducing the model complexity, for example: $l_1$-regularization $R(w) = \lambda \|w\|_1$, or $l_2$-regularization $R(w) = \lambda \|w\|_2$.

(b) **Least squares regression and its variations:** given a response vector $y \in \mathbf{R}^n$ and data matrix $X \in \mathbf{R}^{d \times n}$ (each column of $X$ corresponds to a data sample in $\mathbf{R}^d$), the least squares regression problem is to find a weight vector $w \in \mathbf{R}^d$ so that the $l_2$ norm of the residual $y - X^T w$ is minimized, i.e. solving the following optimization problem:

$$\min_{w} \|y - X^T w\|_2$$

Since minimizing the squared error can lead to sensitive solutions (Golub and Van Loan (1996)), variations of least squares problem that use regularization methods such as

Tikhonov regularization (Tikhonov and Arsenin (1977)) - using $l_2$ norm regularization, and LASSO - *Least Absolute Shrinkage and Selection Operator* (Tibshirani (1996); Efron et al. (2004)) - using $l_1$ norm regularization. Both methods provide some level of regularity to the solution. LASSO is also known for selecting sparse solutions, i.e. having few entries that are non-zero, and therefore identifying the features on the data matrix $X$ that are useful to predict the response vector $y$.

(c) **More general supervised learning problems:** linear classification and regression problems, amongst many others, can be generalized to solving the following optimization problem:

$$\min_{v\in\mathbf{R},w\in\mathcal{C}} f(A^T w + cv) \tag{2.1.2}$$

where the loss function $f : \mathbf{R}^n \to \mathbf{R}_+$ is convex, the data matrix $A \in \mathbf{R}^{d\times n}$ and vector $c \in \mathbf{R}^n$ are given, $\mathcal{C} \subseteq \mathbf{R}^d$ is a convex set that constraining the decision variable $w$. This formulation covers well-known problems, for example:

- By choosing $f(z) := \sum_{i=1}^n (1 - z_i)_+$, $A := [y_1x_1, \ldots, y_nx_n], c := y, v := b$ and the constraint set $\mathcal{C} := \{w : \|w\|_2 \leq \lambda\}$ we obtained the (constrained) $l_2$-SVM problem:

$$\min_{\|\beta\|_2\leq\lambda} \sum_{i=1}^n [1 - y_i(x_i^T w + b)]_+$$

- By choosing $f(z) := \|y - z\|_2$, $c := 0$, $A := [x_1, \ldots, x_n]$ and the constrain set $\mathcal{C} := \{w \in \mathbf{R}^d : \|w\|_1 \leq \lambda\}$ we obtained the LASSO problem:

$$\min_{\|w\|_1\leq\lambda} \|y - X^T w\|_2$$

## 2.2 Robust optimization and Data uncertainty

When a real-world application is formulated as a mathematical optimization problem, more often than not the problem data is not precisely known, but is subject to some sort of uncertainty due to many factors - for example:

- *prediction errors*: occur when we do not have the data at the time the problem is solved and we need to predict their values.
- *measurement errors*: occur when we cannot record the data exactly due to limitation of measurement devices.

The data uncertainty can affect either the *feasibility* of a solution to the optimization problem, or it could affect the *optimality* of a solution, or both. Robust optimization provides a novel and systematic approach to deal with problems involving data uncertainty by generating a robust solution that is immunized against the effect of data uncertainty.

When data uncertainty affects the *feasibility* of a solution (i.e. the constraints), robust optimization seeks to obtain a solution that is feasible for any possible realization of the unknown parameters; and when data uncertainty affects the *optimality* of a solution (i.e. the objective function), robust optimization seeks to obtain a solution that optimizes the worst case scenario. In particular, suppose that the parameters $\mathbf{u} = (u_0, \dots, u_m)$ varies in some **uncertainty set** $\mathcal{U}$ that characterizes the unknown parameters $\mathbf{u}$, the **robust counterpart** problem solves:

$$\begin{array}{ll} \min\limits_{x} & \max\limits_{\mathbf{u}\in\mathcal{U}} f_0(x, u_0) \\ \text{subject to} & \forall \mathbf{u} \in \mathcal{U} : f_i(x, u_i) \leq 0, i = 1, \dots, m. \end{array} \tag{2.2.1}$$

The goal of (2.2.1) is to compute the optimal solution that minimizes the worst case objective function amongst all those solutions which are feasible for all realizations of the disturbances that affects the parameters $\mathbf{u}$ within the uncertainty set $\mathcal{U}$.

Being a conservative (worst-case oriented) methodology, robust optimization is often useful when the optimization problems have hard constraints that must be satisfied no matter what, or the objective function/optimal solutions are highly sensitive to perturbations, or we cannot afford low probability high-magnitude risks (for example in aerospace design, nuclear plant design, etc.).

### 2.2.1 Robust optimization vs. Sensitivity analysis

Sensitivity analysis is another traditional method of dealing with data uncertainty. However, unlike robust optimization that seeks for an uncertainty-immunized solution under worst-case scenario to an optimization problem involving uncertainty data, sensitivity analysis aims at analyzing the goodness of a solution when there are small perturbations in the underlying problem data. It first solves the problem with fixed values of the parameters and then see how the optimal solution is affected by small perturbations. Such post-mortem analysis is not helpful for computing the solutions that are robust to data perturbation.

### 2.2.2 Robust optimization vs. Stochastic optimization

In stochastic optimization, the data uncertainty are assumed to be random and the feasibility of a solution is expressed using chance constraints. Suppose that we know in advance the distribution of the input parameters, the corresponding stochastic optimization is:

$$\begin{array}{ll} \min\limits_{x} & t \\ \text{subject to} & \mathbb{P}_{u_0 \sim P_0}\left[f_0(x, u_0) \leq t\right] \geq 1 - \epsilon_0, \\ \text{and} & \mathbb{P}_{u_i \sim P_i}\left[f_i(x, u_i) \leq 0\right] \geq 1 - \epsilon_i, i = 1, \ldots, m \end{array} \tag{2.2.2}$$

where $\epsilon_i \ll 1, i = 0, \ldots, m$ are given tolerances and $P_i$ is the distribution of the input parameter $u_i, i = 0, \ldots, m$.

Although the above model (2.2.2) is powerful, there are some associated fundamental difficulties. First of all, it is difficult to obtain the actual distributions of the uncertain input parameters, and sometimes it is hard to interpret these distributions as well. Furthermore, even if we can obtain the probabilistic distributions, it is still computationally challenging to check the feasibility of the chance constraints, even more challenging to solve the optimization problem. In many cases, the chance constraints can destroy convexity and make the stochastic optimization problem become computationally intractable.

### 2.2.3 Example: linear programming problem

One could model the uncertainty set as flexibly as he/she wants, either using *probablistic* data model or *uncertainty-but-bounded* data model. To illustrate the concept of robust optimization and data uncertainty, we consider a simple linear programming problem:

$$\begin{array}{ll} \min\limits_{x \in \mathbf{R}^d} & c^T x \\ \text{subject to} & a_i x \leq b_i, i = 1, \ldots, n \end{array}$$

Below are some examples of how we could model data uncertainty and how we could derive the robust counterpart problems:

(a) **data uncertainty in objective function**

Suppose that the cost vector $c \in \mathbf{R}^d$ is uncertain but lies in a ball centered at a known vector $\bar{c} \in \mathbf{R}^d$ and radius $R > 0$: $B(\bar{c}, \lambda) := \{c : \|c - \bar{c}\|_2 \le R\}$, the robust counterpart problem writes:

$$\begin{array}{ll} \min\limits_{x} & \max\limits_{\|c-\bar{c}\|_2 \le R} c^T x \\ \text{subject to} & a_i x \le b_i, i = 1, \dots, n \end{array}$$

Since $\max\limits_{\|c-\bar{c}\|_2 \le R} c^T x = \bar{c}^T x + \max\limits_{\|z\|_2 \le R} z^T x = \bar{c}^T x + R\|x\|_2$, the robust counterpart problem is thus:

$$\begin{array}{ll} \min\limits_{x} & \bar{c}^T x + R\|x\|_2 \\ \text{subject to} & a_i x \le b_i, i = 1, \dots, n \end{array}$$

(b) **data uncertainty in constraints - decoupled**

Suppose that the data uncertainty occurs in the constraints, and the uncertainty for vectors $a_i$ are decoupled, each vector $a_i$ could be perturbed in a $l_2$ fixed ball with a known center and radius : $\mathcal{U}_i := \{a_i : a_i = \bar{a}_i + \delta_i, \|\delta_i\|_2 \le R_i\}, i = 1, \dots, n$, the robust counterpart problem writes:

$$\begin{array}{ll} \min\limits_{x} & c^T x \\ \text{subject to} & (\bar{a}_i + \delta_i)x \le b_i, \forall \delta_i : \|\delta_i\|_2 \le R_i, i = 1, \dots, n \end{array}$$

Since $(\bar{a}_i + \delta_i)^T x \le b_i, \forall \delta_i : \|\delta_i\|_2 \le R_i$ if and only if $\max_{\|\delta_i\|_2 \le R_i} (\bar{a}_i + \delta_i)^T x \le b_i$ , which is equivalent to $\bar{a}_i^T x + R_i\|x\|_2 \le b_i$, the robust counterpart problem is thus:

$$\begin{array}{ll} \min\limits_{x} & c^T x \\ \text{subject to} & \bar{a}_i x + R_i\|x\|_2 \le b_i, i = 1, \dots, n \end{array}$$

(c) **data uncertainty in constraints - coupled**

Suppose that the data uncertainty occurs in the constraints, and the uncertainty for vectors $a_i$ are coupled, given as: $\mathcal{U} := \{A : A = \bar{A} + \Delta, \|\Delta\|_F \le R\}$ where $A \in \mathbf{R}^{n \times d}$

is the matrix with rows $\{a_i\}_{i=1}^n$. The robust counterpart problem writes:

$$\begin{aligned}\min_x \quad & c^T x \\ \text{subject to} \quad & (\bar{A}+\Delta)x \le b, \forall \Delta \in \mathbf{R}^{n\times d} : \|\Delta\|_F \le R\end{aligned}$$

Since $(\bar{A}+\Delta)x \le b, \forall \Delta \in \mathbf{R}^{n\times d} : \|\Delta\|_F \le R$ if and only if $\max_{\|\Delta\|_F\le R}(\bar{A}+\Delta)x \le b$, which is equivalent to $\bar{A}x + R\|x\|_2\mathbf{1} \le b$, the robust counterpart problem is thus:

$$\begin{aligned}\min_x \quad & c^T x \\ \text{subject to} \quad & \bar{A}x + R\|x\|_2\mathbf{1} \le b\end{aligned}$$

(d) **probabilistic data uncertainty model**

For simplicity, we consider the case that $n = 1$, i.e. $A := a^T \in \mathbf{R}^{d\times 1}, b \in \mathbf{R}$. Suppose that our data $(a, b)$ varies in the uncertainty set

$$\mathcal{U} := \left\{ [a_\delta, b_\delta] = [a^{(0)}, b^{(0)}] + \sum_{k=1}^K \delta_k [a^{(k)}, b^{(k)}] : \delta \sim \mathbf{P} \right\}$$

where $a^{(k)}, b^{(k)}, k = 0, \ldots, K$ are known and the distribution $\mathbf{P}$ is also known. The robust counterpart problem writes:

$$\begin{aligned}\min_x \quad & c^T x \\ \text{subject to} \quad & \mathbb{P}_{\delta\sim\mathbf{P}}\left[a_\delta^T x \le b_\delta\right] \ge 1-\epsilon\end{aligned}$$

Dealing with such probabilistic model is difficult, very often the problem is severely computationally intractable. In few cases, for example when $\mathbf{P}$ is Gaussian distribution, i.e. $\delta \sim \mathcal{N}(\mu, \Sigma)$ and $\epsilon < 1/2$, the problem is tractable. Indeed, we observe that

$$z_x = a_\delta^T x - b_\delta = [a^{(0)}]^T x + \sum_{k=1}^K \delta_k [a^{(k)}]^T x - b^{(0)} - \sum_{k=1}^K \delta_k b^{(k)}$$

is a Gaussian random variable with the expectation $\mathbb{E}[z_x] = \alpha^T[x;1]$ and variable $var[z_x] = [x;1]^T Q^T Q [x;1]$ where $\alpha$ and $Q$ are vector and matrix that can be computed from the data $\{[a^{(k)}, b^{(k)}]\}, \mu, \Sigma$. Via some derivations, noting that $\epsilon \in [0, 1/2]$, we can reduce the probabilistic constraint is $\mathbb{P}[z_x \leq 0] \geq 1 - \epsilon$ to the second-order cone constraint $\alpha^T[x;1] + C_\epsilon \|Q[x;1]\|_2 \leq 0$ for some constant $C_\epsilon$. The robust counterpart problem is thus:

$$\begin{array}{ll} \min\limits_{x} & c^T x \\ \text{subject to} & \alpha^T[x;1] + C_\epsilon \|Q[x;1]\|_2 \leq 0 \end{array}$$

Due to the complexity and intractability of probabilistic data uncertainty model, in this thesis we will only focus on uncertain-but-bounded models.

## 2.3 Robust optimization in Machine learning

In recent years, there have been a number of works on the idea of using robust optimization to deal with data uncertainty that arises in many machine learning models. These works essentially consider learning problems in which training samples are assumed to be uncertain - but restricted in some bounded sets, and then proposes robustified learning algorithms that are modified versions of the nominal learning algorithms to guarantee immunity to data pertubations.

### 2.3.1 Data uncertainty models

Bhattacharyya (2004); Bhattachryya et al. (2003) investigated on linear classification problem (SVM) assuming an ellipsoidal model of uncertainty in which each data sample $x_i$ is uncertain but lies in a known ellipsoid $B(\bar{x}_i, \Sigma_i, \gamma_i) := \{x : (x - \bar{x}_i)^T \Sigma^{-1} (x - \bar{x}_i) \leq \gamma_i^2\}$. The corresponding robustified learning algorithm is then formulated as an SOCP (Second Order Cone Programming), which can be solved efficiently:

$$\begin{aligned}
\min_{w,b,\xi} \quad & \sum_{i=1}^{n} \xi_i \\
\text{subject to} \quad & y_i(w^T x_i + b) \geq -\xi_i + \gamma_i \|\Sigma_i^{1/2} w\|_2 \\
& \|w\|_2 \leq A \\
& \xi_i \geq 0, \forall i = 1, \dots, n
\end{aligned}$$

Trafalis and Gilbert (2007) considered $l_p$-norm uncertainty sets for SVM problem, in which each data sample $x_i$ is assumed to be in an $l_p$-ball $B_p(\bar{x}_i, \eta_i) := \{x : \|x - \bar{x}_i\|_p \leq \eta_i\}$. The resulting robust counterpart optimization problem can be transformed into an SOCP when $p = 2$ and or simply a linear programming problem when $p = 1$ or $p = \infty$.

Going further, Shivaswamy et al. (2006) considered linear classification and linear regression problems under probabilistic data uncertain model, assuming that each data sample $x_i$ is a random variable picked from a family of distributions which have a known mean and covariance. The authors also proposed SOCP formulations that robustify the standard learning algorithms and showed that these formulations outperform the imputation based classifiers and regression functions when applying to the missing variable problem.

Globerson and Roweis (2006) introduced a robust optiomization-inspired method for learning SVM classifiers under a worst case scenario of feature deletion at test time. In particular, the data uncertainty model assumes that up to $K$ features may be deleted/missing from each data vector and then compute the worst case objective function. The authors considered the following nominal problem (bias term is not introduced explicitly, and could be included by adding a constant feature):

$$\min_{w} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}[1 - y_i w^T x_i]_+$$

Under worst case scenario when up to $K$ features may be deleted from each data vector, the worst case hinge loss $[1 - y_o w^T x_i]_+$ of sample $i$ can be derived to be $[1 - y_i w^T x_i + s_i]_+$, where

$$\begin{aligned} s_i = \quad & \max_{\alpha_i \in \mathbf{R}^d} \quad y_i w^T (x_i \circ \alpha_i) \\ & \text{s.t.} \qquad 0 \leq \alpha_i \leq 1 \\ & \qquad\qquad \textstyle\sum_j \alpha_{ij} = K \end{aligned}$$

where $\circ$ denotes point-wise multiplication. The resulting robust counterpart problem can be written as the following quadratic problem and can be solved efficiently:

$$\begin{aligned} \min_{w,t,z,v} \quad & \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}[1 - y_i w^T x_i + t_i]_+ \\ \text{subject to} \quad & t_i \geq K z_i + \textstyle\sum_j v_{ij} \\ & v_i \geq 0 \\ & z_i + v_i \geq y_i x_i \circ w \end{aligned}$$

Data uncertainty is not only modeled for data points in the feature space but also for data labels. Caramanis and Mannor (2008) studied two types of adversarial disturbance in which the labels are corrupted: disturbance in the labels of a certain fraction of the data, and disturbance that affects the position of the data points. The authors proposed a general optimization-theoretic algorithmic framework for finding a classifier under these disturbances and provided distribution-dependent bounds on the amount of error as a function of the noise level for the two models.

A comprehensive study of other data uncertainty models for machine learning problems can also be found in (Ben-Tal et al. (2009)).

### 2.3.2 Connection to regularization

Robust optimization and regularization in machine learning are strongly related, and in some cases are equivalent. Early work by Ghaoui and Lebret (1997), Anthony and Bartlett

(1999) showed that structured robust solutions to least squares problems with uncertain data under Frobenius norm can be interpreted as a Tikhonov ($l_2$) regularization procedures:

**Theorem 1** *The following robust optimization formulation of the lease squares problem in which the data uncertainty set is described by* $\mathcal{U} = \{U \in \mathbf{R}^{d\times n} : \|U\|_F \leq \lambda\}$*:*

$$\min_{\beta\in\mathbf{R}^d} \max_{U\in\mathcal{U}} \|y - (X+U)^T\beta\|_2$$

*is equivalent to Tikonov-regularized regression problem:*

$$\min_{\beta\in\mathbf{R}^d} \|y - X\beta\|_2 + \lambda\|\beta\|_2$$

Recently, Xu et al. (2011) created a precise connection between robustness and regularized Support Vector Machine via notions of *sublinear aggregated uncertainty set*, which defined as follows:

**Definition 1** $U_0 \subseteq R^n$ *is called Atomic Uncertainty Set if*

*(i)* $0 \in U_0$

*(ii)* $\forall w_0 \in R^n : \sup\limits_{u\in U_0} w_0^T u = \sup\limits_{u'\in U_0} w_0^T u' < +\infty$

**Definition 2** $U \subseteq R^{n\times m}$ *is called Sublinear Aggregated Uncertainty Set of* $U_0$*, if* $U^- \subseteq U \subseteq U^+$ *where*

$$\begin{aligned} U^- &= \bigcup_{k=1}^{m} U_k^- \text{ where } U_k^- := \{(u_1,\dots,u_m) | u_k \in U_0; u_{i\neq k} = 0\} \\ U^+ &= \left\{(\alpha_1 u_1,\dots,\alpha_m u_m) | \sum_{i=1}^{m} \alpha_i = 1, \alpha_i \geq 0, u_i \in U_0, i = 1,\dots,m\right\} \end{aligned}$$

**Theorem 2** *Xu et al. (2011) Assume $\{x_i, y_i\}_{i=1}^n$ are non-separable and $U$ is a sublinear aggregated uncertainty set then the following problems are equivalent:*

$$(P1) \quad \min_{w,b} \max_{(u_1,\ldots,u_n)\in U} \left\{ R(w,b) + \sum_{i=1}^n [1 - y_i(w^T(x_i + u_i) + b)]_+ \right\}$$
$$(P2) \quad \min_{w,b} \left\{ R(w,b) + \sum_{i=1}^n [1 - y_i(w^T x_i + b)]_+ + \max_{u\in U_0} w^T u \right\}$$

An immediate corollary of Theorem 2 shows the equivalent of robust formulation and norm-regularized SVM:

**Corollary 3** *Assume $\{x_i, y_i\}_{i=1}^n$ are non-separable then the following problems are equivalent:*

$$(P1) \quad \min_{w,b} \max_{(u_1,\ldots,u_n):\sum_{i=1}^m \|u_i\|_* \le \lambda} \sum_{i=1}^n [1 - y_i(w^T(x_i + u_i) + b)]_+$$
$$(P2) \quad \lambda\|w\| + \min_{w,b} \sum_{i=1}^n [1 - y_i(w^T x_i + b)]_+$$

*where $\|.\|_*$ is dual-norm of $\|.\|$.*

## 2.4 Robust optimization and Data approximation

### 2.4.1 Data approximation

Modern supervised learning algorithms such as support vector machine (SVM) or logistic regression have become popular tools for classification problems. Many real-life problems, for instance text classification can be formulated as SVM or logistic regression problem. However, for such problems the number of features can be very large - hundreds of thousands, so is the number of training data, and solving such large scale problems often requires solving a large convex optimization problem which might be impossible at times. Moreover, it is also expensive to store the data matrix in the memory and sometimes the data cannot even be loaded into the memory.

Furthermore, we are often required to do classification/regression on a single dataset multiple times - one for each different response vector $y$. For example, problems involving multiple-label classification in which multiple target labels must be assigned to each sample and the *one-against-all* method is used (Tsoumakas and Katakis (2007)); or problems in which labels are columns of the data matrix (e.g. discovering word association in text datasets as in (Gawalt et al. (2010)). Algorithms that use the original feature matrix directly to solve the learning problem in such setting have extremely large memory space and CPU power requirements, and it is sometimes infeasible due to huge dimensions of the datasets and the number of learning instances need to be run.

One simple solution to deal with large-scale learning problems is to replace the data with a simpler structure. We could first simplify the data and then run learning algorithms to exploit cheap storage, efficient memory allocation. Following are some of data approximation techniques that are applicable in a large-scale setting:

- **Thresholding:** given a matrix $A \in \mathbf{R}^{n \times d}$ and a thresholding level, we want to approximate $A$ by $\hat{A} \in \mathbf{R}^{n \times d}$ such that $\hat{A}_{ij} = A_{ij}$ if $|A_{ij}| \geq t$ and $\hat{A}_{ij} = 0$ otherwise.
- **Low-rank approximation**: given a matrix $A \in \mathbf{R}^{n \times d}$, we want to approximate $A$ as product of two low-rank matrices $B \in \mathbf{R}^{n \times k}, C \in \mathbf{R}^{k \times d}$ such that the spectral norm/largest singular value norm error $\|A - BC\|_2$ is minimal.
- **Non-negative matrix factorization**: given a matrix $A \in \mathbf{R}_+^{n \times d}$, and pre-specified positive integer $r < \min(n, d)$, we want to approximate $A$ as product of two (possibly low-rank) non-negative matrix $W \in \mathbf{R}^{n \times r}$ and $H \in \mathbf{R}^{r \times d}$ such that the Frobenius norm of the error $\|A - WH\|_F$ is minimal.

Unlike other methods that also reduce the data complexity such as random projection or feature selection, the most advantageous property of data approximation approach is that it doesn't change the shape of the original data matrix - therefore we can also work with samples and features of the approximated data the same as with the original data matrix.

### 2.4.2 Is data approximation enough?

Simply replacing the original data with its approximation does allow a dramatic reduction in the computational effort required to solve the problem. However, we certainty lose some information about the original data. Further more, it does not guarantee that the corresponding solutions is feasible for the original problem. Indeed, consider a simple linear program:

$$\min_{x \in \mathbf{R}^2} c^T x : Ax \leq b, x \geq 0$$

where $c = \begin{pmatrix} -0.5 \\ 1 \end{pmatrix}$, $A = \begin{pmatrix} 1 & -1 \\ -1 & 0.001 \end{pmatrix}$, $b = \begin{pmatrix} 0 \\ -1 \end{pmatrix}$.

Solving this linear program we obtain the optimal solution $x^* = \begin{pmatrix} 1.001 \\ 1.001 \end{pmatrix}$. Suppose we approximate matrix $A$ by setting all entries with magnitude less than or equal to 0.001 to 0 (i.e. sparsifying the data with thresholding level 0.001), we obtain an approximation of $A$: $\bar{A} = \begin{pmatrix} 1 & -1 \\ -1 & 0 \end{pmatrix}$. Solving the above linear program with data matrix $\bar{A}$ we obtain the optimal solution $\bar{x}^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, and it is easy to see that $\bar{x}^*$ is not feasible for the constraint with original data matrix: $Ax \leq b$.

To deal with such issues, we proposed a new approach that takes into account, via robust optimization, the approximation error made during the data approximation process. We will use robust optimization principles and consider the approximation error as *artificially introduced* uncertainty on the original data. We will show that the simple structure of the approximated data can also be exploited in the robust counterpart. In fact, our empirical results show that the proposed method is able to achieve good performance, requires significantly less memory space and has drastically better running time complexity. We are also able to deal with huge datasets (with millions of training samples and hundreds of thousands features) on a personal workstation. In the next two chapters, we will examine two data approximation techniques, namely **data thresholding** (chapter 3), **low-rank approximation** (chapter 4).

# Chapter 3

# Data Thresholding

## 3.1 Introduction

In this section, we consider the linear classification problem presented in Chapter 2, in which we desire to solve the following optimization problem:

$$\min_{w \in \mathbf{R}^n, b \in \mathbf{R}} \frac{1}{m} \sum_{i=1}^{m} f\left(y_i(w^T x_i + b)\right) + \lambda \|w\|_1$$

where training samples $X = [x_1, \ldots, x_n] \in \mathbf{R}^{d \times n}, y \in \mathbf{R}^n$ are given, $f(t)$ is the loss function. As the closet convex relaxation of $l_0$-norm, $l_1$-norm regularization not only controls model complexity but also leads to sparse estimation, which provides both interpretable model coefficients and generalization ability.

Many efficient algorithms have been developed for $l_1$-penalized classification or regression problems: Friedman et al. (2010) proposed cyclical coordinate descent algorithm for solving generalized linear model problems that computes the whole solution path, thus help adaptive selection of tuning parameter; Genkin et al. (2007) and Balakrishnan and Madigan (2008) implemented a cyclic coordinate descent method called BBR which minimizes subproblems in a trust region and applies one-dimensional Newton steps to solve $l_1$-regularized LR; Koh et al. (2007) described an efficient interior-point method for solving large-scale $l_1$-regularized LR problems, Shi et al. (2010) introduced hybrid iterative shrinkage algorithm which comprised of a fixed point continuation phrase and an interior point phrase; Fung and Mangasarian (2004), Mangasarian (2006) proposed a fast Newton method for SVM problem, etc. However, the complexity of these algorithms, when it is known, grows fast with size of the dataset. There have been several strategies used in the literature to reduce the computational complexity of solving these problems: decomposition methods that work on subsets of training data - solving several smaller problems instead of the large one as in decomposition method, such as in Perkins et al. (2003), Osuna et al. (1997), Tseng and Yun (2008). The second strategies consists of parallelizing the learning algorithm. Approximation methods try to design less expensive and less complex algorithms that give an approximate solution without compromising the test set accuracy; for instance, Bordes and Bottou (2005) and Loosli et al. (2005) proposed an efficient online algorithm for solving SVM problems using selective sampling.

In this chapter, we propose and analyze a data thresholding technique for solving large scale sparse linear classification problems: sparsifying the dataset with appropriate thresholding level to obtain memory saving and speedups without losing much of the performance. The proposed method can be used on top of any learning algorithm as a preprocessing step. We will provide theoretical bound on the amount of thresholding that can take place in order to guarantee sub-optimality with respect to the original problem and illustrate the effectiveness of the method by showing experiment results on large scale real-life datasets.

## 3.2 Data thresholding for sparse linear classification problems

We consider the sparse linear classification problem

$$\phi_\lambda(X) := \min_{w,b} \frac{1}{m} \sum_{i=1}^{m} f(y_i(x_i^T w + b)) + \lambda \|w\|_1 \tag{3.2.1}$$

and its constrained counterpart

$$\psi_c(X) := \min_{w,b} \frac{1}{m} \sum_{i=1}^{m} f(y_i(x_i^T w + b)) \ : \ \|w\|_1 \le c \tag{3.2.2}$$

with $\lambda, c$ non-negative parameters. Here the data matrix $X = [x_1, \dots, x_m] \in \mathbf{R}^{n \times m}$ is given.

We assume that the loss function $f$ is **decreasing** and **Lipschitz** with constant L, i.e.

$$f(u) \ge f(v), \forall u \le v \text{ and } |f(u) - f(v)| \le L|u - v|$$

Note that both hinge loss and logistic loss satisfy above assumption with constant $L = 1$

In the next sections, we will examine the effect of thresholding the data matrix according to an absolute threshold rule. The rule is defined by a parameter $t$ and the thresholded matrix is $X(t)$, with, for $1 \le i \le n$, $1 \le j \le m$:

$$X_{ij}(t) = \begin{cases} X_{ij} & \text{if } |X_{ij}| > t, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, we have $\|X - X(t)\|_\infty \le t$, where $\| \cdot \|_\infty$ is the largest magnitude of the entries of its matrix argument. In the sequel, we denote by $x_i$, $x_i(t)$ and $z_i$ the $i$-th column of matrices $X$, $X(t)$, and $Z$, $i = 1, \dots, m$.

Our focus here is to determine the largest value of the threshold level $t$ under which we can guarantee that a (suboptimal) solution the problem with $t$-thresholded data achieves a similar suboptimality condition for the original problem.

### 3.2.1 Penalized problem

For a given matrix $X \in \mathbf{R}^{n\times m}$, we define $w(X), b(X)$ to be a minimizer for problem (3.2.1) with data matrix $X$, which means that:

$$\phi_\lambda(X) = \frac{1}{m}\sum_{i=1}^m f(y_i(x_i^T w(X) + b(X))) + \lambda\|w(X)\|_1.$$

Evaluating the objective function of (3.2.1) at $w = 0, b = 0$ we obtain $\phi_\lambda(X) \leq f(0)$, which implies $\|w(X)\|_1 \leq f(0)/\lambda$ for any $X$.

Consider two matrices $X, Z \in \mathbf{R}^{n\times m}$ such that $\|X - Z\|_\infty \leq t$. Using convexity, monotonicity and homogeneity of the loss function $f$, we can show that for every $w$ that is feasible for (3.2.2):

$$\begin{aligned}
\phi_\lambda(X) &\leq \frac{1}{m}\left\{\sum_{i=1}^m f(y_i(x_i^T w(Z) + b(Z)))\right\} + \lambda\|w(Z)\|_1 \\
&= \frac{1}{m}\left\{\sum_{i=1}^m f\left(y_i(z_i^T w(Z) + b(Z))) + y_i(x_i - z_i)^T w(Z)\right)\right\} + \lambda\|w(Z)\|_1 \\
&\leq \frac{1}{m}\left\{\sum_{i=1}^m f\left(y_i(z_i^T w(Z) + b(Z))) + t\|w(Z)\|_1\right)\right\} + \lambda\|w(Z)\|_1 \\
&\leq \frac{1}{m}\left\{\sum_{i=1}^m f\left(y_i(z_i^T w(Z) + b(Z))\right) + Lt\|w(Z)\|_1\right\} + \lambda\|w(Z)\|_1 \\
&\leq \frac{1}{m}\left\{\sum_{i=1}^m f(y_i(z_i^T w(Z) + b(Z))) + \frac{Ltf(0)}{\lambda}\right\} + \lambda\|w(Z)\|_1 \\
&= \phi_\lambda(Z) + \frac{Ltf(0)}{\lambda}
\end{aligned} \tag{3.2.3}$$

Therefore, for any two $n \times m$ matrices $X, Z$ with $\|X - Z\|_\infty \leq t$, we have

$$\phi_\lambda(X) \leq \frac{1}{m}\sum_{i=1}^m f(y_i(x_i^T w(Z) + b(Z))) + \lambda\|w(Z)\|_1 \leq \phi_\lambda(Z) + \frac{Ltf(0)}{\lambda}. \tag{3.2.4}$$

By exchanging the roles of $X, Z$ we can show that: $|\phi_\lambda(X) - \phi_\lambda(Z)| \leq \dfrac{Ltf(0)}{\lambda}$.

Revisiting the sequence (3.2.4) with $Z = X(t)$, and denoting $w(X(t)) = w(t)$, we obtain

$$\phi_\lambda(X) \le \frac{1}{m}\sum_{i=1}^{m} f(y_i(x_i^T w(t) + b(t))) + \lambda\|w(t)\|_1 \le \phi_\lambda(X(t)) + \frac{Ltf(0)}{\lambda} \le \phi_\lambda(X) + \frac{2Ltf(0)}{\lambda}$$

We have obtained

$$0 \le \frac{1}{m}\sum_{i=1}^{m} f(y_i(x_i^T w(t) + b(t))) + \lambda\|w(t)\|_1 - \phi_\lambda(X) \le \frac{2Ltf(0)}{\lambda}.$$

The above implies that solving the penalized problem with thresholded data $X(t)$ in lieu of $X$ we achieve $\epsilon$-suboptimality with $w(t)$ with respect to the original problem provided $t \le (\lambda\epsilon)/(2Lf(0))$. This satisfies the intuition that if $\lambda$ and/or $\epsilon$ are large, then more thresholding can take place.

**Relative bound for penalized problem**

Another observation on the penalized problem is that not only do we have $\|w(X)\|_1 \le f(0)/\lambda$ but we also have $\|w(X)\|_1 \le \phi_\lambda(X)/\lambda$ (this is stronger then $\|w(X)\|_1 \le f(0)/\lambda$ since we always have $\phi_\lambda(X) \le f(0)$). With this observation, (3.2.4) can be modified to get

$$\phi_\lambda(X) \le \phi_\lambda(Z)\left(1 + \frac{Lt}{\lambda}\right)$$

Therefore, in addition to **additive** bound we obtain the following **multiplicative** bound:

$$1 \le \frac{\frac{1}{m}\sum_{i=1}^{m} f(y_i(x_i^T w(t) + b(t))) + \lambda\|w(t)\|_1}{\phi_\lambda(X)} \le \left(1 + \frac{Lt}{\lambda}\right)^2$$

In other words,

$$0 \le \frac{1}{m}\sum_{i=1}^{m} f(y_i(x_i^T w(t) + b(t))) + \lambda\|w(t)\|_1 - \phi_\lambda(X) \le \theta\phi_\lambda(X)$$

where $\theta := \left(1 + \frac{Lt}{\lambda}\right)^2 - 1$. Thus to achieve $\epsilon$ in relative accuracy, we can set

$$t := \frac{\lambda(\sqrt{1+\epsilon} - 1)}{L}$$

### 3.2.2 Constrained problem

For a given $n \times m$ matrix $X$, we define $w(X), b(X)$ to be a minimizer for problem (3.2.2) with data matrix $X$.

Consider two matrices $X, Z \in \mathbf{R}^{n\times m}$ such that $\|X - Z\|_\infty \leq t$. Using convexity, monotonicity and homogeneity of the loss function $f$, we can show that for every $w$ that is feasible for (3.2.2):

$$\begin{aligned}
\psi_c(X) &\leq \frac{1}{m}\sum_{i=1}^{m} f(y_i(x_i^T w(Z) + b(Z))) \\
&= \frac{1}{m}\sum_{i=1}^{m} f\left(y_i(z_i^T w(Z) + b(Z))) + y_i(x_i - z_i)^T w(Z)\right) \\
&\leq \frac{1}{m}\sum_{i=1}^{m} f\left(y_i(z_i^T w(Z) + b(Z))) + t\|w(Z)\|_1\right) \qquad (3.2.5)\\
&\leq \frac{1}{m}\sum_{i=1}^{m} f\left(y_i(z_i^T w(Z) + b(Z))\right) + Lt\|w(Z)\|_1 \\
&\leq \frac{1}{m}\sum_{i=1}^{m} f(y_i(z_i^T w(Z) + b(Z))) + Ltc = \psi_c(Z) + Ltc.
\end{aligned}$$

By exchanging the roles of $X$ and $Z$, we can show that: $|\psi_c(X) - \psi_c(Z)| \leq Ltc$.

Revisiting the above steps with $Z = X(t)$, and denoting $w(X(t)) = w(t), b(X(t)) = b(t)$, we obtain:

$$\psi_c(X) \quad \leq \frac{1}{m}\sum_{i=1}^{m} f(y_i(x_i^T w(t) + b(t))) \leq \psi_c(X(t)) + Ltc \leq \psi_c(X) + 2Ltc \qquad (3.2.6)$$

Therefore,

$$0 \leq \frac{1}{m}\sum_{i=1}^{m} f(y_i(x_i^T w(t) + b(t))) - \psi_c(X) \leq 2Ltc \qquad (3.2.7)$$

Thus, if we solve the problem with thresholded data $X(t)$ in lieu of $X$ we achieve $\epsilon$-suboptimality for the original problem provided $t \leq \epsilon/(2Lc)$. This satisfies the intuition that if $c$ is small, and/or $\epsilon$ is large, then more thresholding can take place.

**Optimality of bounds**

Our results provide an upper bound on how much thresholding can take place while maintaining a given level of sub-optimality. We will show that these bounds are optimal, in the sense that they are attained by some choice of a data set. We consider the constrained SVM problem (problem (3.2.2))

Assume that the data set consists of $m = 2$ vectors, $X = [x, -x] \in \mathbf{R}^{n\times m}$ where $x = (t, 0, \dots, 0) \in \mathbf{R}^n$, and $y_1 = 1, y_2 = -1$. We have

$$\begin{aligned}\psi_c(X) &= \min_{\|w\|_1\le c} \frac{1}{2}\left((1-(w_1t+b))_+ + (1+(-w_1t+b))_+\right)\\ &\ge \min_{\|w\|_1\le c} \frac{1}{2}\left((1-(w_1t+b)+1+(-w_1t+b)\right)_+\\ &= \min_{\|w\|_1\le c} \frac{1}{2}(2-2w_1t)_+ \ge (1-ct)_+\end{aligned}$$

Equality occurs when $w = (c, 0, \dots, 0), b = 0$ hence $\psi_c(X) = (1-ct)_+$

On the other hand,

$$\psi_c(X(t)) = \min_{w,b\,:\,\|w\|_1\le c} \frac{1}{2}\left((1-b)_+ + (1+b)_+\right) \ge \min_{\|w\|_1\le c} \frac{1}{2}\left(1-b+1+b)\right)_+ = 1$$

Equality occurs for any $b(t) \in [-1, 1], w(t) : \|w(t)\|_1 \le c$ hence $\psi_c(X(t)) = 1$

By choosing $b(t) = 0, w(t) = (c, 0, \dots, 0)$ we have:

$$\frac{1}{m}\sum_{i=1}^{m}(1 - y_i(x_i^T w(t) + b(t))_+ = (1-ct)_+ = \psi_c(X)$$

By choosing $b(t) = 0, w(t) = (-c, 0, \dots, 0)$ we have:

$$\frac{1}{m}\sum_{i=1}^{m}(1 - y_i(x_i^T w(t) + b(t)))_+ = (1+ct)_+ = \psi_c(X) + 2ct$$

Therefore, the sub-optimal bound we obtain in equation (3.2.7) is tight.

### 3.2.3 Iterative thresholding

Suppose that we have a sequence of regularization parameters $\lambda_0 > \lambda_1 > \ldots$ and the corresponding thresholding levels $t_0, t_1, \ldots$

Let's define:

$$\phi_\lambda(X, w, b) := \frac{1}{m} \sum_{i=1}^{m} f(y_i(w^T x_i + b)) + \lambda \|w\|_1$$

and $(w^{(k)}, b^{(k)}) = \arg\min_{w,b} \phi_{\lambda_k}(X)$, thus

$$\phi_{\lambda_k}(X) = \min_{w,b} \phi_{\lambda_k}(X, w, b) = \phi_{\lambda_k}(X, w^{(k)}, b^{(k)})$$

Also define the objective obtained in step $k^{th}$ to be $\hat{\phi}_k = \phi_{\lambda_k}(X, w(t_k), b(t_k))$

We observe that:

$$\phi_{\lambda_{k+1}}(X) \leq \phi_{\lambda_k}(X) \leq \phi_{\lambda_k}(X, w(t_k), b(t_k)) = \hat{\phi}_k$$

Using relative error bound in Section 1.2, we have

$$0 \leq \frac{1}{m} \sum_{i=1}^{m} f(y_i(x_i^T w(t_{k+1}) + b(t_{k+1}))) + \lambda_{k+1} \|w(t)\|_1 - \phi_{\lambda_{k+1}}(X) \leq \theta \phi_{\lambda_{k+1}}(X) \leq \theta \hat{\phi}_k$$

where $\theta := \left(1 + \frac{L t_{k+1}}{\lambda_{k+1}}\right)^2 - 1$

So to obtain $\epsilon$ absolute error, we need $\theta \hat{\phi}_k \leq \epsilon$ which means that we can do more thresholding by setting the $(k+1)$ thresholding level to

$$t_{k+1} := \frac{\lambda_{k+1}(\sqrt{1 + \epsilon / \hat{\phi}_k} - 1)}{L}$$

## 3.3 Experimental results

All experiments are conducted on a personal workstation with 16GB RAM and 2.6GHz quad-core Intel processor.

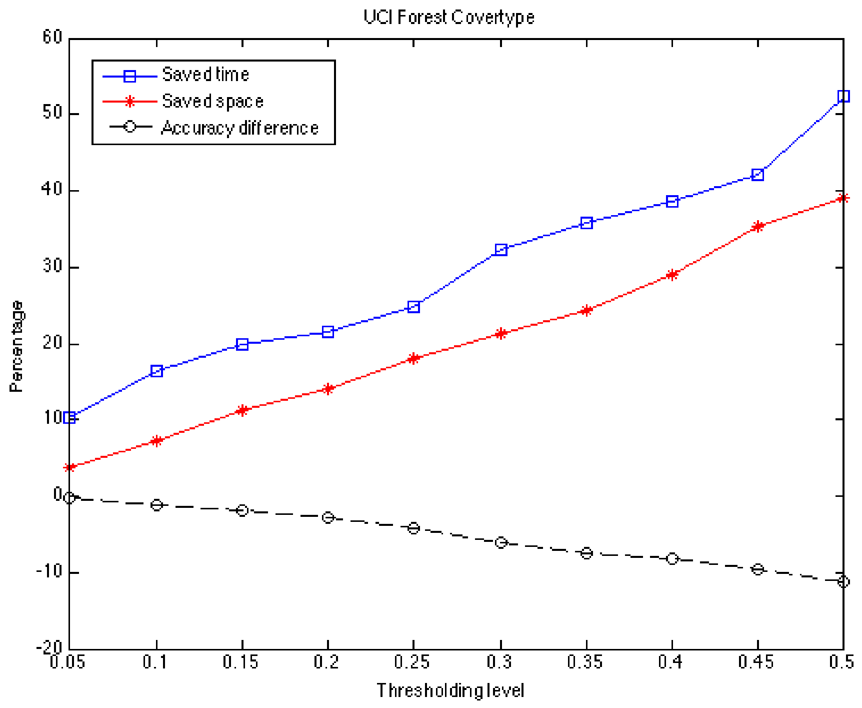


Figure 3.1. Result of SVM with data thresholding on UCI Forest Covertype dataset.

### 3.3.1 UCI Forest Covertype dataset

[1] We modified the 7-class classification problem into a binary classification problem where the goal was to separate class 1 (Spruce/Fir) from the other 6 classes. Each example was described by 54 input features, in which the first 10 are real numbers, the rest are binary variable. We normalized the first ten by scaling the first 10 attributes to $[-1, 1]$. The dataset had more than 500,000 examples. We sample 10 times, each time we randomly select 50,000 examples to be training data and 50,000 examples to be test data. We compute the average statistics on these 10 samples and show the result in Figure 3.1.

As can be seen from Figure 3.1, as we increase the thresholding level, we can solve the

[1] http://archive.ics.uci.edu/ml/datasets/Covertype

| t = 0 | t = 0.02 | t = 0.04 | t = 0.06 | t = 0.08 | t = 0.1 |
|---|---|---|---|---|---|
| 'polygon' | 'polygon' | 'polygon' | 'polygon' | 'polygon' | 'pov' |
| 'animation' | 'animation' | 'animation' | 'tiff' | 'pov' | 'graphics' |
| 'tiff' | 'sphere' | 'tiff' | 'graphics' | 'graphics' | 'polygon' |
| 'graphics' | 'tiff' | 'graphics' | 'pov' | 'animation' | 'tiff' |
| 'cview' | 'graphics' | 'sphere' | 'sphere' | 'tiff' | 'sphere' |
| 'sphere' | 'images' | 'pov' | 'images' | 'sphere' | 'cview' |
| 'pov' | 'cview' | 'images' | 'cview' | 'cview' | 'images' |
| 'images' | 'pov' | 'cview' | 'animation' | 'images' | 'bezier' |
| 'mpeg' | 'mpeg' | 'bezier' | 'bezier' | 'bezier' | 'animation' |
| 'image' | 'image' | 'mpeg' | 'diablo' | 'image' | 'wireframe' |

Table 3.1. Top 10 keywords obtained with topic **comp.graphics** as the thresholding level increases. The top keywords do not change significantly when more thresholding applies.

problem faster (blue line) and use less memory (red line) to store the data by sacrificing some minor amount of accuracy (black line).

### 3.3.2 20 Newsgroup Dataset

This dataset consist of Usenet articles Lang collected from 20 different newsgroups (e.g. comp.graphics, comp.windows.x, misc.forsale, sci.space, talk.reglision.misc, etc.) which contains an overall number of 20000 documents. Except for a small fraction of the articles, each document belongs to exactly one newsgroup. The task is to learn which newsgroup an article was posted to. The dataset is divided so that the training set contains 2/3 the number of examples and the test set contains the rest. The dimension of the problem is 61K (number of words in the dictionary). For each document, word we compute the corresponding TFIDF score and then threshold the data based on TF-IDF score.

In Table 3.1, we show the list of top 20 keywords obtained with topic **comp.graphics** as we varied the thresholding level, as can be seen as we increase the thresholding levels the top

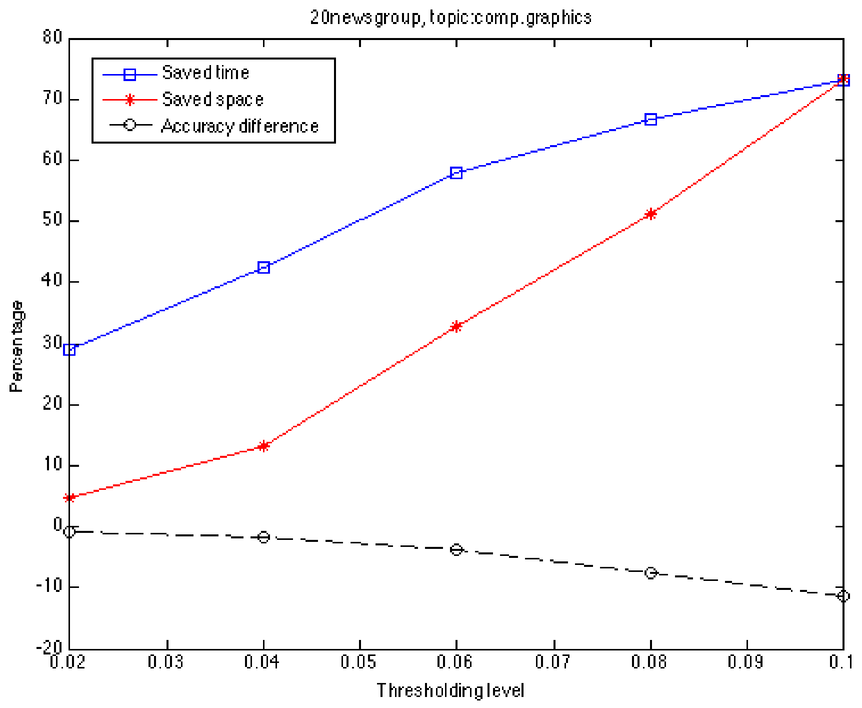


Figure 3.2. Speed-up and space-saving for SVM with data thresholding on 20 Newsgroup dataset, topic = comp.graphics. Number of training samples: 13K, number of features: 61K.

features (words) do not change significantly. This suggests that the result (weight vector) of sparse linear classification with thresholding can still be used for feature selection.

Figure 3.2 shows the speed-up and space-saving for SVM with data thresholding on 20 Newsgroup dataset for topic **comp.graphics**. As we increase the thresholding level, we can solve the problem faster (red line) and use less memory (blue line) to store the data by sacrificing some minor amount of accuracy (black line)

### 3.3.3 UCI NYTimes dataset

UCI NYTimes dataset contains 300,000 news articles with the vocabulary consists of 102,660 words. In this experiment, we use the first 100,000 news articles (which has approx-

| | | | |
|---|---|---|---|
| 1 | stock | 11 | bond |
| 2 | nasdaq | 12 | forecast |
| 3 | portfolio | 13 | thomson financial |
| 4 | brokerage | 14 | index |
| 5 | exchanges | 15 | royal bank |
| 6 | shareholder | 16 | fund |
| 7 | fund | 17 | marketing |
| 8 | investor | 18 | companies |
| 9 | alan greenspan | 19 | bank |
| 10 | fed | 20 | merrill |

Table 3.2. SVM Result: Top 20 keywords for topic '**stock**' on UCI NYTimes dataset with thresholding level of 0.05 on TF-IDF score (reduced dataset is only 3% of the full dataset). Total runtime: 4317s

imately 30 millions words in the collection) for training task. It is impossible to train SVM with the whole dataset on a laptop/PC, so in this experiment we will just run SVM on the thresholded dataset (by TF-IDF scores) and then report the top features in Table 3.2. The thresholded dataset size with the thresholding level 0.05 only contains 850,000 non-zeroes, which is roughly 3% of the full dataset.

# Chapter 4

# Low-rank Approximation

## 4.1 Introduction

A standard task in scientific computing is to determine for a given matrix $A \in \mathbf{R}^{n \times d}$ an approximate decomposition $A \approx BC$ where $B \in \mathbf{R}^{n \times k}, C \in \mathbf{R}^{k \times d}$; $k$ is called the numerical rank of the matrix. When $k$ is much smaller than either $n$ or $d$, such decomposition allows the matrix to be stored inexpensively, and to be multiplied to vectors or other matrices quickly.

Low-rank approximation has been used for solving many large-scale problems that involves large amounts of data. By replacing the original matrices with approximated (low-rank) ones, the perturbed problems often requires much less computational effort to solve. Low-rank approximation methods have been shown to be successful on a variety of learning tasks, such as Spectral partitioning for image and video segmentation Fowlkes et al. (2004), Manifold learning Talwalkar et al. (2008). Recently, there have been interesting works on using low-rank approximation in Kernel learning: Fine and Scheinberg (2002) proposed an efficient method for kernel SVM learning problem by approximating the kernel matrix by a low-rank positive semidefinite matrix. Bach and Jordan (2005) and Zhang et al. (2012) also proposed similar approaches for more general kernel learning problems which also exploit side information in the computation of low-rank decompositions for kernel matrices.

One important point worth noting here is that all of the above works just replace the original data matrices directly with the low-rank ones, and then provide a bound on the error made by solving the perturbed problem compared to the solution of the original problem. In this thesis, we propose a new modeling approach that takes into account the approximation error made when replacing the original matrices with low-rank approximations to modify the original problem accordingly, via **robust optimization** perspective.

In this chapter, we focus on solving LASSO problem:

$$\min_{\|\beta\|_1 \leq \lambda} \|y - X^T\beta\|_2 \tag{4.1.1}$$

However, the result can also be generalized to more supervised learning problems:

$$\min_{v \in \mathbf{R}, w \in \mathcal{C}} f(A^T w + cv) \tag{4.1.2}$$

where the loss function $f : \mathbf{R}^n \to \mathbf{R}$ is convex, the data matrix $A \in \mathbf{R}^{d \times n}$ and vector $c \in \mathbf{R}^n$ are given, $\mathcal{C} \subseteq \mathbf{R}^d$ is a convex set that constraining the decision variable $w$.

In practice, a low-rank approximation may not be directly available, but has to be computed. While the effort in finding low-rank approximation is typically linear in the size of data matrix, in our approach leads to the biggest savings in what we refer to as the *repeated instances* setup. In such a setup, the task is to solve multiple instances of similar programs, where all the instances involve a common (or very similar) coefficient matrix. In that setup, of which we provide real-world examples later, it makes sense to invest some effort in finding the low-rank approximation once, and then exploit the same low-rank approximation for each instance. The robust low-rank approach then offers enormous computational savings, and produces solutions that are guaranteed to be feasible for the original problem.

## 4.2 Algorithms for Low-rank Approximation

In general, to find rank-k approximation of a matrix $A \in \mathbf{R}^{n \times d}$, we wish to find matrices $B \in \mathbf{R}^{n \times k}, C \in \mathbf{R}^{k \times d}$ such that the *spectral norm/largest singular value norm* error $\|A - BC\|_2$ is minimal. The truncated singular value decomposition is known to provide the best low-rank approximation for any given fixed rank Eckart and Young (1936); however, it is also very costly to compute. There have been many different approaches proposed for computing low-rank approximations, such as rank-revealing factorization (QR or LU) Gu and Eisenstat (1996); Miranian and Gu (2003), subspace iteration methods, Lancoz algorithms Demmel and Heath (1997). Recently, randomized algorithms have been developed and proved to be reliable and computationally efficient (see a comprehensive review in Halko et al. (2011)). In this work, we use fast randomized subsampling algorithm (Algorithm 1) presented in Halko et al. (2011).

**Algorithm 1** Fast Randomized Subsampling (Halko et al. (2011))

```
Input: $A \in \mathbf{R}^{n \times d}$ with $d \leq n$, $l \geq k > 0$
Output: a rank-k approximation
1. Draw an $d \times l$ SRFT test matrix $\Omega$
2. Form the $n \times l$ matrix $Y = A\Omega$ using a (subsampled) FFT.
3. Compute an orthogonal column basis $Q$ for $Y$
4. Compute $B = Q^T A$
5. Compute $B_k$, the rank-k truncated SVD of $B$
6. Return $QB_k$
```

In algorithm (1), SRFT (subsampled random Fourier transform) test matrix $\Omega$ is of the form $\Omega = \sqrt{d/l}DFR$, where $D \in \mathbf{R}^{d \times d}$ diagonal matrix whose entries are independent random signs, $F \in \mathbf{R}^{d \times d}$ is the unitary discrete Fourier transform: $F_{pq} =$

$d^{-1/2}\exp\left(-2\pi i(p-1)(q-1)/d\right)$ and $R \in \mathbf{R}^{d\times l}$ which samples $l$ coordinates from $d$ uniformly at random.

Computing $Y = A\Omega$ can be done in $O(nd\log(l))$ flops via a subsampled fast Fourier transform. If $A$ has special structure (e.g. sparsity) this can be fastened to $O(lN_A+(n+d)k^2)$ where $N_A$ is the number of non-zero entries in $A$.

When $l \geq \Omega((k+\log(kd))\log(k))$ then with high probability we have:

$$\|A - QB_k\| \leq O\left(\sqrt{\frac{d}{l}}\right)\sigma_{k+1}(A).$$

### 4.2.1 First order methods with Low-rank approximation

There has been extensive research on algorithms for solving the constrained LASSO problem. Tibshirani (1996), Osborne et al. (2000) and Efron et al. (2004) amongst many others developed algorithms using quadratic programming methods. These algorithms often need to compute matrix inversions repeatedly, the computational cost increases rapidly as the input size increases and hence they might not be applicable to very large scale datasets.

First order methods have also been used to solve the constrained LASSO problem: NESTA - Nesterov's proximal gradient method by Becker et al. (2011), GPSR - projected gradient method by Figueiredo et al. (2007), PGN - projected quasi-Newton algorithm by Schmidt et al. (2009), SPG - Spectral projected gradient algorithm by Birgin et al. (2003), etc. In this work, we will use SPG (Algorithm 2) to solve LASSO and its variations, due to its effectiveness for very large-scale convex optimization. SPG is also known to be more efficient than classical gradient projected method due to spectral step length selection and non-monotone search Birgin et al. (2003); Figueiredo et al. (2007); van den Berg et al. (2008). However, our work can be applied to any first-order algorithm for solving the constrained LASSO problem.

**Algorithm 2** Spectral Projected Gradient Algorithm (Birgin et al. (2000))

```
goal: $\min_{x \in C} f(x)$, where $f : \mathbf{R}^n \to \mathbf{R}$ convex, differentiable; $C \subseteq \mathbf{R}^n$ is a closed convex set.
parameters: steplength bound parameters $\alpha_{\max} > \alpha_{\min} > 0$; safeguarding parameters $0 < \sigma_1 < \sigma_2 < 1$; $\gamma \in (0,1)$.
initialize $k = 0, x^{(0)} \in C, \alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$.
while $\|P_C(x^{(k)} - \nabla f(x^{(k)})) - x^{(k)}\| \geq \epsilon$ do
    1. compute $d^{(k)} = P_C\left(x^{(k)} - \alpha_k \nabla f(x^{(k)})\right) - x^{(k)}$
    2. line search
        (i) set $x_+ \leftarrow x^{(k)} + d^{(k)}, \eta = 1$ and $\delta \leftarrow \nabla f(x^{(k)})^T d^{(k)}$
        (ii) while $f(x_+) > \max_{0 \leq j \leq \min(k,M)} f\left(x^{(k-j)}\right) + \gamma\eta\delta$
            • compute $\hat{\eta} = \dfrac{-\eta^2\delta}{2(f(x_+) - f(x^{(k)}) - \eta\delta)}$
            • if $\hat{\eta} \geq \sigma_1\eta$ and $\hat{\eta} \leq \sigma_2\eta$ then set $\eta \leftarrow \hat{\eta}$, else set $\eta \leftarrow \eta/2$
            • set $x_+ \leftarrow x^{(k)} + \eta d^{(k)}$
        (iii) set steplength $\rho_k \leftarrow \eta$
    3. set $x^{(k+1)} = x^{(k)} + \rho_k d^{(k)}, s^{(k)} = x^{(k+1)} - x^{(k)}$, and $y^{(k)} = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)})$
    4. if $(s^{(k)})^T y^{(k)} \leq 0$ then set $\alpha_{k+1} \leftarrow \alpha_{\max}$
       else set $\alpha_{k+1} \leftarrow \min\left(\max\left(\dfrac{(s^{(k)})^T s^{(k)}}{(s^{(k)})^T y^{(k)}}, \alpha_{\min}\right), \alpha_{\max}\right)$
    5. set $k \leftarrow k + 1$
```

*($P_C(z)$ denotes the projection of vector $z$ onto the set $C$ : $P_C(z) = \arg\min_{w \in C} \|z - w\|$)*

For sparse learning problems such as LASSO, the set $C$ is the $l_1$-ball $B_1(\lambda) = \{w : \|w\|_1 \leq \lambda\}$. We adopt the following algorithm developed by Duchi et al. (2008) to compute projections onto the $l_1$ ball efficiently in linear time:

**Algorithm 3** Linear time projection onto $l_1$ ball (Duchi et al. (2008))

```
input: $z \in \mathbf{R}^d$ and $\lambda > 0$
initialize $U = \{1, \ldots, d\}, s = 0, \eta = 0$
while $U \neq \emptyset$ do
    1. pick $k \in U$ randomly.
    2. partition $U$ into $G = \{j \in U | z_j \geq z_k\}$, $L = U \setminus G$
    3. if $(s + \sum_{j \in G} z_j) - (\eta + |G|) z_k < \lambda$ then $s \leftarrow s + \sum_{j \in G} z_j; \eta \leftarrow \eta + |G|; U \leftarrow L$
    4. else $U \leftarrow G \setminus \{k\}$
set $\theta = (s - \lambda)/\eta$
output $w = \{\max(z_i - \theta, 0)\}_{i=1}^d$
```

We now show how to use low-rank approximation to speedup first order methods. As we will see, the speedup when applying our approach will be achieved with any first order algorithm that involves computing the gradient. Generally, first-order method for LASSO requires us to compute the gradient of the objective function in which for the constrained LASSO problem, it is of the form:

$$\nabla_\beta \|y - X^T \beta\|_2 = X \frac{X^T \beta - y}{\|X^T \beta - y\|_2}$$

Finding such gradient involves computing the product of a matrix $X \in \mathbf{R}^{n \times d}$ and a vector $u \in \mathbf{R}^d$ as well as the product of $X^T$ with a vector $v \in \mathbf{R}^n$. Each operation costs $O(nd)$ flops for dense matrix $X$ or $O(N)$ flops for sparse matrix $X$ with $N$ non-zero entries.

Assuming that the data matrix has a low-rank structure, that is $X \approx UV^T$ where $U \in \mathbf{R}^{n\times k}, V \in \mathbf{R}^{d\times k}$ with $k << \min(n,d)$. We can then exploit the low-rank structure in order to improve the efficiency of the above matrix-vector multiplication by writing $Xu = U(V^Tu)$. Computing $z = V^Tu$ costs $O(kd)$ flops, and computing $Uz$ costs $O(kn)$ flops, thus the total number of flops is $O(k(n+d))$. This is a significant improvement compared to the cost of matrix-vector multiplication on the original data matrix, especially when $X$ is dense, high dimensional and approximately low-rank. When $X$ is sparse and $k$ is much smaller than the average number of non-zero entries per sample, the matrix-vector multiplication computation that exploits the low-rank structure is also much faster than direct multiplication. Furthermore, the space complexity required to store the low-rank approximation is only $O(k(n+d))$ compared to $O(nd)$ when $X$ is dense or $O(N)$ if X is a sparse matrix.

For more general supervised learning problem (4.1.2), we could also compute the gradient exploiting the low-rank structure of the data matrix: suppose that $A = VU^T$ where $U \in \mathbf{R}^{n\times k}, V \in \mathbf{R}^{d\times k}$ with $k << \min(n,d)$, let $h(w,v) = f(A^Tw + cv)$, the gradient can be computed as following:

$$\begin{aligned} \nabla_w h(w,v) &= A\nabla f(A^Tw + cv) = VU^T\nabla f(UV^Tw + cv) \\ \nabla_v h(w,v) &= c^T\nabla f(A^Tw + bv) = c^T\nabla f(UV^Tw + cv) \end{aligned}$$

Similarly, the cost of computing the above gradient is only $O(k(n+d))$ for dense data matrix $X$ or $O(N)$ for sparse one.

## 4.3 Robust Low-rank LASSO

Important questions worth asking when replacing the original data matrix by a low-rank approximation are "*how much information do we lose*?", and "*how does it effect the LASSO* *problem*?". In this section, we will examine the effect of using low-rank approximation, or in other words, thresholding the singular values of the data matrix according to an absolute threshold level $\epsilon_k$. We replace the matrix $X$ by $\hat{X}$, the closest (in largest singular value norm) rank-$k$ approximation to $X$, and the error $\Delta := \hat{X} - X$ satisfies $\|\Delta\| \le \epsilon_k$, where $\|\cdot\|$ denotes the largest singular value norm.

As seen in previous section, computing the low-rank approximation of a matrix is feasible even in a large scale setting. Our basic goal is to end up solving a slightly modified LASSO problem using rank-$k$ approximation, while controlling for the error made.

We want to take into account the worst-case error in the objective function that is made upon replacing $X$ with its rank-$k$ approximation, $\hat{X}$. To this end, we consider the following robust counterpart to (4.1.1):

$$\begin{aligned}\psi_{\epsilon_k,\lambda}(\hat{X}) &:= \min_{\|\beta\|_1 \le \lambda} \max_{\|X - \hat{X}\| \le \epsilon_k} \left\|y - X^T\beta\right\|_2 \\ &= \min_{\|\beta\|_1 \le \lambda} \max_{\|\Delta\| \le \epsilon_k} \left\|y - (\hat{X} - \Delta)^T\beta\right\|_2\end{aligned} \tag{4.3.1}$$

Let us define $f(z) = \|z\|_2$, we have:

$$\begin{aligned}
\max_{\|\Delta\|\leq\epsilon_k} \left\|y-(\hat{X}-\Delta)^T\beta\right\|_2 &= \max_{\|\Delta\|\leq\epsilon_k} f(y-\hat{X}^T\beta+\Delta^T\beta)\\
&= \max_{\|\Delta\|\leq\epsilon_k}\max_{u\in\mathbf{R}^m}\left\{u^T(y-\hat{X}^T\beta+\Delta^T\beta)-f^*(u)\right\}\\
&= \max_{u\in\mathbf{R}^m}\left\{u^T(y-\hat{X}^T\beta)-f^*(u)+\max_{\|\Delta\|\leq\epsilon_k} u^T\Delta^T\beta\right\}\\
&= \max_{u\in\mathbf{R}^m}\left\{u^T(y-\hat{X}^T\beta)-f^*(u)+\max_{\|\Delta\|\leq\epsilon_k} \langle\Delta^T,\beta u^T\rangle\right\}\\
&= \max_{u\in\mathbf{R}^m}\left\{u^T(y-\hat{X}^T\beta)-f^*(u)+\epsilon_k\|\beta u^T\|_*\right\}\\
&= \max_{u\in\mathbf{R}^m}\left\{u^T(y-\hat{X}^T\beta)-f^*(u)+\epsilon_k\|\beta\|_2\|u\|_2\right\}\\
&= \max_{u\in\mathbf{R}^m}\left\{u^T(y-\hat{X}^T\beta)-f^*(u)+\max_{\|z\|_2\leq\epsilon_k\|\beta\|_2} u^Tz\right\}\\
&= \max_{\|z\|_2\leq\epsilon_k\|\beta\|_2}\max_{u\in\mathbf{R}^m}\left\{u^T(y-\hat{X}^T\beta+z)-f^*(u)\right\}\\
&= \max_{\|z\|_2\leq\epsilon_k\|\beta\|_2} f(y-\hat{X}^T\beta+z)\\
&= \max_{\|z\|_2\leq\epsilon_k\|\beta\|_2} \|y-\hat{X}^T\beta+z\|_2\\
&= \|y-\hat{X}^T\beta\|_2+\epsilon_k\|\beta\|_2
\end{aligned}$$

here $\|.\|_*$ is the nuclear norm (the dual of spectral norm) and $f^*$ is the conjugate dual of $f$.

Therefore, we can write the robust counterpart of LASSO problem in (2) as:

$$\psi_{\epsilon_k,\lambda}(\hat{X}) = \min_{\|\beta\|_1\leq\lambda} \|y-\hat{X}^T\beta\|_2+\epsilon_k\|\beta\|_2 \tag{4.3.2}$$

Let $g(\beta)=\|y-\hat{X}^T\beta\|_2+\epsilon_k\|\beta\|_2$, its gradient is:

$$\nabla_\beta g(\beta) = \hat{X}\frac{\hat{X}^T\beta-y}{\|\hat{X}^T\beta-y\|_2}+\epsilon_k\frac{\beta}{\|\beta\|_2}$$

Hence the cost of computing this gradient is also similar as for the low-rank LASSO problem, which is $O(k(n+d))$.

### 4.3.1 Theoretical Analysis

In this section, we present some theoretical analysis for the Low-rank LASSO models, in particular we bound how far their solutions from the true weight vector. In order to do so, we will need the following definitions:

**Restricted nullspace** (Cohen et al. (2009)): for a given subset $S \subseteq \{1, \ldots, d\}$ and a constant $\alpha \geq 1$, define:

$$\mathcal{C}(S, \alpha) := \{\theta \in \mathbf{R}^d : \|\theta_{S^C}\|_1 \leq \alpha \|\theta_S\|_1\}$$

Given $k \leq d$, the matrix $X$ is said to satisfy the restricted nullspace condition of order $k$ if $null(X) \cap \mathcal{C}(S, 1) = \{0\}, \forall S \subseteq \{1, \ldots, d\} : |S| = k$.

**Restricted eigenvalue** (Bickel et al. (2008)): the sample covariance matrix $X^T X/n$ is said to satisfy the restricted eigenvalue condition over a set $S$ with parameters $\alpha \geq 1, \gamma > 0$ if $\frac{1}{n}\|X\theta\|_2^2 \geq \gamma^2 \|\theta\|_2^2, \forall \theta \in \mathcal{C}(S, \alpha)$. We denote that $X \in RE(S, \alpha, \gamma)$ if the above condition is true.

**Main result:** we consider the classical linear model with noisy settings: $y = X\beta^* + w$, where $y \in \mathbf{R}^n$ is the vector of responses, the matrix $X \in \mathbf{R}^{n \times d}$ is feature matrix, and $w \in \mathbf{R}^d$ is a random white noise vector: $w \sim \mathcal{N}(0, \sigma^2 I_{n \times n})$. Let define:

$$\begin{aligned} \bar{\beta}(\lambda) &= \arg \min_{\|\beta\|_1 \leq \lambda} \|y - X\beta\|_2 \\ \hat{\beta}(\lambda, \eta) &= \arg \min_{\|\beta\|_1 \leq \lambda} \|y - \hat{X}\beta\|_2 + \eta \|\beta\|_2 \end{aligned}$$

Hence, $\hat{\beta}(\lambda, \epsilon_k)$ is the solution for the robust counterpart optimization problem (4.3.2) and $\hat{\beta}(\lambda, 0)$ is the solution for the LASSO problem if we just replace the feature matrix $X$ by its low rank approximation $\hat{X}$.

We will now show that when the feature matrix $X$ is "close" to low-rank (i.e. $\epsilon_k$ is small), the estimators $\hat{\beta}(\lambda, \eta)$ is closed to the optimal weight vector $\beta^*$ for any $0 \leq \eta \leq \epsilon_k$.

To shorten the equations, we abbreviate $\hat{\beta}(\lambda, \eta)$ as $\hat{\beta}$.

We also use standard assumptions as for nominal LASSO problem as in Raskutti et al. (2010):

(a) $\lambda = \|\beta^*\|_1$, S is the support of $\beta^*$ and $|S| = p$.

(b) $X \in RE(S, 1, \gamma)$ for some $\gamma > 0$.

Since $\|\Delta x\|_2 \leq \|\Delta\|_2 \|x\|_2 \leq \epsilon_k \|x\|_2$, we have:

$$\begin{aligned}
\|y - X\hat{\beta}\|_2 &= \|y - \hat{X}\hat{\beta} + \Delta\hat{\beta}\|_2 \\
&\leq \|y - \hat{X}\hat{\beta}\|_2 + \|\Delta\hat{\beta}\|_2 \\
&\leq \|y - \hat{X}\hat{\beta}\|_2 + \epsilon_k\|\hat{\beta}\|_2 \\
&= \Big(\|y - \hat{X}\hat{\beta}\|_2 + \eta\|\hat{\beta}\|_2\Big) + (\epsilon_k - \eta)\|\hat{\beta}\|_2 \\
&\leq \Big(\|y - \hat{X}\bar{\beta}\|_2 + \eta\|\bar{\beta}\|_2\Big) + (\epsilon_k - \eta)\|\hat{\beta}\|_2 \\
&= \big(\|y - X\bar{\beta} - \Delta\bar{\beta}\|_2 + \eta\|\bar{\beta}\|_2\big) + (\epsilon_k - \eta)\|\hat{\beta}\|_2 \\
&\leq \big(\|y - X\bar{\beta}\|_2 + \|\Delta\bar{\beta}\|_2 + \eta\|\bar{\beta}\|_2\big) + (\epsilon_k - \eta)\|\hat{\beta}\|_2 \\
&\leq \|y - X\bar{\beta}\|_2 + (\epsilon_k + \eta)\|\bar{\beta}\|_2 + (\epsilon_k - \eta)\|\hat{\beta}\|_2 \\
&\leq \|y - X\beta^*\|_2 + (\epsilon_k + \eta)\|\bar{\beta}\|_2 + (\epsilon_k - \eta)\|\hat{\beta}\|_2
\end{aligned}$$

In addition, $\|\bar{\beta}\|_2 \leq \|\bar{\beta}\|_1 \leq \lambda$, $\|\hat{\beta}\|_2 \leq \|\hat{\beta}\|_1 \leq \lambda$, so:

$$\|y - X\hat{\beta}\|_2 \leq \|y - X\beta^*\|_2 + 2\epsilon_k\lambda \tag{4.3.3}$$

Now let $\zeta = \beta^* - \hat{\beta}$, we can write $y = X\beta^* + w$ as $y - X\hat{\beta} = X\zeta + w$, therefore (4.3.3) implies that:

$$\begin{aligned}
& \|X\zeta + w\|_2 \leq \|w\|_2 + 2\epsilon_k\lambda \\
\Rightarrow\ & \|X\zeta + w\|_2^2 \leq (\|w\|_2 + 2\epsilon_k\lambda)^2 \\
\Rightarrow\ & \|X\zeta\|_2^2 \leq 4\epsilon_k\lambda\|w\|_2 + 4\epsilon_k^2\lambda^2 - 2\zeta^T X^T w \\
\Rightarrow\ & \frac{1}{n}\|X\zeta\|_2^2 \leq \frac{4\epsilon_k\lambda}{n}\|w\|_2 + \frac{4\epsilon_k^2\lambda^2}{n} + 2\|\zeta\|_1 \left\|\frac{X^T w}{n}\right\|_\infty
\end{aligned}$$

Using assumption (a), we obtain:

$$\begin{aligned}
&\|\hat{\beta}_{S^C}\|_1 + \|\hat{\beta}_S\|_1 = \|\hat{\beta}\|_1 \le \lambda = \|\beta^*\|_1 = \|\beta^*_S\|_1 \\
\Rightarrow\ &\|\hat{\beta}_{S^C}\|_1 \le \|\beta^*_S\|_1 - \|\hat{\beta}_S\|_1 \le \|\beta^*_S - \hat{\beta}_S\|_1 \\
\Rightarrow\ &\|\beta^*_{S^C} - \hat{\beta}_{S^C}\|_1 \le \|\beta^*_S - \hat{\beta}_S\|_1 \\
\Rightarrow\ &\|\zeta_{S^C}\|_1 \le \|\zeta_S\|_1 \Rightarrow \zeta \in \mathcal{C}(S,1) \text{ and} \\
&\|\zeta\|_1 \le 2\|\zeta_S\|_1 \le 2\sqrt{p}\|\zeta_S\|_2 \le 2\sqrt{p}\|\zeta\|_2
\end{aligned}$$

Using assumption (b) $X \in RE(S,1,\gamma)$ and the fact that $\zeta \in \mathcal{C}(S,1)$, we have $\frac{1}{n}\|X\zeta\|_2^2 \ge \gamma^2\|\zeta\|_2^2$, therefore:

$$\gamma^2\|\zeta\|_2^2 \le \frac{4\epsilon_k\lambda}{n}\|w\|_2 + \frac{4\epsilon_k^2\lambda^2}{n} + 4\sqrt{p}\|\zeta\|_2\left\|\frac{X^T w}{n}\right\|_\infty \qquad (4.3.4)$$

**Lemma 4** *(Wainwright (2010)) Suppose $X$ is bounded by $L$, i.e. $|X_{ij}| \le L$, $w \sim \mathcal{N}(0,\sigma^2 I_{n\times n})$ then with high probability we have:*

$$\left\|\frac{X^T w}{n}\right\|_\infty \le L\sqrt{\frac{3\sigma^2\log d}{n}}$$

**Lemma 5** *Suppose $w \sim \mathcal{N}(0,\sigma^2 I_{n\times n})$ and $c > 1$, then with probability at least $1-e^{-\frac{3}{16}n(c-1)^2}$ we have:*

$$\|w\|_2 \le \sqrt{\sigma c n}$$

**Proof:** Since $w \sim \mathcal{N}(0,\sigma^2 I_{n\times n})$, $Z = \sum_{i=1}^n (w_i/\sigma)^2 \sim \chi^2_n$, using the tail bound for Chi-square random variable by Johnstone (2000), for any $\rho > 0$ we have:

$$P\left[|Z-n| \ge n\rho\right] \le \exp\left(-\frac{3}{16}n\rho^2\right),$$

Thus, by setting $\rho = c-1$, with probability at least $1-e^{-\frac{3}{16}n(c-1)^2}$ we have $Z \le cn$, i.e. $\|w\|_2 \le \sqrt{\sigma c n}$.

$\square$

Assuming $L, \sigma, \lambda$ are constants, using (4.3.4) and the results of Lemma 1 and 2, with high probability we have:

$$\|\zeta\|_2^2 \leq O\left(\frac{\epsilon_k}{\sqrt{n}}\right) + O\left(\frac{\epsilon_k^2}{n}\right) + O\left(\sqrt{\frac{p \log d}{n}}\right) \|\zeta\|_2$$

Solving this inequality we obtain the upper bound on $\|\beta^* - \hat{\beta}\|_2 = \|\zeta\|_2$ (with high probability) as following:

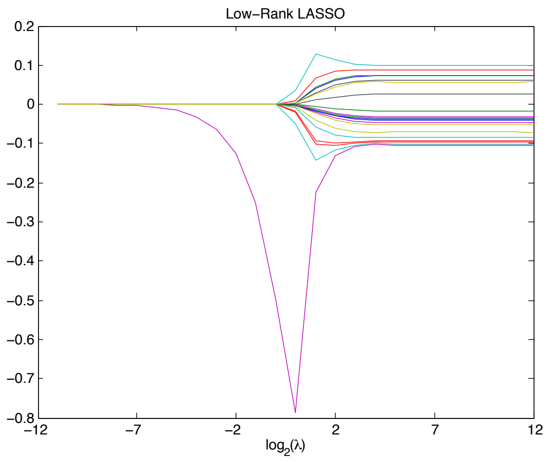
$$\|\beta^* - \hat{\beta}(\lambda, \eta)\|_2 \leq O\left(\sqrt{\frac{p \log d}{n} + \frac{\epsilon_k}{\sqrt{n}} + \frac{\epsilon_k^2}{n}}\right)$$

The above result shows that when the error made by replacing the feature matrix with its low-rank approximation (LR-LASSO) is small enough compared to $\sqrt{n}$ (i.e. $\epsilon_k << \sqrt{n}$), the corresponding estimator $\hat{\beta}(\lambda, 0)$ (for LR-LASSO) is closed to the optimal solution $\beta^*$. The solution of the robust counterpart of LASSO (RLR-LASSO) which is $\hat{\beta}(\lambda, \epsilon_k)$, is also closed to the optimal solution $\beta^*$.

### 4.3.2 Discussion

One might ask what happen if we just use the low-rank approximation matrix directly. We will show that we might end up with unexpected solutions if we do so. Indeed, we consider the rank 1 LASSO problem in which the data matrix $X$ is approximated with rank-1 matrix $\hat{X} = uv^T$ for some $u \in \mathbf{R}^n, v \in \mathbf{R}^d$: $\min_{\|\beta\|_1 \leq \lambda} \|y - uv^T\beta\|_2$. We randomly generate two vector $u \in \mathbf{R}^{20}, v \in \mathbf{R}^{20}$ and solve the rank-1 LASSO problem for $\lambda = 2^{-12}, \ldots, 2^{12}$, figure 4.1 shows the sparsity pattern of the solution of rank-1 LASSO vs robust rank-1 LASSO problem.

We can also explain analytically the reason why solution of non-robust rank-1 LASSO



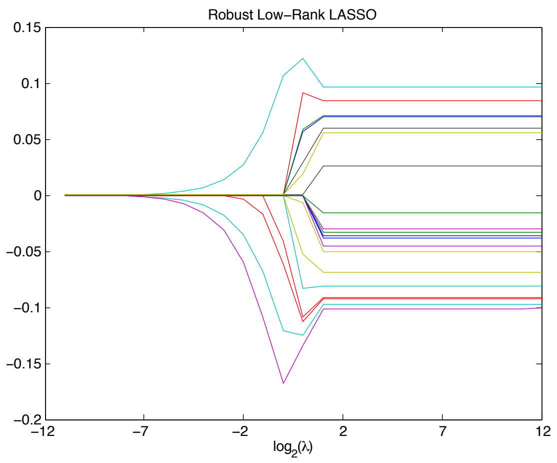


Figure 4.1. Rank-1 LASSO (left) and Robust Rank-1 LASSO (right) with random data. The plot shows the elements of the optimizer as a function of the $l_1$-norm penalty parameter $\lambda$. The non-robust solution has cardinality 1 or 0 for all of $0 < \lambda < C$ for some C. The robust version allows a much better control of the sparsity of the solution as a function of $\lambda$.

misbehaves for small value of $\lambda$ as follows: suppose $\hat{X} = pq^T$, $p \in R^n, q \in R^d$, the rank-1 LASSO problem is: $\phi(\lambda) = \min_{\|\beta\|_1 \le \lambda} \|pq^T\beta - y\|_2$

We can always normalize $p, q, y$ so that $\|p\|_2 = \|q\|_2 = \|y\|_2 = 1$.

$$\begin{aligned}
\phi(\lambda)^2 &= \min_{\|\beta\|_1 \le \lambda} \|pq^T\beta - y\|_2^2 = \min_{\|\beta\|_1 \le \lambda; t = q^T\beta} \|tp - y\|_2^2 \\
&= \max_{\mu \ge 0, \gamma} \min_{t, \beta} \|tp - y\|_2^2 + \mu(\|\beta\|_1 - \lambda) + \gamma(q^T\beta - t) \text{ (by taking the Lagrangian)} \\
&= \max_{\mu \ge 0, \gamma} \left[ -\lambda\mu + \min_t \left(t^2 - (2p^Ty + \gamma)t + 1\right) + \min_\beta \gamma q^T\beta + \mu\|\beta\|_1 \right] \\
&= \max_{|\gamma|\|q\|_\infty \le \mu} -\lambda\mu + 1 - (p^Ty + \gamma/2)^2 = 1 - \min_\gamma \lambda\|q\|_\infty|\gamma| + (p^Ty + \gamma/2)^2
\end{aligned}$$

The problem is one-dimensional and we can obtain the exact solutions:

$$\gamma^* = -2\mathbf{sign}(p^Ty)\max(0, |p^Ty| - \lambda\|q\|_\infty), \mu^* = |\gamma^*|\|q\|_\infty$$

From the Lagrangian and Slater condition, $\beta^*$ has to satisfy: $\gamma^* q^T\beta^* = |\gamma^*|\|q\|_\infty\|\beta^*\|_1$. Therefore $\forall\lambda < |p^Ty|/\|q\|_\infty$, we have $\gamma^* = 0$ so $\beta_i^* = 0$ whenever $|q_i| < \|q\|_\infty$.

### 4.3.3 Regularized Robust Rank-1 LASSO

In this section, we consider a special case in which the rank of the approximated matrix $\hat{X}$ is exactly 1. Note that the constrained LASSO and regularized LASSO are related - via Lagrangian. Similar to constrained LASSO problem, we can also formulate the robust regularized LASSO as following:

$$\min_{w} \|y - \hat{X}^T\beta\|_2 + \epsilon\|\beta\|_2 + \lambda\|\beta\|_1$$

When $\hat{X}$ is exactly rank-one, regularized robust LASSO problem becomes very easy - one-dimensional. Suppose that $\hat{X}^T = pq^T$ for some $p \in \mathbf{R}^n, q \in \mathbf{R}^d$, the robust counterpart is expresses as:

$$\psi_{\epsilon,\lambda}(p,q) = \min_{\beta} \ \|qp^T\beta - y\|_2 + \epsilon\|\beta\|_2 + \lambda\|\beta\|_1. \tag{4.3.5}$$

We can always normalize $p, q$ so that $\|p\|_2 = \|q\|_2 = 1$. Indeed, upon replacing $\lambda$ (resp. $\epsilon$) by $\|p\|_2\|q\|_2\lambda$ (resp. $\|p\|_2\|q\|_2\epsilon$), we are back to the normalized case $\|p\|_2 = \|q\|_2 = 1$. Finally, we can divide both values $\lambda, \epsilon$ by $\|y\|_2$ to reduce further our problem to one with $\|y\|_2 = 1$. We have:

$$\psi_{\epsilon,\lambda}(p,q) = \min_{\beta,\, t=\beta^T p} \ \|tq - y\|_2 + \epsilon\|\beta\|_2 + \lambda\|\beta\|_1$$

Using Lagrangian with multiplier $\mu$, we obtain

$$\psi_{\epsilon,\lambda}(p,q) = \max_{\mu} \left\{ \left(\min_{t} \ \|tq - y\|_2 + \mu t\right) + \left(\min_{\beta} \ \epsilon\|w\|_2 + \lambda\|\beta\|_1 - \mu p^T\beta\right)\right\}$$

Let us define $A(\mu) := \min_t \ \|tq - y\|_2 + \mu t$ and $B(\mu) := \min_\beta \ \epsilon\|w\|_2 + \lambda\|\beta\|_1 - \mu p^T\beta$.

The first term, $A(\mu)$, is $-\infty$ when $|\mu| > 1$. Otherwise, it can be expressed in terms of

$c := q^T y \in [-1,1]$ and $s = \sqrt{1-c^2} \geq 0$ as follows:

$$\begin{aligned} A(\mu) &= \min_t \|tq - y\|_2 + \mu t = \min_t \sqrt{t^2 - 2tc + 1} + \mu t \\ &= \min_t \sqrt{(t-c)^2 + s^2} + \mu t = \mu c + \min_\xi \sqrt{\xi^2 + s^2} - \mu \xi \ [\xi = t - c] \\ &= \mu c + s\sqrt{1-\mu^2} \ [t^* = c + (\mu s)/\sqrt{1-\mu^2}.] \end{aligned}$$

The second term, $B(\mu)$, writes

$$\begin{aligned} B(\mu) &= \min_\beta \epsilon \|\beta\|_2 + \lambda \|\beta\|_1 - \mu p^T \beta \\ &= \max_{v,r} 0 \ : \ \mu p = v + r, \ \ \|v\|_\infty \leq \lambda, \ \ \|r\|_2 \leq \epsilon. \end{aligned}$$

The quantity $B(\mu)$ is finite (and indeed, zero) if and only if there exist $v$ such that

$$\|v\|_\infty \leq \lambda, \ \ \|v - \mu p\|_2 \leq \epsilon.$$

Hence, our problem becomes:

$$\max_{\mu,v} \mu c + s\sqrt{1-\mu^2} \ : \ |\mu| \leq 1, \ \ \|v\|_\infty \leq \lambda, \ \ \|v - \mu p\|_2 \leq \epsilon.$$

Note that the function $F(\mu) := \min_v \|v - \mu p\|_2 \ : \ \|v\|_\infty \leq \lambda$ has an optimal point is $v^* = \min(\lambda, \max(-\lambda, \mu p))$ in which the corresponding value is $F(\mu) = \|(|\mu p| - \lambda \mathbf{1})_+\|_2$. Therefore, our problem is reduced to a one-dimensional problem:

$$\max_\mu \mu c + s\sqrt{1-\mu^2} \ : \ |\mu| \leq 1, \ \ \|(|\mu p| - \lambda \mathbf{1})_+\|_2 \leq \epsilon.$$

The optimal $\mu$ is of the same sign as $c$, and the problem is equivalent to

$$\max_\mu \mu |c| + s\sqrt{1-\mu^2} \ : \ 0 \leq \mu \leq 1, \ \ \|(\mu |p| - \lambda \mathbf{1})_+\|_2 \leq \epsilon,$$

which can be easily solved by bisection in $O(n)$. Note that the dependence on $q$ is only via $c = q^T y$, $s = \sqrt{1-c^2}$. The cost of computing $c$ is $O(m)$, so the robust counterpart (4.3.5) can be solved in $O(n+m)$.

When $\epsilon = 0$, we recover an ordinary LASSO problem with rank-one matrix:

$$\max_{\mu} \ \mu|c| + s\sqrt{1-\mu^2} \ : \ 0 \le \mu \le 1, \ \ \mu\|p\|_\infty \le \lambda.$$

## 4.4 Experimental Results

The purpose of our experiments is to compare the performance of LASSO, LR-LASSO and RLR-LASSO in two different tasks: multi-label classification (4.1) and word associations for multiple queries (4.2).

For models using low-rank approximation (LR-LASSO and RLR-LASSO), we first run experiments computing low-rank approximation matrices with different values of $k$ using Fast randomized subsampling algorithm. The low-rank dimension $k$ is ranging from 5 to 50, and $l$ is set to $k+10$. Figure 1 shows the running time of computing the low-rank approximation for different datasets as $k$ varied; the ratio $\sigma_{k+1}/\sigma_1$ is also reported. As can be seen from the table, low-rank approximation can be computed quickly even for large datasets with million of samples and hundreds of thousands of features. The $(k+1)^{th}$ singular value is also small compared to the largest singular value of the data matrix.

For the experiments presented in this section, we chose $\epsilon = 10^{-6}, \gamma = 10^{-4}, \sigma_1 = 0.1, \sigma_2 = 0.9, M = 10, \alpha_{\min} = 10^{-3}, \alpha_{\max} = 10^{3}, \sigma_0 = 1$ as suggested by Birgin et al. (2001). The regularization parameter $\lambda$ is tuned from $\{10^i : i = 3, 4, 5, 6\}$ using five-fold cross-validation on the training data (with F1-score for each *one-vs-all* sub-classification instance).

All experiments are conducted on a personal workstation with 16GB RAM and 2.6GHz quad-core Intel processor.

### 4.4.1 Datasets

**RCV1-V2**: Reuters Corpus Volume 1 - Version 2 is a large-scale dataset for text classification task that is based on the well known benchmark dataset for text classification, the Reuters (RCV1) dataset. In our experiments, we use the full topics set containing 804,414 news articles, each article is assigned to a subset of 101 topics. We use the pre-processed data that prepared by Lewis et al. (2004), which splits the data into 23,149 training documents and 781,265 test documents. The number of features in this dataset is $46,236$ and the density of the data is 0.031.

**TMC2007**: Text Mining Competition dataset is based on the competition organized by the text mining workshop of the $7^{Th}$ SIAM international conference on data mining. It contains 28,596 aviation safety reports in free text form, each annotated with a subset of the 22 problem types that appear during certain flights. TMC2007 has 49,060 features and density of 0.098.

**PUBMED, NYTimes Datasets**: we use two largest UCI Bag-of-word datasets [1]: NYTimes news articles dataset which contains 300,000 documents, 102,660 words and approximately 100,000,000 non-zero terms resulting in a file of size 1GB; and PUBMED abstracts dataset which contains 8,200,000 documents, 141,043 words and approximately 730,000,000 non-zero terms giving a file size of 7.8GB.

Figure 4.2 shows the top 5000 singular values for each data, scaled by the maximum singular value - as can be seen from the plot, the singular values decay quickly thus, we are safe to say that the corresponding data matrices for the above datasets are approximately low-rank. We believe that such phenomenon is also true for most text datasets.

[1]http://archive.ics.uci.edu/ml/datasets/Bag+of+Words

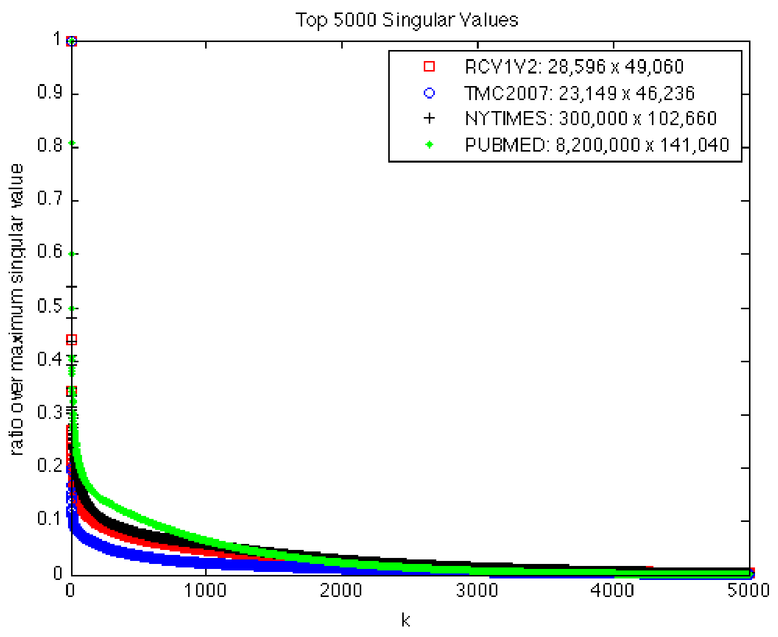
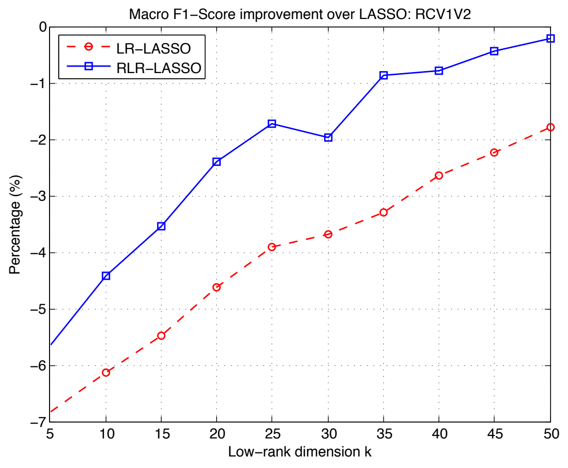


Figure 4.2. Top 5000 singular values for each dataset, scaled by maximum singular value. The size of each dataset (number of samples x number of features) is also shown.

### 4.4.2 Multi-label classification

Since we treat each label as a single classification subproblem (*one-against-all* method), we evaluate the performance of all three models using the **Macro-F1 measure**. The Macro-F1 measure treats every label equally and calculates the global measure as the means of the local measures of all labels: **Macro-F1** $= \frac{2\bar{R}\bar{P}}{\bar{R}+\bar{P}}$ where $\bar{R} = \frac{1}{L}\sum_{l=1}^{L} R_i, \bar{P} = \frac{1}{L}\sum_{l=1}^{L} P_i$ are the average recall and precision across all labels. We will also report the average running time of each model.

Figure 4.3 and 4.4.2 show the average training time speedup for LR-LASSO, RLR-LASSO compared to LASSO on all labels (RCV1V2 has 101 labels, TMC2007 has 22 labels) over different runs for each value of regularization parameter $\lambda \in \{10^i : i = 3, 4, 5, 6\}$, as the low-rank dimension $k$ varies (one run for each value of the regularization parameter $\lambda$). LR-LASSO and RLR-LASSO run much faster than the nominal LASSO for both datasets,



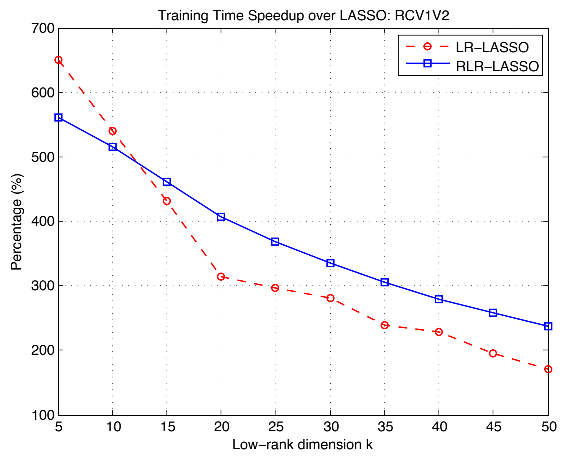


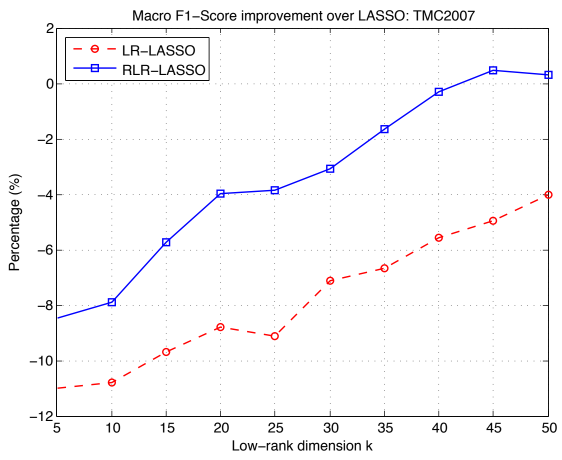
Figure 4.3. Multi-label classification: RCV1V2 dataset

around 3-6 times in magnitude. We also observe that RLR-LASSO run slightly faster than LR-LASSO. Perhaps the $l_2$ regularization term in the objective function of RLR-LASSO model helps SPG algorithm converges faster than LR-LASSO.

As can also be seen from Figure 4.3 and 4.4.2, as $k$ increases, the performances of low-rank approximation models improve gradually. Although LR-LASSO does not perform anywhere close to LASSO, its robust model RLR-LASSO has almost the same Macro F1-Score as the nominal LASSO with a large enough $k$.

### 4.4.3 When labels are columns of data matrix

For many applications - especially in text classification such as in Gawalt et al. (2010), the goal is to learn a short list of words that are predictive for a given query term in a text dataset. LASSO can be used to produce such list of predictive words. To be manageable by a human reader, the list should be very short w.r.t the number of features, thus we solve the LASSO problem with different values of $\lambda$ and choose the one that generates enough non-zeros in the solution.



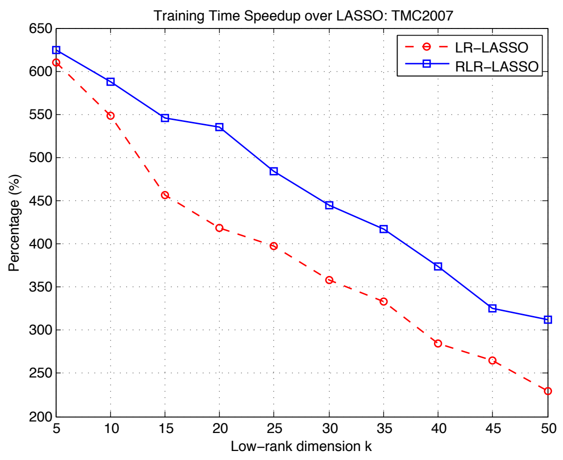


Figure 4.4. Multi-label classification: TMC2007 dataset

Given the original data matrix $X$, for each query term we need to solve a LASSO problem in which the response vector $y$ is a column vector of $X$ and the corresponding feature matrix is obtained by removing that column from $X$. The low-rank approximation for new data matrix given the low-rank approximation of the original matrix can be derived quickly. Suppose that $X = UV^T$ where $U \in \mathbf{R}^{n\times k}, V \in \mathbf{R}^{d\times k}$. Let's $X^{(i)} \in \mathbf{R}^{n\times(d-1)}$ be the matrix obtained from $X$ by removing the $i^{th}$ column, $X_{-i} \in \mathbf{R}^{n\times d}$ be the matrix obtained from $X$ by replacing its $i^{th}$ column with zeroes. Let $c_i$ be the $i^{th}$ column of $X$, $e_i$ is the $i^{th}$ unit vector in $\mathbf{R}^d$, we can write $X_{-i}$ as:

$$X_{-i} = X - c_i e_i^T = UV^T - c_i e_i^T = [U, -c_i][V, e_i]^T$$

Let $U^{(i)} = [U, -c_i]$ and $V^{(i)}$ be the matrix obtained from $[V, e_i]$ by removing its $i^{th}$ row. We can then write $X^{(i)} = U^{(i)}(V^{(i)})^T$, therefore the rank of $X^{(i)}$ can be bounded as following:

$$\begin{aligned}\mathbf{rank}(X^{(i)}) &\leq \min\{\mathbf{rank}(U^{(i)}), \mathbf{rank}(V^{(i)})\}\\ &\leq \min\{\mathbf{rank}(U)+1, \mathbf{rank}(V)+1\}\\ &\leq k+1\end{aligned}$$

| **automotive** | **agriculture** | **technology** | **tourism** | **aerospace** |
|---|---|---|---|---|
| car | government | company | tourist | boeing |
| vehicle | farm | computer | hotel | aircraft |
| auto | farmer | system | business | space |
| sales | food | web | visitor | program |
| model | water | information | economy | jet |
| driver | trade | internet | travel | plane |
| ford | land | american | tour | nasa |
| driving | crop | job | local | flight |
| engine | economic | product | room | airbus |
| consumer | country | software | plan | military |
| **defence** | **financial** | **healthcare** | **petroleum** | **gaming** |
| afghanistan | company | health | oil | game |
| attack | million | care | prices | gambling |
| forces | stock | cost | gas | casino |
| military | market | patient | fuel | player |
| gulf | money | corp | company | online |
| troop | business | al_gore | barrel | computer |
| aircraft | firm | doctor | gasoline | tribe |
| terrorist | fund | drug | bush | money |
| president | investment | medical | energy | playstation |
| war | economy | insurance | opec | video |

Table 4.1. **NYTimes news articles** - top 10 predictive words for different query terms (10 industry sectors). Low-rank approximation with $k = 20$ is used, total training time (for all query terms) is 42918 seconds

We could also compute $rank - k$ decomposition of $X^{(i)}$ with slightly more complicated derivation as presented in (Brand (2006)).

We experiment with NYTimes and PubMed datasets, both of these datasets are so large that it's impossible to run nominal LASSO on a personal workstation. Thus, our experiment's goal is to find the top predictive words for each given query term. In Table 4.1 and Table 4.2, we report the top 10 predictive words for given lists of 10 query terms, produced by RLR-LASSO with $k = 20$.

| **arthritis** | **asthma** | **cancer** | **depression** | **diabetes** |
|---|---|---|---|---|
| joint | bronchial | tumor | effect | diabetic |
| synovial | asthmatic | treatment | treatment | insulin |
| infection | children | carcinoma | disorder | level |
| chronic | respiratory | cell | depressed | glucose |
| pain | symptom | chemotherapy | pressure | control |
| treatment | allergic | survival | anxiety | plasma |
| fluid | infant | risk | symptom | diet |
| knee | inhalation | dna | drug | liver |
| acute | airway | malignant | neuron | renal |
| therapy | fev1 | diagnosis | response | normal |
| **gastritis** | **hiv** | **leukemia** | **migraines** | **parkinson** |
| gastric | aid | cell | headache | treatment |
| h_pylori | infection | acute | headaches | effect |
| chronic | cell | bone_marrow | pain | nerve |
| ulcer | hiv-1 | leukemic | disorder | syndrome |
| acid | infected | tumor | women | disorder |
| stomach | antibodies | remission | chronic | neuron |
| atrophic | risk | t_cell | duration | receptor |
| antral | positive | antigen | symptom | alzheimer |
| reflux | transmission | chemotherapy | gene | response |
| treatment | drug | expression | therapy | brain |

Table 4.2. **PUBMED abstracts** - top 10 predictive words for different query terms (10 diseases). Low-rank approximation with $k = 20$ is used, total training time (for all query terms) is 56352 seconds

# Chapter 5

# Conclusion

In this dissertation, we introduced new learning models that are capable of solving large scale machine learning problems using data approximation under robust optimization perspective. We have showed that data approximation under robust scheme is efficient and scalable and more reliable than just applying the approximation directly.

In Chapter 3, we have described data thresholding technique for large-scale sparse linear classification and provided theoretical bound on the amount of thresholding level needed to obtain desired performance. The proposed method is a promising pro-processing method that can be applied in any learning algorithm to efficiently solve large-scale sparse linear classification problems, both in terms of required memory as well as the running time of the algorithm.

Next, in Chapter 4, we have described an efficient and scalable robust low-rank model for LASSO problem, in which the approximation error made by replacing the original data matrix with a low-rank approximation is taken into account in the model itself. The experimental results lead us to believe that the proposed model could make statistical learning that involves running multiple instances of LASSO feasible for extremely large datasets. Although we only focus on first-order methods in which the main computational effort is to compute the gradient, we believe that similar approach can be taken on how low-rank structures could help in speeding up second-order methods for solving LASSO problem. Another interesting direction we also wish to pursue is to construct robust low-rank models for other learning problems.

# Bibliography

Anthony, M. and Bartlett, P. L. (1999). *Neural Network Learning: Theoretical Foundations.* Cambridge University Press.

Bach, F. R. and Jordan, M. I. (2005). Predictive low-rank decomposition for kernel methods. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 33–40, New York, NY, USA. ACM.

Balakrishnan, S. and Madigan, D. (2008). Algorithms for sparse linear classifiers in the massive data setting. *J. Mach. Learn. Res.*, 9:313–337.

Becker, S., Bobin, J., and Candès, E. J. (2011). Nesta: A fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Sciences*, 4(1):1–39.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization (Princeton Series in Applied Mathematics).* Princeton University Press.

Bhattacharyya, C. (2004). Robust classification of noisy data using second order cone programming approach. In *Intelligent Sensing and Information Processing, 2004. Proceedings of International Conference on*, pages 433 – 438.

Bhattachryya, S., Grate, L., Mian, S., Ghaoui, L. E., and Jordan, M. (2003). Robust sparse hyperplane classifiers: application to uncertain molecular profiling data. *Journal of Computational Biology*, 11(6):1073–1089.

Bickel, P. J., Ritov, Y., and Tsybakov, A. B. (2008). Simultaneous analysis of lasso and dantzig selector.

Birgin, E. G., Martínez, J. M., and Raydan, M. (2000). Nonmonotone spectral projected gradient methods on convex sets. *SIAM Journal on Optimization*, pages 1196–1211.

Birgin, E. G., Martínez, J. M., and Raydan, M. (2001). *ACM Trans. Math. Softw.*, 27(3):340–349.

Birgin, E. G., Martinez, J. M., and Raydan, M. (2003). Inexact spectral projected gradient methods on convex sets. *IMA Journal on Numerical Analysis*, 23:539–559.

Bordes, A. and Bottou, L. (2005). The huller: a simple and efficient online svm. In *In Machine Learning: ECML 2005, Lecture Notes in Artificial Intelligence, LNAI 3720*, pages 505–512. Springer Verlag.

Brand, M. (2006). Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and Its Applications*, 415(1):20–30.

Caramanis, C. and Mannor, S. (2008). Learning in the limit with adversarial disturbances. In *COLT*, pages 467–478.

Cohen, A., Dahmen, W., and Devore, R. (2009). Compressed sensing and best k-term approximation. *J. Amer. Math. Soc*, pages 211–231.

Demmel, J. W. and Heath, M. T. (1997). Applied numerical linear algebra. In *Society for Industrial and Applied Mathematics*. SIAM.

Duchi, J., Shalev-Shwartz, S., Singer, Y., and Chandra, T. (2008). Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 272–279, New York, NY, USA. ACM.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, 32(2):407–451.

Figueiredo, M. A. T., Nowak, R., and Wright, S. (2007). Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):586–597.

Fine, S. and Scheinberg, K. (2002). Efficient svm training using low-rank kernel representations. *J. Mach. Learn. Res.*, 2:243–264.

Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004). Spectral grouping using the nystrom method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225.

Friedman, J. H., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

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

Gawalt, B., Jia, J., Miratrix, L., El Ghaoui, L., Yu, B., and Clavier, S. (2010). Discovering word associations in news media via feature selection and sparse classification. In *Proceedings of the international conference on Multimedia information retrieval*, MIR '10, pages 211–220, New York, NY, USA. ACM.

Genkin, A., Lewis, D. D., and Madigan, D. (August 2007). Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49:291–304(14).

Ghaoui, L. E. and Lebret, H. (1997). Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.*, 18(4):1035–1064.

Globerson, A. and Roweis, S. (2006). Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 353–360, New York, NY, USA. ACM.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA.

Gu, M. and Eisenstat, S. C. (1996). Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM J. Sci. Comput.*, 17(4):848–869.

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

Johnstone, I. (2000). *Chi-square Oracle Inequalities*. Mathematics research report. Department of Statistics, Stanford University.

Koh, K., Kim, S.-J., and Boyd, S. (2007). An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, 8:1519–1555.

Lewis, D. D., Yang, Y., Rose, Y. G., Li, F., and Dietterich, G. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397.

Loosli, G., Canu, S., and Bottou, L. (2005). Training invariant support vector machines using selective sampling.

Mangasarian, O. L. (2006). Exact 1-norm support vector machines via unconstrained convex differentiable minimization. *J. Mach. Learn. Res.*, 7:1517–1530.

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

Osborne, M., Presnell, B., and Turlach, B. (2000). A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389.

Osuna, E., Freund, R., and Girosi, F. (1997). An improved training algorithm for support vector machines. pages 276–285. IEEE.

Perkins, S., Lacker, K., and Theiler, J. (2003). Grafting: fast, incremental feature selection by gradient descent in function space. *J. Mach. Learn. Res.*, 3:1333–1356.

Raskutti, G., Wainwright, M. J., and Yu, B. (2010). Restricted eigenvalue properties for correlated gaussian designs. *J. Mach. Learn. Res.*, 99:2241–2259.

Schmidt, M., van den Berg, E., Friedlander, M. P., and Murphy, K. (2009). Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. In *Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS) 2009*, volume 5, pages 456–463, Clearwater Beach, Florida.

Shi, J., Yin, W., Osher, S., and Sajda, P. (2010). A fast hybrid algorithm for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, 11:713–741.

Shivaswamy, P. K., Bhattacharyya, C., and Smola, A. J. (2006). Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314.

Talwalkar, A., Kumar, S., and Rowley, H. A. (2008). Large-scale manifold learning. In *Computer Vision and Pattern Recognition*.

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

Tikhonov, A. N. and Arsenin, V. Y. (1977). *Solutions of Ill-Posed Problems*. V. H. Winston & Sons, Washington, D.C.: John Wiley & Sons, New York,.

Trafalis, T. B. and Gilbert, R. C. (2007). Robust support vector machines for classification and computational issues. *Optimization Methods Software*, 22:187–198.

Tseng, P. and Yun, S. (2008). A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program.*, 117(1):387–423.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13.

van den Berg, E., Schmidt, M., Friedlander, M. P., and Murphy, K. (2008). Group sparsity via linear-time projection.

Wainwright, M. (2010). High-dimensional statistics: Some progress and challenges ahead. In *Winedale Workshop*.

Xu, H., Caramanis, C., and Mannor, S. (2011). *Robust Optimization in Machine Learning.* Book Chapter in Optimization for Machine Learning. MIT Press.

Zhang, K., Lan, L., Liu, J., Rauber, A., and Moerchen, F. (2012). Inductive kernel low-rank decomposition with priors: A generalized nystrom method. *CoRR*, abs/1206.4619.