Title
Low-Dimensional Models for PCA and Regression

Author
Omidiran, Christian Ladapo

Publication Date
2013

Peer reviewed|Thesis/dissertation

Low-Dimensional Models for PCA and Regression

by

Christian Ladapo Omidiran

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

in

Engineering - Electrical Engineering and Computer Sciences

in the

Graduate Division
of the
University of California, Berkeley

Committee in charge:
Professor Laurent El Ghaoui, Co-Chair
Professor Martin Wainwright, Co-Chair
Professor Sandrine Dudoit

Spring 2013

Low-Dimensional Models for PCA and Regression

Abstract

Low-Dimensional Models for PCA and Regression

by

Christian Ladapo Omidiran
Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Laurent El Ghaoui, Co-Chair
Professor Martin Wainwright, Co-Chair

This thesis examines two separate statistical problems for which low-dimensional models are effective.

In the first part of this thesis, we examine the Robust Principal Components Analysis (RPCA) problem: given a matrix $\mathbf{X}$ that is the sum of a low-rank matrix $\mathbf{L}^*$ and a sparse noise matrix $\mathbf{S}^*$, recover $\mathbf{L}^*$ and $\mathbf{S}^*$. This problem appears in various settings, including image processing, computer vision, and graphical models. Various polynomial-time heuristics and algorithms have been proposed to solve this problem. We introduce a block coordinate descent algorithm for this problem and prove a convergence result. In addition, our iterative algorithm has low complexity per iteration and empirically performs well on synthetic datasets.

In the second part of this thesis, we examine a variant of ridge regression: unlike in the classical setting where we know that the parameter of interest lies near a single point, we instead only know that it lies near a known low-dimensional subspace. We formulate this regression problem as a convex optimization problem, and introduce an efficient block coordinate descent algorithm for solving it. We demonstrate that this "subspace prior" version of ridge regression is an appropriate model for understanding player effectiveness in basketball. In particular, we apply our algorithm to real-world data and demonstrate empirically that it produces a more accurate model of player effectiveness by showing that (1) the algorithm outperforms existing approaches and (2) it leads to a profitable betting strategy.

To my parents.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank Professor Laurent El Ghaoui for his support, mentorship, and for having faith in me even during very dark times when I stopped believing in myself. I am deeply grateful to Professor Martin Wainwright for his mentorship and research collaborations during my first few years at Berkeley. I am thankful that he agreed to serve on my thesis committee. I am thankful to Professor Sandrine Dudoit for serving on my qualifying exam and agreeing to serve again on my thesis committee. Thanks to Professor Kannan Ramchandran for serving on my qualifying exam.

I am very thankful to the professors at Berkeley, who have taught me a lot from classes, projects, and the conversations/interactions I have had with them. just conversions and interacting with them. I am especially grateful to Professors Jim Pitman and David Aldous for the Stat 205A/205B probability sequence. It was one of the most enjoyable courses I have taken in my life. Thanks to Professors Peter Bartlett, Laurent El Ghaoui, Michael Jordan, and Martin Wainwright for their courses in statistical machine learning and optimization. I have learned some very powerful tools over the past few years, and I plan to put them to very good use going forward.

Hari. I don't know what to say about this guy. He is one of the closest friends I have in life. Within my first two weeks at Berkeley, I found someone with whom I can discuss anything from graphical models to sports and pop culture. Finding a deep friend like him is one of the best things that has ever happened to me.

I am thankful to Sahand for his friendship and the conversations we have had over the past past few years about technical matters, sports, and life.

Thanks to Galen, Pulkit, Jiening, Anand, Bobak, Krish and many of the friendly older graduate students I met when I first came to Berkeley who helped smooth the transition. Thanks to the folks at the RSF. Graduate school is pretty stressful and depressing at times, and those I played pickup basketball with over the years have helped me stay balanced.

I am very thankful to Professors Richard Baraniuk and Don Johnson from Rice University. It is because of them that I got into research and thought seriously about applying to graduate school.

I met Professor Deborah Nolan in 2005 at the UCLA Summer Program in Statistics. I am very thankful for the conversations I had with her during that week-long experience, and that she encouraged me to apply to Berkeley for graduate school.

Thanks to the administrative staff at UCB. . . Shirley, Pat, Mary and Ruth for helping my navigate paperwork smoothly. I am especially thankful to Shiela Humphries for her mentorship and support over the past few years. She has always had my back, and for that I am eternally grateful. I am very grateful to the National Science Foundation, UC Berkeley, the EECS department, and Professors Laurent El Ghaoui and Martin Wainwright for their financial support over the past few years. Simply put, without you guys grad school would have not been feasible.

Finally, as clichéd as it may sound, thanks to the people of the United States in general and the people of Maryland, Texas and California more specifically. America

is truly the land of opportunity, in part because it allowed what was once an immigrant with a funny-sounding name and a strange accent to be able to pursue some of the best experiences this country has to offer.

Chapter 1

Introduction

In the first part of this thesis, we propose a fast algorithm for solving the Robust Principal Components Analysis problem: given a matrix $\mathbf{X}$ that is the sum of a low-rank matrix $\mathbf{L}^*$ and a sparse matrix $\mathbf{S}^*$, recover $\mathbf{L}^*$ and $\mathbf{S}^*$. This problem appears in various settings, including image processing [Torre and Black, 2003], computer vision [Ke and Kanade, 2005], and graphical models [Chandrasekaran et al., 2010]. Various polynomial-time heuristics and algorithms have been proposed to solve this problem under certain conditions. We introduce a block coordinate descent algorithm for this problem and prove that its limit points are also stationary points. In addition, this iterative algorithm has low complexity per iteration and performs well on synthetic datasets.

In the second part of this thesis, we develop a new penalized regression model for basketball, use cross-validation to select its tuning parameters, and then use it to produce ratings of player ability. We apply the model to the 2010-2011 NBA season to predict the outcome of games. We compare the performance of our procedure to other known regression techniques for this problem and demonstrate empirically that our model produces substantially better predictions. We evaluate the performance of our procedure against the Las Vegas gambling lines, and show that with a sufficiently large number of games to train on our model outperforms those lines. Finally, we demonstrate how the technique developed here can be used to quantitively identify "overrated" players who are less impactful than common wisdom might suggest.

Chapter 2

Robust Principal Components Analysis

2.1 Notation

Given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, we use the notation (a) $\text{rank}(\mathbf{X})$ to denote the rank of $\mathbf{X}$, (b) $||\mathbf{X}||_0$ to denote its number of non-zero entries, (c) $s_i(\mathbf{X})$ to denote the i^{th} largest singular value, (d) $\|\mathbf{X}\|_* := \sum_i s_i(\mathbf{X})$ to denote the nuclear norm of $\mathbf{X}$, (e) $\|\mathbf{X}\|_F$ to denote the Frobenius norm, and (f) $\|\mathbf{X}\|_1 := \sum_{ij} |\mathbf{X}_{ij}|$ to denote the element-wise 1-norm.

2.2 Introduction

Given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ that is the sum of a low rank matrix $\mathbf{L}^*$ and sparse matrix $\mathbf{S}^*$, we seek to recover both $\mathbf{L}^*$ and $\mathbf{S}^*$. This is the *Robust Principal Components Analysis* (RPCA) problem: as in Principal Components Analysis (PCA), we want to learn a low-rank matrix $\mathbf{L}^*$ that is contaminated by errors, represented by the matrix $\mathbf{S}^*$. Unlike the standard PCA problem, the corruption matrix $\mathbf{S}^*$ may contain gross, but sparse errors. This problem appears in a variety of settings including image processing [Torre and Black, 2003], computer vision [Ke and Kanade, 2005], graphical models [Chandrasekaran et al., 2010], traffic anomaly detection [Abdelkefi et al., 2010], astronomical spectroscopy [Budavari et al., 2009] and system identification [Chandrasekaran et al., 2011].

Assuming that the number of nonzeros s of $\mathbf{S}^*$ is known, RPCA is equivalent to solving the intractable optimization problem

$$\min_{\mathbf{L} \in \mathbb{R}^{m \times n}} \text{rank}(\mathbf{L}) \text{ subject to } ||\mathbf{X} - \mathbf{L}||_0 \leq s. \tag{2.1}$$

To address the computational intractability of (2.1), many researchers have proposed heuristics for performing this decomposition. Torre and Black [2003] proposes a technique inspired by robust statistics [Huber, 1974]. Ke and Kanade [2005] proposes directly solving a non-convex program similar to (2.1) using block coordinate descent. Unfortunately, there is no discussion of whether the algorithm provably converges, nor of what it converges to. The work of Ke and Kanade [2005] is similar in spirit to our own: the technique we introduce can be viewed as a provably convergent alternative to their algorithm.

Candès et al. [2011] (see also Chandrasekaran et al. [2010]) proposes a convex relaxation to (2.1) called Principal Components Pursuit (PCP) that successfully decomposes $\mathbf{X}$ when certain conditions on $\mathbf{L}^*$ and $\mathbf{S}^*$ are satisfied. The PCP relaxation is the convex program

$$\min_{\mathbf{L}\in\mathbb{R}^{m\times n},\mathbf{S}\in\mathbb{R}^{m\times n}} \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1 \text{ subject to } \mathbf{X} = \mathbf{L} + \mathbf{S}. \tag{2.2}$$

Candès et al. [2011] shows that when the true low-rank component $\mathbf{L}^*$ and true sparse component $\mathbf{S}^*$ satisfy certain technical conditions, the solution $(\hat{\mathbf{L}}, \hat{\mathbf{S}})$ to (2.2) is the true factorization of $\mathbf{X}$ with high probability.

Unfortunately, while (2.2) can be solved numerically by reformulating as a semidefinite program (SDP) [Vandenberghe and Boyd, 1996], the resulting program is solved with complexity $O\left((mn)^3\right)$ per iteration using standard interior point techniques (see Appendix A of Chandrasekaran et al. [2011] for this formulation). Lin et al. [2010] and Yuan and Yang [2009] develop an alternating direction method of multipliers (ADMM) algorithm (see Boyd et al. [2011] for an excellent modern overview of the ADMM technique) that brings the per iteration complexity down to the cost of a SVD, which is $O(m^3)$ under the assumption that $n = \Theta(m)$. In contrast, assuming that $n = \Theta(m)$ our algorithm has complexity $O(rm^2)$ per iteration, where r is the desired rank of the low-rank factorization. As a consequence, our approach is easier to scale to very large problems for which r is small relative to m.

We now introduce more precisely our algorithm, prove its convergence to a stationary point, and then present results evaluating its performance on synthetic data.

2.3 Block Coordinate Descent for Robust Principal Components Analysis

We consider for a given matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ that is the sum of a rank r matrix $\mathbf{L}^*$ and a sparse matrix $\mathbf{S}^*$ the problem

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_1, \tag{2.3}$$

where $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\mathbf{V} \in \mathbb{R}^{r \times n}$ are decision variables. We are ultimately interested in understanding (a) when iterative algorithms for the non-convex problem (2.3) converge, (b) and under what circumstances the solution $(\hat{\mathbf{U}}, \hat{\mathbf{V}})$ to (2.3) represents the true low rank matrix $\mathbf{L}^*$ (in other words, when $\mathbf{L}^*$ equals $\hat{\mathbf{U}}\hat{\mathbf{V}}$).

One natural heuristic for solving (2.3) is to perform alternating minimization over $\mathbf{U}$ and $\mathbf{V}$, resulting in Algorithm 1.

Algorithm 1 KeKanade($\mathbf{X}, r, k$)

$\mathbf{U}^0 \leftarrow \mathbf{U}_{init}$
$\mathbf{V}^0 \leftarrow \mathbf{V}_{init}$
for $i \in \{1, 2, \ldots, k\}$ **do**
  $\mathbf{U}^i \leftarrow \arg\min_{\mathbf{U} \in \mathbb{R}^{m \times r}} \|\mathbf{X} - \mathbf{U}\mathbf{V}^{i-1}\|_1$
  $\mathbf{V}^i \leftarrow \arg\min_{\mathbf{V} \in \mathbb{R}^{r \times n}} \|\mathbf{X} - \mathbf{U}^i\mathbf{V}\|_1$
end for

This is the approach taken by Ke and Kanade [2005]. Algorithm 1 is parallelizable due to the separability of both the $\mathbf{U}^i$ and $\mathbf{V}^i$ computations and often performs well empirically on data. Unfortunately there are no known convergence results for Algorithm 1. In fact, it is known that the iterates obtained by performing alternating minimization in this manner sometime fail to converge [Powell, 1973]. Powell [1973] provides a concrete example where the iterates cycle infinitely.

Instead of directly minimizing (2.3), we solve the equivalent problem

$$\min_{\mathbf{U} \in \mathbb{R}^{m \times r}, \mathbf{V} \in \mathbb{R}^{r \times n}, \mathbf{X} \in \mathbb{R}^{m \times r}, \mathbf{Y} \in \mathbb{R}^{r \times n}} f(\mathbf{U}, \mathbf{V}, \mathbf{X}, \mathbf{Y}), \tag{2.4}$$

with

$$f(\mathbf{U}, \mathbf{V}, \mathbf{X}, \mathbf{Y}) := \|\mathbf{X} - \mathbf{U}\mathbf{V}\|_1 + ||\mathbf{U} - \mathbf{X}||_F^2 + ||\mathbf{V} - \mathbf{Y}||_F^2. \tag{2.5}$$

The formulation (2.4) is equivalent to (2.3) in the sense that an optimal solution for one can be used to construct an optimal solution for the other. We prove this in Appendix A.1.

For computational reasons, rather than directly optimizing the objective function with respect to the matrix variables $\mathbf{U}, \mathbf{V}$, we optimize the columns of $\mathbf{U}$, $\mathbf{V}^T$ separately. In other words, our proprosed algorithm is block coordinate descent with the columns of $\mathbf{U}, \mathbf{V}^T$ viewed as variables.

Let us use the notation $f_{U_q}(u)$ to denote (2.5) as a function of the q^{th} column of $\mathbf{U}$, with all other columns of $\mathbf{U}, \mathbf{V}^T$ fixed. Similarly, we use $f_{V_q}(v)$ to denote (2.5) as a function of the q^{th} column of $\mathbf{V}^T$.

We then have Algorithm 2, a block coordinate descent procedure.

Algorithm 2 BCD($\mathbf{X}, r, k$)

$\mathbf{U}^0 \leftarrow \mathbf{U}_{init}$
$\mathbf{V}^0 \leftarrow \mathbf{V}_{init}$
for $i \in \{1, 2, \ldots, k\}$ **do**
  for $q \in \{1, 2, \ldots, r\}$ **do**
    $\mathbf{U}_q^i \leftarrow \arg\min_{u \in \mathbb{R}^m} f_{U_q}(u)$
  end for
  for $q \in \{1, 2, \ldots, r\}$ **do**
    $\mathbf{V}_q^i \leftarrow \arg\min_{v \in \mathbb{R}^n} f_{V_q}(v)$
  end for
end for

Algorithm 2 has the following desirable convergence property:

Theorem 2.3.1. *The limit points of the iterates* $\{(\boldsymbol{U}_i, \boldsymbol{V}_i)\}_{i=1}^{\infty}$ *produced by Algorithm 2 are all stationary points of the optimization problem* (2.4).

Proof. This is a consequence of Proposition A.3.1 in Appendix A.3. □

Each of the subproblems corresponds to the computation of $\mathbf{U}^i, \mathbf{V}^i$. Assume that we want to compute $\mathbf{U}_q^i$, the q^{th} column of $\mathbf{U}^i$. This is the problem

$$\mathbf{U}_q^i = \arg\min_{u \in \mathbb{R}^m} \|\tilde{\mathbf{X}}_q - u\mathbf{V}_q^{i-1}\|_1 + ||u - \mathbf{U}_q^{i-1}||_F^2 \tag{2.6}$$

where $\mathbf{V}_q^{i-1}$ is the q^{th} row of the matrix $\mathbf{V}^{i-1}$ and $\tilde{\mathbf{X}}_q := \mathbf{X} - \mathbf{U}^{i-1}\mathbf{V}^{i-1} + \mathbf{U}_q^{i-1}\mathbf{V}_q^{i-1}$.
Problem (2.6) is equivalent to

$$\sum_{j=1}^{m} \min_{t \in \mathbb{R}} \left[\|r_j - t\mathbf{V}_q^{i-1}\|_1 + (t - \mathbf{U}_{j,q}^{i-1})^2\right],$$

with r_j the j^{th} row of $\tilde{M}_q$, and $\mathbf{U}_{j,q}^{i-1}$ denoting the $(j,q)^{th}$ entry of the matrix $\mathbf{U}^{i-1}$.
Thus, we have reduced the computation of $\mathbf{U}_q^i$ to m single-variable convex optimization problems of the form

$$\min_{t \in \mathbb{R}} \|a - tb\|_1 + (t - t_0)^2, t_0 \in \mathbb{R}, a, b \in \mathbb{R}^n. \tag{2.7}$$

There are a variety of techniques for solving (2.7). Since the objective function is piecewise differentiable, it can be solved to within ϵ accuracy in $O(n)$ time by a modified version of the bisection algorithm, or by using the subgradient method [Bertsekas, 1999]. See Appendix A.2 for more details.
Thus, $\mathbf{U}_q^i$ can be computed in $O(mn)$ time. Since $\mathbf{U}^i$ has r columns, then it can be computed in $O(rmn)$ time. Similarly $\mathbf{V}^i$ can be computed in $O(rmn)$ time. Thus, Algorithm 2 takes $O(rmn)$ time per iteration.

2.4 Experimental Results

In this section, we compare the performance of an Alternating Directions Method of Multipliers (ADMM) solver for the PCP convex program (2.2) (Boyd et al. [2011]) and Algorithm 2 on random experiments. In these problems, the low rank matrix $\mathbf{L}^* = \mathbf{U}^*\mathbf{V}^* \in \mathbb{R}^{m \times m}$ with $\mathbf{U}^*, \mathbf{V}^{*T} \in \mathbb{R}^{m \times r}$ and the entries of $\mathbf{U}^*, \mathbf{V}^*$ i.i.d $N(0, \frac{1}{m})$. The sparse matrix $\mathbf{S}^*$ has k non-zeros chosen uniformly at random, each with random signs.

Each algorithm produces estimates $\hat{\mathbf{L}}, \hat{\mathbf{S}}$. We consider the metrics *a*) $\frac{||\hat{\mathbf{L}}-\mathbf{L}^*||_F}{||\mathbf{L}^*||_F}$, the relative error of the low rank matrix $\hat{\mathbf{L}}$ produced by an algorithm, *b*) $\frac{||\hat{\mathbf{S}}-\mathbf{S}^*||_F}{||\mathbf{S}^*||_F}$, the relative error of the sparse matrix $\hat{\mathbf{S}}$ produced by an algorithm, *c*) $Time(s)$, the algorithm's running time, *d*) and k, the number of iterations the algorithm runs before terminating.

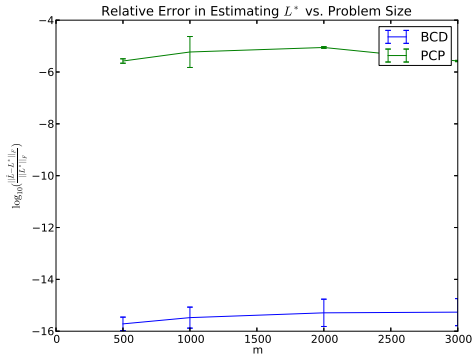
We are interested in comparing the performance of the two algorithms when the rank r of the low-rank matrix $\mathbf{L}^*$ is *a*) $\Theta(1)$ and *b*) $\Theta(m)$.

Figure (2.1) summarizes the performance of the algorithms when the underlying low-rank matrix has rank $r = 1$, representing the case where $r = \Theta(1)$. As Figure (2.1a) indicates, Algorithm 2 is able to recover the low rank component $\mathbf{L}^*$ with accuracy superior to that of the ADMM solver. Similarly, we see from Figure (2.1b) that the sparse component $\mathbf{S}^*$ is recovered accurately by our technique. Finally, for this constant rank experiment we observe that Algorithm 2 runs far more quickly than the ADMM solver, and in roughly 3 iterations across a wide range of problem sizes (see Figure (2.1c) and (2.1d), respectively). In summary, Algorithm 2 appears to be comparable to or outperform the ADMM solver for PCP by all of these four metrics considered.

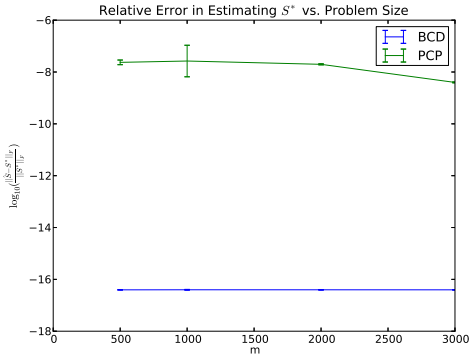
However, as Figure (2.2) indicates, when $r = \Theta(m)$, the relative performance of the two algorithms changes. From Figure (2.2a) and Figure (2.2b), we see that in this new regime Algorithm 2 is not able to recover $\mathbf{L}^*$ or $\mathbf{S}^*$ with accuracy comparable to the ADMM solver. Furthermore, the running time of Algorithm 2 is substantially worse than that of the ADMM solver. Finally, the number of iterations used by Algorithm 2 appears to be slightly increasing, while that of the ADMM solver seems to be constant despite m increasing (see Figure (2.2c) and (2.2d), respectively).

2.5 Conclusion

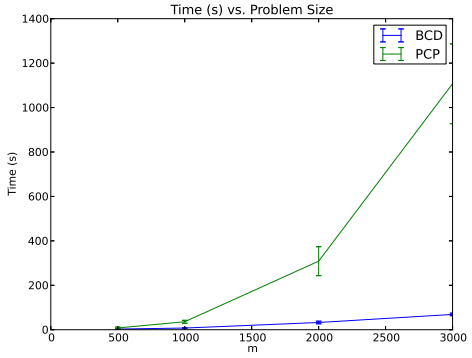
We have introduced a new algorithm for Robust PCA that satisfies a certain convergence condition and has low computational complexity for problems for which the rank r of the low-rank matrix is small. We have also demonstrated that the algorithm empirically performs as well as more computationally expensive approaches on randomly generated problems for small r.



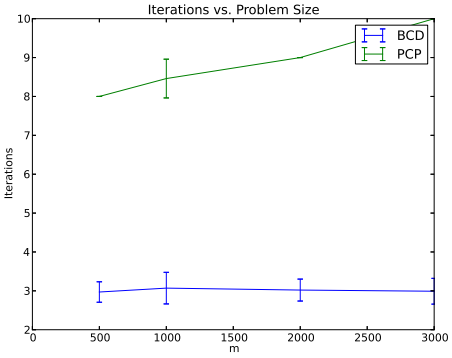
(a) Relative Error in estimating $\mathbf{L}^*$



(b) Relative Error in estimating $\mathbf{S}^*$

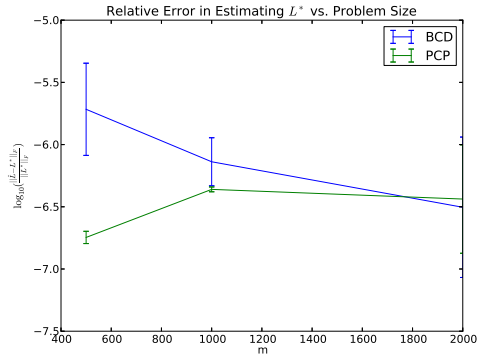


(c) Running Time

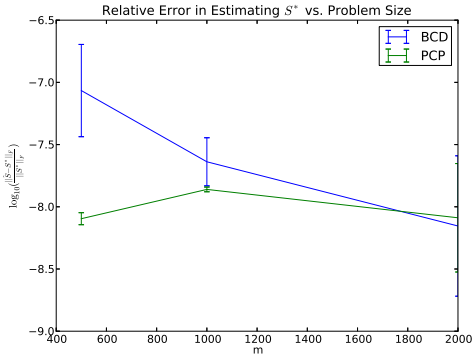


(d) Number of Iterations

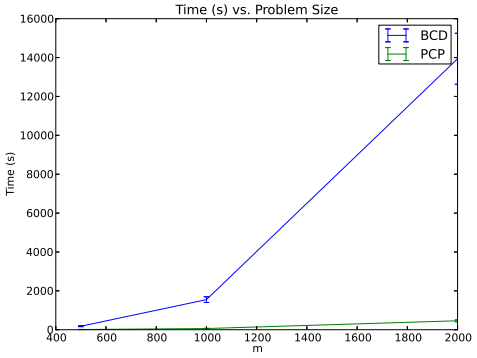
Figure 2.1: Experimental Results for constant rank, $r = \Theta(1)$



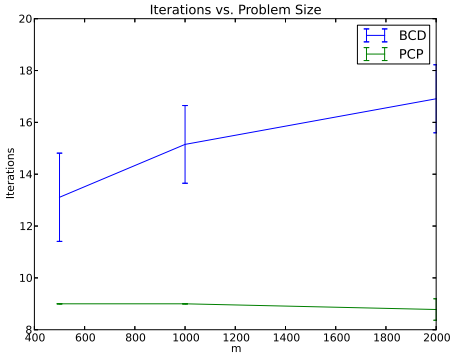
(a) Relative Error in estimating $\mathbf{L}^*$



(b) Relative Error in estimating $\mathbf{S}^*$



(c) Running Time



(d) Number of Iterations

Figure 2.2: Experimental Results for linear rank, $r = \Theta(m)$

In future work, we would like to provide bounds on how many iterates are required for convergence to a stationary point and also find conditions under which (a) there is only one limit point of the iterates and (b) the stationary points correspond to local minima or a global minimum. We would also like to test the empirical behavior of this approach on large, real-world datasets for which the sparse plus low-rank model is appropriate.

Chapter 3

Penalized Regression Models for the NBA

3.1 Introduction

The National Basketball Association (NBA) is a multi-billion dollar business. Each of the thirty franchises in the NBA try their best to put forward the most competitive team possible within their budget. To accomplish this goal, a key task is to understand how good players are.

A large fraction of the thirty NBA teams have quantitative groups analyzing data to evaluate and rate players. The website ESPN.com has many analysts providing statistical analysis for casual fans. Gambling houses use quantitative analysis to price bets on games, while gamblers try to use quantitative analysis to find attractive wagers.

A popular technique for producing player ratings is weighted least-squares (LS) regression[1]. However, as we show later show, least squares is an approach with many flaws.

In this paper, we introduce a new penalized regression technique for estimating player ratings which we call Subspace Prior Regression (henceforth, SPR). SPR corrects some of the flaws of least squares for this problem setting, and has substantially better out-of-sample predictive performance. Furthermore, given sufficient training data SPR outperforms the Las Vegas wagering lines.

We interpret the ratings produced by SPR, discussing it identifies as the best players in the NBA (Section 3.6.1), who are the most overrated and underrated players (Section 3.6.2), and what SPR suggests is the relative importance of different basic actions within the game like three point shooting and turnovers (Section 3.6.3). Finally, we discuss some possible improvements to this model (Section 3.7).

[1]this technique is also known as Adjusted Plus/Minus (APM) in the quantitative basketball community.

3.2 Notation

We use the notation $\mathbb{R}^d_+$ to indicate the set $\{x \in \mathbb{R}^d \mid x_i > 0 \; \forall i\}$. Let $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ denote the column vector of ones and $\mathbf{e}_i \in \mathbb{R}^{n\times 1}$ signify the i^{th} standard basis vector. We use $\mathbf{I}_p \in \mathbb{R}^{p\times p}$ to denote the identity matrix of size p, and $\mathbf{Diag}(\mathbf{w})$ to stand for a diagonal matrix with entries given by the vector $\mathbf{w}$. Given $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ and $\mathbf{c} \in \mathbb{R}^n_+$ we define the inner product as $\mathbf{a}^T\mathbf{b} := \sum_{i=1}^n a_i b_i$, the ℓ_p norm $||\mathbf{a}||_p := [\sum_{i=1}^n a_i^p]^{\frac{1}{p}}$ and finally the $\mathbf{c}$-weighted ℓ_p norm as $||\mathbf{a}||_{p,\mathbf{c}} := [\sum_{i=1}^n c_i a_i^p]^{\frac{1}{p}}$.

3.3 A Brief Introduction to the Game of Basketball

Each of the thirty teams in the NBA plays 82 games in a season, where 41 of these games are at their home arena and 41 are played away. Thus, there are 1,230 total games in an NBA regular season. Each team has a roster of roughly twelve to fifteen players. Games are usually 48 minutes long, and each of the two competing teams has exactly five players on the floor at a time. Thus, there are ten players on the floor for the duration of the game. Associated with each game is a box score, which records the statistics of the players who played in that game. Figure 3.1 contains a sample box score from an NBA game played on February 2nd, 2011 by the Dallas Mavericks (the home team) against the New York Knicks (the away team). Note that we only display the box score for the Mavericks players. Observe that there are 12 players listed in the box score, but only 11 who actually played for the Mavericks in this game. Each of the columns of this box score corresponds to a basic statistic of interest (the column REB in the box score denotes rebounds, AST denotes steals, etc.)

3.3.1 Statistical Modeling of Basketball

To statistically model the NBA, we must first extract from each game a dataset suitable for quantitative analysis. There is a standard procedure for this currently used by many basketball analysts [Kubatko et al., 2007, Oliver, 2004], which we describe as follows.

We model each basketball game as a sequence of n distinct events between two teams. During event i the home team scores Y_i more points than the away team. We use the variable p to denote the total number of players in the league (in a typical NBA season, $p \approx 450$.) We can then represent the current players on the floor for event i with a vector $\mathbf{X}_i \in \mathbb{R}^p$ defined as

Figure 3.1: Sample single-game boxscore for the Dallas Mavericks

Dallas Mavericks

STARTERS	MIN	FGM-A	3PM-A	FTM-A	OREB	DREB	REB	AST	STL	BLK	TO	PF	+/-	PTS
Brian Cardinal, PF	10	1-1	1-1	0-0	0	0	0	3	0	0	0	1	-3	3
Dirk Nowitzki, PF	33	10-16	2-4	7-7	1	10	11	3	0	1	1	2	+27	29
Tyson Chandler, C	29	6-9	0-0	3-4	3	8	11	0	0	1	1	3	+16	15
Jason Kidd, PG	32	2-10	2-7	0-0	1	5	6	10	1	0	1	1	+16	6
DeShawn Stevenson, SG	31	5-9	3-6	0-0	2	3	5	1	0	0	1	1	+9	13
BENCH	**MIN**	**FGM-A**	**3PM-A**	**FTM-A**	**OREB**	**DREB**	**REB**	**AST**	**STL**	**BLK**	**TO**	**PF**	**+/-**	**PTS**
Jason Terry, SG	30	6-12	0-3	0-1	1	3	4	2	1	0	2	2	+7	12
Shawn Marion, SF	23	1-8	0-0	3-4	1	9	10	2	0	0	4	3	-9	5
Brendan Haywood, C	16	2-4	0-0	0-0	0	2	2	0	0	1	0	4	+3	4
Jose Juan Barea, PG	30	7-12	3-4	5-5	1	2	3	3	1	0	3	0	+21	22
Ian Mahinmi, C	4	1-1	0-0	0-0	0	1	1	0	0	1	1	1	-3	2
Dominique Jones, SG	3	1-2	0-0	0-0	0	1	1	0	0	0	1	0	-4	2
Peja Stojakovic, SF	DNP NOT WITH TEAM													
TOTALS		**FGM-A**	**3PM-A**	**FTM-A**	**OREB**	**DREB**	**REB**	**AST**	**STL**	**BLK**	**TO**	**PF**		**PTS**
		42-84	**11-25**	**18-21**	**10**	**44**	**54**	**24**	**3**	**4**	**15**	**18**		**113**
		50.0%	**44.0%**	**85.7%**										

+/- denotes team's net points while the player is on the court.

Fast break points: 15
Points in the paint: 38
Team TO (points off): 16 (16)

$$\mathbf{X}_{ij} = \begin{cases} 1 & \text{Player } j \text{ is on the floor for the home team} \\ \text{-1} & \text{Player } j \text{ is on the floor for the away team} \\ 0 & \text{otherwise.} \end{cases}$$

Associated with event i is a weighting factor w_i. Roughly speaking, the i^{th} event happens for w_i minutes.

Figure 3.1 contains a sample box score. We summarize box score data like that of Figure 3.1 with the matrix $\mathbf{R}_{\text{Mavericks,Game 1}}$ which looks like

$$\mathbf{R}^{\text{Mavericks}}_{\text{Game \#1}} = \begin{array}{c} \\ \text{Brian Cardinal} \\ \text{Dirk Nowitzki} \\ \vdots \\ \text{Peja Stojakovic} \end{array} \begin{array}{c} \begin{array}{ccccc} \text{MIN} & \text{FGM} & \text{FGA} & \dots & \text{PTS} \end{array} \\ \begin{pmatrix} 10 & 1 & 1 & \dots & 3 \\ 33 & 10 & 16 & \dots & 29 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} \end{array}$$

This matrix records the statistics of the 12 players on the Dallas Mavericks roster for that particular game. If there are d basic statistics of interest in this box score, then $\mathbf{R}^{\text{Mavericks}}_{\text{Game \#1}}$ is a matrix of size 12 by d.

One can imagine computing the aggregate box score matrix

$$\mathbf{R}^{\text{Mavericks}} = \sum_{t=1}^{82} \mathbf{R}^{\text{Mavericks}}_{\text{Game \#t}}$$

that summarizes the total statistics of these 12 players for an entire season. Finally, define the $p \times d$ matrix $\mathbf{R}$ that vertically concatenates $\mathbf{R}_j$ across the 30 teams in the NBA:

$$\mathbf{R} := \begin{matrix} \text{Team 1} \\ \text{Team 2} \\ \vdots \\ \text{Team 30} \end{matrix} \begin{pmatrix} \mathbf{R}^{\text{Mavericks}} \\ \mathbf{R}^{\text{Bulls}} \\ \vdots \\ \mathbf{R}^{\text{Celtics}} \end{pmatrix}.$$

$\mathbf{R}$ summarizes the season box score statistics for all p players who played in the NBA for that year.

3.3.2 Least Squares Estimation

We want to determine the relationship between $\mathbf{X}_i$ and $\mathbf{Y}_i$, i.e., find a function f such that $\mathbf{Y}_i \approx f(\mathbf{X}_i)$. One natural way to do this is through a linear regression model, which assumes that

$$\mathbf{Y}_i = \alpha^*_{\text{hca}} + \mathbf{X}_i^T \boldsymbol{\beta}^* + e_i, i = 1, 2, \ldots n.$$

Recall that the event i has a weighting factor w_i associated with it. Roughly speaking, event i happens for w_i minutes.

The scalar variable α^*_{hca} represents a home court advantage term, while the variable $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is interpreted as the number of points each of the p players in the league "produces" per minute. This model recognizes players for whom their team is more effective because of their presence on the floor.

For notational convenience, we stack the variables $\mathbf{Y}_i$, w_i, and e_i into the n vectors $\mathbf{Y}$, $\mathbf{W}$, and $\mathbf{E}$ and the variables $\mathbf{X}_i$ into the $n \times p$ matrix $\mathbf{X}$. This yields the matrix expression

$$\mathbf{Y} = \mathbf{1}_n \alpha^*_{\text{hca}} + \mathbf{X}\boldsymbol{\beta}^* + \mathbf{E}. \tag{3.1}$$

Given observations $(\mathbf{Y}, \mathbf{X})$ and weights $\mathbf{W}$, we define the $\mathbf{W}$-weighted quadratic loss function as

$$L_{\text{quadratic}}(\alpha_{\text{hca}}, \boldsymbol{\beta}) := \frac{1}{\sum_{i=1}^n w_i} ||\mathbf{Y} - \mathbf{1}_n \alpha_{\text{hca}} - \mathbf{X}\boldsymbol{\beta}||^2_{\mathbf{W}}. \tag{3.2}$$

A natural technique for estimating the variables α^*_{hca} and $\boldsymbol{\beta}^*$ is to minimize (3.2), i.e.,

$$(\hat{\alpha}^{\text{LS}}_{\text{hca}}, \hat{\boldsymbol{\beta}}^{\text{LS}}) = \arg\min_{\alpha_{\text{hca}}, \boldsymbol{\beta}} L_{\text{quadratic}}(\alpha_{\text{hca}}, \boldsymbol{\beta}), \tag{3.3}$$

resulting in a weighted least squares (LS) problem. The values $\hat{\boldsymbol{\beta}}^{\text{LS}}$ are known in the quantitative basketball community as the *adjusted plus/minus ratings*[2]. The website Basketballvalue[3] has computed $\hat{\boldsymbol{\beta}}^{\text{LS}}$ for several recent seasons.

3.3.3 Is least squares regression a good estimator of player value?

Table 3.1. LS Player Ratings

Rank	Player	$\hat{\boldsymbol{\beta}}_i^{\text{LS}}$
1	James, LeBron	12.62
2	Durant, Kevin	11.5
3	Nash, Steve	11.39
4	Paul, Chris	10.42
5	Nowitzki, Dirk	10.33
6	Collison, Nick	9.87
7	Wade, Dwyane	9.59
8	Hilario, Nene	8.56
9	Deng, Luol	8.46
10	Howard, Dwight	8.13

Table 3.1 lists the top ten players in the NBA for the combined 2009-2010 and 2010-2011 NBA regular seasons by their ratings produced from least squares[4]. By this ranking, LeBron James was the best player in the league over this two year period. Since $\hat{\boldsymbol{\beta}}^{\text{LS}}_{\text{LeBron James}} = 12.62$, this procedure suggests that he is worth an additional 12.62 net points to his team for every 100 possessions the team plays.

How believable are the player ratings of Table 3.1? The list has many of the widely-considered best players in the NBA. However, there are also some names on this list that are questionable. If we believe these ratings, then Nick Collison, a player considered by most fans and analysts to be at best a merely average player at his position, is better than Dywane Wade and Dwight Howard, two of the premiere superstars in the league. Similarly, while Nene Hilario and Luol Deng are good players, they are not considered by most fans and analysts to be amongst the top ten players in the NBA.

[2] `http://www.82games.com/ilardi1.htm`

[3] `http://www.basketballvalue.com`

[4] These numbers were obtained from `http://basketballvalue.com/topplayers.php?&year=2010-2011`

This contradiction between common wisdom and least squares is useful, since it can either reveal to us that the common wisdom is wrong or that the least squares approach is incorrect. We need some basis of comparison to evaluate how well least squares is performing.

In classical linear regression, assuming that the generative model satisfies certain conditions, the least squares estimate has several desirable properties (maximum likelihood estimate, best linear unbiased estimate, consistency, asymptotic normality, etc). However, these properties typically assume that the underlying model satisfies certain technical conditions like normality, linearity, and statistical independence. It is unreasonable to expect that these technical conditions hold for the game of basketball. Thus, we must find other ways to evaluate how trustworthy the $\hat{\boldsymbol{\beta}}^{\text{LS}}$ values are, and whether they should be believed over common wisdom about players. One simple approach for evaluating the the least squares model is to test its predictive power versus a simple dummy estimator.

To do this, we

1. define a dummy estimator that sets $\hat{\beta}_i^{\text{Dummy}} = 0$ for each player, and the home court advantage term $\hat{\beta}_{\text{hca}}^{\text{Dummy}} = 3.5$. In other words, each player is rated a zero, and the home team is predicted to win every 100 possessions by 3.5 points.

2. We then can compute both the least squares estimate and dummy estimate for the first 820 games of an NBA season, and measure how well each technique does in estimating the margin of victory of the home team for the remaining 410 games of that season.

If least squares accurately models the NBA, then at a minimum it must substantially outperform the dummy estimator. Let us use the variable A_k to denote the *actual* number of points by which the home team wins game k, $\hat{A}_k$ to denote the *predicted* number of points by the statistical estimator of interest, and $\hat{E}_k := \hat{A}_k - A_k$ to denote the error this statistical estimator makes in predicting the outcome of game k.

Figure 3.2 is a histogram of the error variable $\hat{E}_k$ over the course of the 410 games under consideration from the 2010-2011 NBA season for each technique. A perfect estimator would have a spike of height 410 centered around zero. Thus, the "spikier" the histogram looks, the better a method performs. It is hard to immediately say from Figure 3.2 that the least squares estimate yields better predictions than the simple dummy estimate. We can also study some of the empirical properties of $\hat{E}_k$ for each approach. Table 3.2 summarizes the results.

When comparing least squares to the dummy estimate, we notice that

1. least squares reduces the percentage of games in which the wrong winner is identified from 39.27% to 33.66% over the block of 410 games of interest.

Table 3.2: Performance of Statistical Estimators over the last 410 games

Metric	Dummy	LS	RR	SPR	SPR2
RMSE (millions)	547.0447	558.0350	551.0912	539.7495	541.9725
Fraction of games guessed wrong	0.3927	0.3366	0.3293	0.2854	0.2951
Mean of $\lvert\hat{E}_i\rvert$	10.5394	18.0507	16.0016	10.5540	11.8109
Variance of $\lvert\hat{E}_i\rvert$	68.5292	187.1446	147.8993	60.2527	70.7671
Median of $\lvert\hat{E}_i\rvert$	9.4885	15.2577	13.6127	8.9687	10.2200
Min of $\lvert\hat{E}_i\rvert$	0.0279	0.0346	0.0109	0.0260	0.1445
Max of $\lvert\hat{E}_i\rvert$	38.8338	79.1965	72.0122	40.9012	45.9275
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 1)$	0.9171	0.9707	0.9683	0.9463	0.9512
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 3)$	0.7683	0.9000	0.8878	0.8171	0.8732
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 5)$	0.6707	0.8366	0.8122	0.7244	0.7780
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 10)$	0.4854	0.6585	0.6049	0.4415	0.5098

2. Unfortunately, the empirical behavior of $\hat{E}_k$ seems to be substantially worse for least squares. For example, the empirical mean of $\lvert\hat{E}_i\rvert$ is 18.05 for least squares, while only 10.54 for the dummy estimator. Thus, least squares makes larger average errors when predicting the final margin of victory of games.

As a result, it is hard to convincingly argue that least squares approach is a better model for the NBA than the dummy estimate.

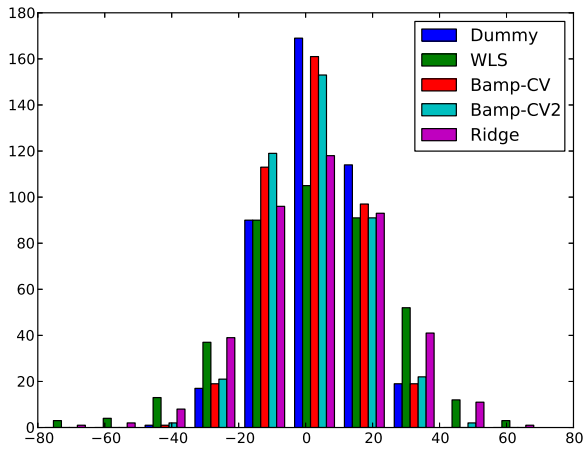


Figure 3.2. Comparison of Dummy, Least Squares, Ridge Regression, SPR and SPR2 trained on 820 games.

3.4 SPR: Improving Least Squares

Although Figure 3.2 and Table 3.2 suggest that the LS estimate performs poorly, this doesn't necessarily mean that the linear model (3.1) is without promise. The least squares estimate simply doesn't take into account the following two key pieces of information we have about the problem domain:

1. Model sparsity: The NBA is a game dominated by star players. Lesser players have far less impact on wins and losses. This folk wisdom informs player acquisitions and salaries. For example, with a \$60 million budget, one would much rather acquire three elite \$15 million stars and fill out the rest of the roster with cheap role-players, than spend tons of money on role-players and skimp on stars.

   This "elites first" strategy was used by the Boston Celtics in the summer of 2007 when they traded their role-players and other assets to build a team around Kevin Garnett, Paul Pierce and Ray Allen[5], and more recently by the Miami Heat in the summer of 2010 who built a team around LeBron James, Dywane Wade and Chris Bosh[6]. We shall incorporate this prior information through ℓ_1 regularization. This penalizes non-sparse models, and should cause only the very best players to stand out in the regression. This suggests a penalty term of the form $\lambda_1||\boldsymbol{\beta}||_1$.

2. Box score information: Another valuable piece of information useful in inferring player worth is the box score statistics matrix $\mathbf{R}$. One expects good players to not only have high APM ratings, but to also produce rebounds, assists, blocks, steals, etc. Thus, we prefer ratings $\hat{\boldsymbol{\beta}}$ which are consistent with box score statistics. In other words, we expect a ratings vector to be "close" to the column space of $\mathbf{R}$. We therefore should penalize ratings for which the distance from $\hat{\boldsymbol{\beta}}$ to $\mathbf{Rz}$ is large. Although there are many different possible penalties one can choose, in this work we choose a quadratic penalty term of the form $\lambda_2||\boldsymbol{\beta} - z_0\mathbf{1}_p - \mathbf{Rz}||_2^2$.

We can encode the above prior information through the function $g(\alpha_{\text{hca}}, \boldsymbol{\beta}, z_0, \mathbf{z}; \vec{\lambda})$ defined as

$$g(\alpha_{\text{hca}}, \boldsymbol{\beta}, z_0, \mathbf{z}; \vec{\lambda}) := \underbrace{L_{\text{quadratic}}(\alpha_{\text{hca}}, \boldsymbol{\beta})}_{\text{Weighted least squares}} + \underbrace{\lambda_1||\boldsymbol{\beta}||_1}_{\text{Sparse player ratings}} + \underbrace{\lambda_2||\boldsymbol{\beta} - z_0\mathbf{1}_p - \mathbf{Rz}||_2^2}_{\text{Box score prior}}, \tag{3.4}$$

1. $\mathbf{R}$ is a $p \times d$ matrix containing the box-score statistics of the p different players,

[5] `http://www.nba.com/celtics/news/press073107-garnett.html`
[6] `http://sports.espn.go.com/nba/news/story?id=5365165`

2. The variable $\mathbf{z}$ gives us weights for each of the box score statistics,

3. and the vector $(\lambda_1, \lambda_2) \in \mathbb{R}_+^2$ are the regularization parameters.

We shall use the shorthand $\vec{\lambda}$ to denote the pair (λ_1, λ_2). We can find a model consistent with both the data and the prior information by solving the convex optimization problem

$$\hat{\beta}_{\text{hca}}^{\vec{\lambda}}, \hat{\boldsymbol{\beta}}^{\vec{\lambda}}, \hat{\beta}_0^{\vec{\lambda}}, \hat{\mathbf{z}}^{\vec{\lambda}} = \arg\min g(\alpha_{\text{hca}}, \boldsymbol{\beta}, z_0, \mathbf{z}; \vec{\lambda}). \tag{3.5}$$

We call the procedure described by Equation (3.5) the SPR algorithm, and the vector $\hat{\boldsymbol{\beta}}^{\vec{\lambda}}$ are the player ratings produced by it. One very important difference between SPR and the least squares approach is that it yields both a player rating vector $\hat{\boldsymbol{\beta}}^{\vec{\lambda}}$ and a box score weights vector $\hat{\mathbf{z}}^{\vec{\lambda}}$. The weights vector $\hat{\mathbf{z}}^{\vec{\lambda}}$ is a valuable tool in its own right. It provides numerical values for different basic box score statistics like scoring, rebounding, and steals. We further interpret $\hat{\mathbf{z}}^{\vec{\lambda}}$ in Section 3.6.3.

Furthermore, it yields a linear formula for transforming player box score effectiveness into a player productivity rating through the equation

$$\hat{\boldsymbol{\theta}}^{\vec{\boldsymbol{\lambda}}} := \mathbf{R}\hat{\boldsymbol{z}}^{\vec{\lambda}} + \hat{\beta}_0^{\vec{\lambda}}\mathbf{1}_p. \tag{3.6}$$

$\hat{\boldsymbol{\theta}}^{\vec{\boldsymbol{\lambda}}}$ can be viewed as an additional player rating vector produced by SPR, one that linearly transforms each player's box score production into a "points per 100 posession" rating similar to least squares or $\hat{\boldsymbol{\beta}}^{\vec{\lambda}}$. Thus, $\hat{\boldsymbol{\theta}}^{\vec{\boldsymbol{\lambda}}}$ succintly converts the box score production of each player into a single number.

Thus for player i, we can compare the variable $\hat{\boldsymbol{\beta}}_i^{\vec{\lambda}}$ to the variable $\hat{\boldsymbol{\theta}}^{\vec{\boldsymbol{\lambda}}}{}_i$ to understand how "overrated" or underrated he is relative to his box score production. This is useful, since many players produce great box score statistics but don't necessarily impact team competitiveness to the level the box score might suggest. We explore this aspect of SPR in further detail in Section 3.6.2.

3.4.1 Bayesian Interpretation of SPR

SPR can be interpreted as the posterior mode for a Bayesian statistics model. Suppose that $\mathbf{Y}, \alpha_{\text{hca}}, \boldsymbol{\beta}, z_0, \mathbf{z}$ are all random variables.

Let

- $\mathbf{Y}_i | \alpha_{\text{hca}}, \boldsymbol{\beta} \sim \mathcal{N}(\alpha_{\text{hca}} + \mathbf{X}_i^T \boldsymbol{\beta}, \frac{1}{2})$,
- α_{hca} have the improper prior $\mathbb{P}(\alpha_{\text{hca}} = \alpha) \propto 1$

Table 3.3: Regularization parameters obtained from 10-fold cross-validation

Setting	λ_1	λ_2
$\vec{\lambda}^{820}_{CV}$	2^{-10}	2^{-3}
$\vec{\lambda}^{820}_{CV,2}$	2^{-10}	2^{-1}
$\vec{\lambda}^{410}_{CV}$	2^{-10}	2^{-2}

- $\mathbb{P}(\boldsymbol{\beta}|z_0, \mathbf{z}) \propto e^{-\lambda_1||\boldsymbol{\beta}||_1 - \lambda_2||\boldsymbol{\beta} - z_0\mathbf{1}_p - \mathbf{R}\mathbf{z}||_2^2}$,
- z_0 have the improper prior $\mathbb{P}(z_0 = \gamma) \propto 1$,
- and $\mathbf{z}$ has the improper prior $\mathbb{P}(\mathbf{z} = \kappa) \propto 1$.

Then the solution to SPR with $\mathbf{w} = \mathbf{1}_n$ is exactly the mode of the posterior distribution $\mathbb{P}(\alpha_{\text{hca}}, \boldsymbol{\beta}, z_0, \mathbf{z}|\mathbf{Y})$.

3.4.2 Selecting the regularization parameter $\vec{\lambda}$

For SPR to be useful, we need to be able to select a good choice of $\vec{\lambda}$ quickly. Cross-validation [Stone, 1974] is one standard technique in statistics for doing this. To select regularization parameters, we use 10-fold cross-validation. We cross-validate over regularization parameters from the set

$$\Lambda := \{(2^a, 2^b)|\ a, b \in F\}$$

where

$$F := \{-10, -9, \ldots, 9\}.$$

K-fold cross-validation on T different values of $\vec{\lambda}$ means solving TK different SPR problems, each of which are convex programs of moderate size ($n \approx 20000$, $p \approx 450$, $d \approx 20$).

Thus, it is necessary that

1. for each fixed value of $\vec{\lambda}$, SPR can be solved quickly
2. and that many values of $\vec{\lambda}$ can be evaluated at once.

To address the first issue, we implemented a fast numerical algorithm for solving SPR for a fixed valued of $\vec{\lambda}$. See Appendix A.5 for a derivation.

To address the second issue, our cross-validation code takes advantage of the cloud computing service PiCloud[7] to perform the computations in parallel.

The resulting regularization parameters learned by cross-validation are summarized in Table 3.3.

[7] `http://www.picloud.com`

3.5 The Performance of SPR

Our ultimate goal is to produce substantially better estimates of player value than least squares. If it turns out that despite all the additional computational work that SPR requires that there is little or no statistical improvement, then SPR is not of much practical value. In this section, we discuss the performance of SPR on the 2010-2011 NBA dataset. We demonstrate that SPR substantially outperforms both the dummy estimate and least squares estmate, and outperforms even Las Vegas given a sufficient amount of training data.

3.5.1 SPR outperforms least squares

From Table 3.3, we see that the cross-validation methodology described in Section 3.4.2 on the first 820 games of the 2010-2011 season yields the regularization parameter

$$\vec{\lambda}_{CV}^{820} = (2^{-10}, 2^{-3}).$$

Armed with this choice, we can now compare least squares to SPR on the final 410 games of the 2010-2011 NBA regular season. Each procedure produces a player rating vector $\hat{\boldsymbol{\beta}}$, and we can use these ratings to predict the final margin of victory over this collection of games.

Recall that we use the variable $\hat{A}_i$ to denote the number of points that a statistical estimator predicts that the home team will win game i, A_i to denote the *actual* number of points by which the home team wins game i, and $\hat{E}_k = \hat{A}_i - A_i$ to denote the difference between these quantities.

Figure 3.2 is a histogram of the variable $\hat{E}_k$ for each technique. It is clear from Figure 3.2 that SPR produces better estimates than APM. The histogram of the SPR errors are "spikier" around the origin than the APM errors. We can also study some of the empirical properties of the variable $\hat{E}_k$ for each approach. Table 3.2 summarizes the results. As Table 3.2 indicates, SPR represents a substantial improvement on APM in nearly all of these statistical measures. In particular,

- the fraction of games in which the wrong winner is guessed decreases from 33.66% with LS to 28.54% with SPR; and
- the average absolute error in predicting the margin of victory decreases from 18.05 to 10.554.

Comparing SPR to the dummy estimator, we

- see an enormous improvement in ability to predict the winning team. The percentage of games in which the wrong winner is predicted falls from 39.27% to 28.54%.

- Both techniques obtain a similar average absolute error in predicting the margin of victory, with 10.54 for the dummy estimator and 10.55 with SPR.

Overall, this suggests that SPR more accurately models the NBA than the least squares estimator.

3.5.2 SPR outperforms Las Vegas

To convincingly evaluate the performance of SPR, we examine whether it actually results in a profitable gambling strategy against the Vegas lines. In fact, we will compare the dummy, least squares and SPR estimators. Given predictions by each of the above estimates, we have the following natural gambling strategy:

1. If the deviation Δ between the estimate's prediction of the outcome of a game and the Vegas lines is greater than 3, place a bet on the team the estimator favors.

2. Otherwise, don't bet.

Due to transaction costs that the sportbooking companies charge[8] a gambling strategy must win more than roughly 52.5% of the time to at least break even. Table 3.4 summarizes the result of this gambling rule for each of the three techniques of interest over the last 410 games of the 2010-2011 NBA season. The dummy-based gambling strategy places 263 bets on the 410 games and loses 3 more bets than it wins, for a winning percentage below 50%, which is performance comparable to random guessing, and not enough to break even. The least squares-based strategy has a winning percentage of 51.97% on 356 bets made. In comparison, SPR places wagers on 290 games and wins 57.24% of these bets. This represents a very profitable betting strategy, and thus suggests that SPR more accurately models the NBA than major alternatives, including the estimators used by Las Vegas. Finally, SPR obtains this improved performance while only having access to the first 820 games of the regular season.

Table 3.4: Betting Strategy over the last 410 games, $\Delta = 3$

Statistic	Dummy	LS	RR	SPR	SPR2
# of bets possible	410.0000	410.0000	410.0000	410.0000	410.0000
# of bets made	263.0000	356.0000	346.0000	290.0000	321.0000
Net # of bets won	-3.0000	14.0000	26.0000	42.0000	21.0000
Winning percentage	0.4943	0.5197	0.5376	0.5724	0.5327

[8]The fee is called the "vigorish" in the gambling community.

3.5.3 Robustness of Results

How sensitive is the SPR algorithm to our choice of training on the first 410 games? Does the performance relative to the least squares estimate degrade if the estimators are trained on much fewer games? To evaluate this, we train estimators on the first 410 games and then evaluate predictive power on the remaining 820 games. From Table 3.3, we obtain the cross-validation selected regularization parameter

$$\vec{\lambda}_{CV}^{410} = (2^{-7}, 2^{-2}).$$

We also compare against the Las Vegas predictions for that block of 820 games. Table 3.5 summarizes the results of this experiment. As before, SPR outperforms both the Dummy estimator and LS. Furthermore, by increasing Δ to 5 (from the value 3 used when training on 820 games), SPR still leads to a successful betting strategy, as Table 3.6 shows.

Table 3.5: Robustness Experiment, First 410 Games

Metric	Dummy	LS	RR	SPR
RMSE (millions)	1087.4967	1209.3042	1130.7781	1078.7885
Fraction of games guessed wrong	0.4024	0.4073	0.3732	0.3049
Mean of $\lvert\hat{E}_i\rvert$	10.3326	28.9714	19.9875	11.4719
Variance of $\lvert\hat{E}_i\rvert$	64.9664	524.1851	238.1755	74.1957
Median of $\lvert\hat{E}_i\rvert$	9.2783	23.3688	17.3077	9.5853
Min of $\lvert\hat{E}_i\rvert$	0.0279	0.0491	0.0158	0.0208
Max of $\lvert\hat{E}_i\rvert$	49.2299	150.4097	98.2374	43.1902
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 1)$	0.9207	0.9756	0.9659	0.9573
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 3)$	0.7768	0.9329	0.9000	0.8378
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 5)$	0.6744	0.8817	0.8195	0.7378
Empirical $\mathbb{P}(\lvert\hat{E}_i\rvert > 10)$	0.4720	0.7805	0.6890	0.4780

Table 3.6: Betting Strategy over the last 820 games, $\Delta = 5$

Statistic	Dummy	LS	RR	SPR
# of bets possible	820.0000	820.0000	820.0000	820.0000
# of bets made	342.0000	700.0000	658.0000	503.0000
Net # of bets won	-10.0000	6.0000	14.0000	57.0000
Winning percentage	0.4854	0.5043	0.5106	0.5567

3.6 What does SPR say about the NBA?

In the previous section, we evaluated the performance of SPR by testing its ability to predict the outcome of unseen games. In this section, we interpret the box score weights vector $\hat{\mathbf{z}}^{\vec{\lambda}}$ and player rating vector $\hat{\boldsymbol{\beta}}^{\vec{\lambda}}$ returned by SPR, and discuss what they say about the NBA.

3.6.1 Top 10 players in the league

Table 3.7. SPR Player Ratings

Player	$\hat{\boldsymbol{\beta}}_i^{\vec{\lambda}}$
James, LeBron	8.6006
Garnett, Kevin	8.2860
Paul, Chris	8.1899
Nowitzki, Dirk	7.7296
Howard, Dwight	7.4706
Gasol, Pau	6.9251
Odom, Lamar	6.5630
Hilario, Nene	6.2170
Evans, Jeremy	6.1725
Nash, Steve	6.0349

From $\hat{\boldsymbol{\beta}}^{\vec{\lambda}}$ we can extract a list of the top 10 players in the league who have played at least 10 possessions. Table 3.7 summarizes these results. This list contains some of the most prominent star players in the league (LeBron James, Chris Paul, Dirk Nowitzki, Dwight Howard), thus agreeing with common basketball wisdom. However, this ranking contradicts common basketball wisdom in the following ways:

1. The list noticeably omits Kobe Bryant, a player pop culture and common basketball wisdom considers one of the league's superstars. Yet SPR thinks very highly of Pau Gasol and Lamar Odom, two of Kobe Bryant's teammates who are individually credited far less for the success of the Lakers than Kobe is.

2. The list includes Nene Hilario and Jeremy Evans, players who are not considered by most to be amongst the top 10 players in the league.

3.6.2 Top 10 most underrated and overrated players

There are certain players in the NBA for whom their impact on the game seems to be far more (or less) than their raw box score production suggests. SPR allows

us to identify these players and quantify their impact by measuring the discrepancy between their SPR rating and their weighted box score ratings $\hat{\boldsymbol{\theta}}^{\vec{\lambda}}{}_i$.

We define the underrated vector $\mathbf{U}$ as

$$\mathbf{U} := \hat{\boldsymbol{\beta}}^{\vec{\lambda}} - \hat{\boldsymbol{\theta}}^{\vec{\lambda}}.$$

.

Similarly, we can examine which players impact the game much less than their box score production suggests with the vector $\mathbf{O} := -\mathbf{U}$.

Table 3.9 lists the top 10 most underrated/overrated players in the league relative to their box score production. For at least a few of these players, it is easy to understand why box scores alone do a poor job of capturing their impact:

- Andris Biedrens is a severe liability offensively, due to both his inability to score outside of 5 feet of the basket and poor free throw shooting. This makes it much more difficult for his teammates to score, since his defender can shift attention away from him and instead provide help elsewhere. Biedrens is also a liability defensively.

- Goran Dragic is a point guard with a scoring mentality. While a "shoot-first" point guard is not necessarily harmful to a team, if he doesn't do a good enough job in setting up his teammates and creating easy scoring opportunities for them, it hurts his team's ability to score.

3.6.3 Box score weights produced by SPR

The SPR regression also produces box score weights $\hat{\mathbf{z}}^{\vec{\lambda}}$ that tell us the relative importance of the different box score statistics. $\hat{\mathbf{z}}^{\vec{\lambda}}$ gives us a method to linearly transform box score data into the player effectiveness rating $\hat{\boldsymbol{\theta}}^{\vec{\lambda}}$ defined in Equation 3.6. For player j, the variable $\hat{\boldsymbol{\theta}}^{\vec{\lambda}}{}_j$ is a weighted linear combination of his box score statistics.

We can examine each entry of the vector $\hat{\mathbf{z}}^{\vec{\lambda}}$ to compare the relative importance of different box score variables like rebounds, assists and steals. Table 3.8 summarizes the results. We also display the relevant row in the box score matrix $\mathbf{R}$ for LeBron James, which we call $\mathbf{R}^T_{\text{Lebron James}}$.

Examining this table, we see that LeBron James made two point shots at a rate of 7.83 per 36 minutes, and attempted two point shots at a rate of 14.16 per 36 minutes. The corresponding weightings from $\hat{\mathbf{z}}^{\vec{\lambda},\text{rescaled}}$ are 3.38 and -1.54 respectively, suggesting that overall LeBron's rating from his two point shooting is $7.83 \times 3.38 + 14.16 \times -1.54 \approx 4.66$ points. In fact, from these weightings we can calculate that according to the SPR model all players in the league must hit their two point shots roughly 45% of the time for their rating from two point shooting to be non-negative.

Interestingly enough, a similar calculation reveals that three point shots must only be hit at a roughly 14% rate to break even. This is counterintuitive: naively one would believe that hitting two point shots q percent of the time should be equivalent to hitting three point shots $\frac{2}{3}q$ of the time. However, three point shooting increases the amount of spacing on the floor and perhaps missed three point shots are easier to rebound for the offensive team.

According to this interpretation of the $\hat{\mathbf{z}}^{\vec{\lambda}}$ variable turnovers are extremely costly, with the corresponding entry of $\hat{\mathbf{z}}^{\vec{\lambda},\text{rescaled}}$ equal to $-.76$. Thus, LeBron's turnover rate of 3.34 turnovers per 36 minutes hurts his rating box score rating by roughly 6.28 points.

Table 3.8: Box Score Weights

Statistic	Description	$\hat{\mathbf{z}}^{\vec{\lambda}}$	$\mathbf{R}^T_{\text{LeBron James}}$
2M	Per 36 Minute	3.38	7.83
2A	Per 36 Minute	-1.54	14.16
3M	Per 36 Minute	1.48	1.08
3A	Per 36 Minute	-0.21	3.28
FTM	Per 36 Minute	0.73	5.91
FTA	Per 36 Minute	-0.33	7.79
OR	Per 36 Minute	0.11	0.94
DR	Per 36 Minute	0.50	5.99
AS	Per 36 Minute	0.85	6.51
ST	Per 36 Minute	1.66	1.46
TO	Per 36 Minute	-1.88	3.34
BK	Per 36 Minute	0.86	0.58
PF	Per 36 Minute	-0.37	1.92
TC	Per 36 Minute	2.81	0.09
DQ	Per 36 Minute	6.98	0.00
P1	Boolean	-0.17	0.00
P2	Boolean	-0.71	0.00
P3	Boolean	0.29	1.00
P4	Boolean	1.65	0.00

3.7 Extending SPR by augmenting the box score

In this section, we discuss a possible extension to the SPR model.

The box score matrix $\mathbf{R}$ keeps track of statistics like rebounds, assists, and steals. However, one might imagine augmenting this basic box score matrix with products of raw statistics such as rebounds $\times$ assists, blocks $\times$ steals, turnovers $\times$ free throws

Table 3.9: Underrated/Overrated Players

Player	$\hat{\boldsymbol{\beta}}^{\vec{\lambda}}$	$\mathbf{R}\hat{\mathbf{z}}^{\vec{\lambda}} + \hat{\beta}_0^{\vec{\lambda}}\mathbf{1}_p$	Underrated
Dooling, Keyon	1.3531	-0.3840	1.7371
Watson, Earl	0.9973	-0.2920	1.2893
Aldridge, LaMarcus	5.0226	3.8456	1.1770
Ginobili, Manu	5.4336	4.2599	1.1738
Tolliver, Anthony	1.9424	0.7742	1.1682
Bosh, Chris	4.6311	3.4681	1.1630
Carter, Vince	1.4315	0.3376	1.0939
Collins, Jason	-2.4071	-3.4634	1.0563
Hill, George	2.0535	1.0201	1.0333
Bass, Brandon	2.5777	1.5792	0.9985

Player	$\hat{\boldsymbol{\beta}}^{\vec{\lambda}}$	$\mathbf{R}\hat{\mathbf{z}}^{\vec{\lambda}} + \hat{\beta}_0^{\vec{\lambda}}\mathbf{1}_p$	Overrated
Dragic, Goran	-2.1439	-0.4758	-1.6680
Marion, Shawn	0.7049	2.2120	-1.5071
Gortat, Marcin	2.7433	4.1681	-1.4249
Ellis, Monta	-0.1175	1.2634	-1.3810
Biedrins, Andris	0.9205	2.2408	-1.3203
Jefferson, Al	1.7706	3.0598	-1.2893
Felton, Raymond	1.4721	2.7513	-1.2792
Bell, Raja	-2.8779	-1.5992	-1.2787
Dudley, Jared	1.1287	2.3858	-1.2572
Law, Acie	-1.7759	-0.6247	-1.1512

made, etc. By capturing some of these product statistics and incorporating them into SPR, one might more accurately model the value of multifaceted players.

We expand the matrix $\mathbf{R}$ to include all pairwise product of the basic variables. If $\mathbf{R}$ is a p by d matrix, this leads to a p by $d + \binom{d}{2}$ matrix called

$$\text{Poly}(\mathbf{R}, 2).$$

Let us use the notations $\text{SPR}(\mathbf{R})$ and $\text{SPR}(\mathbf{R}, 2)$ to denote SPR with the box score matrices $\mathbf{R}$ and $\text{Poly}(\mathbf{R}, 2)$, respectively. Applying the cross-validation procedure described in Section 3.4.2 on the first 820 games of the 2010-2011 produces the regularization parameter

$$\vec{\lambda}_{CV,2}^{820} = (2^{-10}, 2^{-1}).$$

With this choice of parameter for the expanded box score matrix Poly($\mathbf{R}, 2$), we can then empirically compare its performance to that of the ordinary SPR algorithm using the basic box score matrix $\mathbf{R}$. Figure 3.2 demonstrates the result of this experiment. From this figure we see that the additional box score statistics don't seem to substantially improve performance. The histogram of the SPR($\mathbf{R}, 2$) errors are fairly similar to the SPR($\mathbf{R}$) errors. We can also study some of the empirical properties of the variable $\hat{E}_k$ for each approach. Table 3.2 summarizes the results.

As Table 3.2 indicates, SPR($\mathbf{R}, 2$) doesn't improve upon the predictive power of SPR($\mathbf{R}$). The fraction of games in which the wrong winner is guessed actually increases from 28.54% to 29.51%, the average absolute error in predicting games increases from 10.55 to 11.81.

A possible explanation for this poor statistical performance is that the pairwise interaction terms that SPR($\mathbf{R}, 2$) models are too many, and thus the model is overfitting.

3.8 Conclusion

We have introduced SPR, a powerful new statistical inference procedure for the NBA. We compared the statistical performance of our approach to an existing popular technique based on least squares and demonstrate empirically that SPR gives more predictive power. We also compare SPR to the Las Vegas lines and show that with sufficient training data, SPR seems to better predict the NBA than Vegas. We interpret the estimates produced by SPR and discuss what they suggest about who the best players in the NBA are, and which players are overrated or underrated. Finally, we discuss a possible extension to the SPR model.

Chapter 4

Conclusion

In this thesis, we have introduced a new technique for solving the RPCA problem, proved a convergence result for it, and demonstrated empirically that it performs well on synthetic datasets for which the rank of the low-rank component is small. We have also developed a new penalized regression model and demonstrated its usefulness for modeling player effectiveness in the NBA by comparing against existing techniques as well as the Las Vegas lines.

Bibliography

Atef Abdelkefi, Yuming Jiang, Wei Wang, Arne Aslebo, and Olav Kvittem. Robust Traffic Anomaly Detection with Principal Component Pursuit. In *Proceedings of the ACM CoNEXT Student Workshop*. ACM, 2010.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*, volume 3. 2011.

Tamas Budavari, Vivienne Wild, Alexander Szalay, Laszlo Dobos, and Ching W. Yip. Reliable Eigenspectra for New Generation Surveys. *Monthly Notices of the Royal Astronomical Society*, 2009.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust Principal Component Analysis? *J. ACM*, 58(3):11:1–11:37, June 2011.

Venkat Chandrasekaran, Pablo A. Parrilo, and Alan S. Willsky. Latent Variable Graphical Model Selection via Convex Optimization. *ArXiv e-prints*, August 2010.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-Sparsity Incoherence for Matrix Decomposition. *SIAM Journal on Optimization*, 21:572, 2011.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Annals of Statistics*, 32(2):407–499, 2004.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise Coordinate Optimization. *Annals of Applied Statistics*, 2:302–332, 2007.

Luigi Grippo and Marco Sciandrone. Globally Convergent Block-coordinate Techniques for Unconstrained Optimization. *Optimization Methods and Software*, 10: 587–637, 1999.

Luigi Grippo and Marco Sciandrone. On the Convergence of the Block Nonlinear Gauss-Seidel Method Under Convex Constraints. *Operations Research Letters*, 26: 127–136, 2000.

Peter Huber. *Robust Statistics*. Wiley, New York, 1974.

Qifa Ke and Takeo Kanade. Robust ℓ_1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, June 2005.

Justin Kubatko, Dean Oliver, Kevin Pelton, and Dan Rosenbaum. A Starting Point for Analyzing Basketball Statistics. *Journal of Quantitative Analysis in Sports*, 2007.

Su I. Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient ℓ_1 Regularized Logistic Regression. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*. 2006.

Zhouchen Lin, Minming Chen, Leqin Wu, and Yi Ma. The Augmented Lagrange Multiplier Method for Exact Recovery of Corrupted Low-Rank Matrices. *Mathematical Programming*, 2010.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*. Springer Netherlands, 1 edition, 2003.

Dean Oliver. *Basketball on Paper: Rules and Tools for Performance Analysis*. Potomac Books, 2004.

Michael J. D. Powell. On Search Directions for Minimization Algorithms. *Mathematical Programming*, 4(2):193–201, April 1973.

G. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, 1970.

N. Z. Shor, K. C. Kiwiel, and A. Ruszcaynski. *Minimization methods for non-differentiable functions*. Springer-Verlag New York, Inc., New York, NY, USA, 1985.

M. Stone. Cross-validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):pp. 111–147, 1974.

Fernando De La Torre and Michael J. Black. A Framework for Robust Subspace Learning. *International Journal of Computer Vision*, 54:2003, 2003.

Paul Tseng. Convergence of Block Coordinate Descent Method for Nondifferentiable Minimization. *J. Optim. Theory Appl.*, 109:475–494, 2001.

Lieven Vandenberghe and Stephen Boyd. Semidefinite Programming. *SIAM Review*, 38(1):49–95, 1996.

Tong Tong Wu and Kenneth Lange. Coordinate Descent Algorithms for Lasso Penalized Regression. *Annals of Applied Statistics*, 2:224–244, 2008.

Xiaoming Yuan and Junfeng Yang. Sparse and Low-rank Matrix Decomposition via Alternating Direction Methods. *Pacific Journal of Optimization (to appear)*, 2009.

Appendix A

Appendices

A.1 Equivalence of (2.3) and (2.4)

Observe that if (A, B) minimizes (2.3), then (A, B, A, B) is an optimal solution for (2.4). On the other hand, consider any optimal solution $(C, D, \tilde{C}, \tilde{D})$ for (2.4). It must be the case that $C = \tilde{C}$ and $D = \tilde{D}$, otherwise $f(C, D, C, D) < f(C, D, \tilde{C}, \tilde{D})$, contradicting the optimality of $(C, D, \tilde{C}, \tilde{D})$. Thus, if (C, D) does not minimize (2.3), then there exists some $(\hat{C}, \hat{D})$ which acheives a smaller value, implying that $f(\hat{C}, \hat{D}, \hat{C}, \hat{D}) < f(C, D, \tilde{C}, \tilde{D})$, a contradiction.

A.2 Techniques for solving (2.7)

The convex program (2.7) can be solved in $O(n)$ time, using either bisection or the subgradient method. We focus on the subgradient method in this example.

The objective function of (2.7) is subdifferentiable, with the subgradients at c of the form

$$\sum_{i=1}^{n} w_i z_i + 2(c - c_0), \tag{A.1}$$

where

$$z_i \in \partial |c - v_i| = \begin{cases} \{1\} & \text{if } c > v_i \\ [-1, 1] & \text{if } c = v_i \\ \{-1\} & \text{if } c < v_i. \end{cases}$$

The computation of (A.1) takes $O(n)$ time, and to get within ϵ of the optimal value of the program (2.7) takes a constant number of steps, with the constant dependent on ϵ.

A.3 Block Coordinate Descent Proof

Consider the optimization problem

$$\min_x f(x_1, x_2, \ldots, x_p) \text{ s.t. } x \in \mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \ldots \times \mathcal{X}_p, \tag{A.2}$$

with $\mathcal{X}_i$ closed and convex. Note that (2.5) is a special case of this formulation, with $p = 4$ and with $\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3, \mathcal{X}_4$ set to $\mathbb{R}^{m \times r}, \mathbb{R}^{r \times n}, \mathbb{R}^{m \times r}, \mathbb{R}^{r \times n}$ respectively.

One possible approach for solving the problem (A.2) is block coordinate descent (BCD), Algorithm 3. Observe that Algorithm 2 is a special case of Algorithm 3.

We want to prove that Algorithm 3 converges to a stationary point of (A.2). We use a proof technique similar to that of Bertsekas [1999], Grippo and Sciandrone [2000], Grippo and Sciandrone [1999], and Tseng [2001].

Algorithm 3 General Block Coordinate Descent.

1: $x^0 \leftarrow x^0_{init}$
2: **for** $k \in \{1, 2, \ldots\}$ **do**
3: **for** $i \in \{1, 2, \ldots, p\}$ **do**
4: $x_i^{k+1} \leftarrow \arg\min_{\xi \in \mathcal{X}_i} f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, \xi, x_{i+1}^k, \ldots, x_p^k)$
5: **end for**
6: **end for**

Proposition A.3.1. *Suppose that for each i, f is a strictly convex function of x_i, when the values of the other components of x are held constant. Let $\{x^k\}$ be the sequences generated by the BCD algorithm. Then every limit point of $\{x^k\}$ is a stationary point of f over $\mathcal{X}$.*

Proof. Given a limit point $\bar{x}$ of the BCD, there are two possibilities

1. f is not differentiable at $\bar{x}$. In which case, $\bar{x}$ is a stationary point by definition.
2. f is differentiable at $\bar{x}$. We shall show that

$$\nabla f(\bar{x})^T(x - \bar{x}) \geq 0 \; \forall x,$$

and thus $\bar{x}$ is a stationary point of f.

For Case 2, since $\mathcal{X}$ is the Cartesian product of sets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_p$ and

$$\nabla f(\bar{x})^T(x - \bar{x}) = \sum_i t_i.$$

then it is sufficient to show that

$$t_i := \nabla_i f(\bar{x})^T(\alpha_i - \bar{x}_i) \geq 0 \; \forall \alpha_i \in \mathcal{X}_i, \forall i. \tag{A.3}$$

Lemma A.3.2 proves (A.3). □

Lemma A.3.2. *Let $\bar{x}$ be a limit point of the BCD updates. Then we have that*

$$t_s := \nabla_s f(\bar{x})^T(\alpha - \bar{x}_s) \geq 0 \; \forall \alpha \in \mathcal{X}_s.$$

Proof. Let x^{k_j} be a subsequence of the BCD updates that converges to $\bar{x}$. From the definition of the BCD algorithm, we have that

$$f(x_1^{k_j+1}, \ldots, x_s^{k_j+1}, x_{s+1}^{k_j}, \ldots, x_p^{k_j}) \leq f(x_1^{k_j+1}, \ldots, \alpha, x_{s+1}^{k_j}, \ldots, x_p^{k_j}) \; \forall \alpha \in \mathcal{X}_s.$$

This is true simply because the algorithm optimizes each coordinate assuming that the others are fixed.

By Lemma A.3.3, $\lim_{j\to\infty} x_s^{k_j} = \bar{x}_s \forall s \in 1, 2, \ldots, p$. Using this and the continuity of f, we have that

$$f(\bar{x}) \leq f(\bar{x}_1, \ldots, \bar{x}_{s-1}, \alpha, \bar{x}_{s+1}, \ldots, \bar{x}_p) \ \forall \alpha \in \mathcal{X}_s. \tag{A.4}$$

Define the function $g(\alpha) := f(\bar{x}_1, \ldots, \bar{x}_{s-1}, \alpha, \bar{x}_{s+1}, \ldots, \bar{x}_p)$. Then (A.4) is equivalent to

$$g(\bar{x}_s) \leq g(\alpha) \ \forall \alpha \in \mathcal{X}_s.$$

In other words, $\bar{x}_s$ is a global minimum for the function g. Since we are considering Case 2 and are assuming that f is differentiable at $\bar{x}$, then g is differentiable at $\bar{x}_1$. Therefore

$$\nabla g(\bar{x}_s)^T(\alpha - \bar{x}_s) \geq 0 \ \forall \alpha \in \mathcal{X}_s.$$

Since $\nabla g(\bar{x}_s)^T = \nabla_s f(\bar{x})$, we have our desired result. □

Lemma A.3.3. *Let $\{x^k\}_{k=1}^{\infty}$ be the BCD updates. Let $\bar{x}$ be a limit point of this sequence, and $\{x^{k_j}\}_{j=1}^{\infty}$ be a subsequence converging to $\bar{x}$. Then $x_s^{k_j} \to \bar{x}_s \forall s \in 1, 2, \ldots, p$.*

Proof. Let us introduce the intermediate vectors produced by the BCD algorithm

$$\begin{aligned} z_0^k &:= x^k, \\ z_i^k &:= (x_1^{k+1}, \ldots, x_i^{k+1}, x_{i+1}^k, \ldots, x_p^k), \ i = 1, 2, \ldots, p. \end{aligned}$$

For $i \in 1, \ldots, p$ define $\epsilon_i^k := z_i^k - z_{i-1}^k$ and $\gamma_i^k := ||\epsilon_i^k||_2$.
Since $z_0^k = x^k \to \bar{x}$, the lemma is true if $\forall i \in 1, \ldots, p$, we have

$$\lim_{k\to\infty} \gamma_i^k = 0. \tag{A.5}$$

Suppose for the sake of a contradiction that there is some $i \in 1, \ldots p$ for which i is false. Without loss of generality, assume that i is the smallest value that violates (A.5).

Then there exists some $\gamma^* > 0$ such that a subsequence S of $\{k_j\}_{j=1}^{\infty}$ satisfies $\gamma_t \geq \gamma^* \forall t \in S$. We replace $\{k_j\}_{j=1}^{\infty}$ with this subsequence.

We can write $z_i^k = z^{i-1} + \gamma_i^k \frac{\epsilon_i^k}{\gamma_i^k}$. Since $\frac{\epsilon_i^k}{\gamma_i^k}$ is of length 1, it belongs to the compact set $\{x | ||x||_2 = 1\}$, and thus the sequence $\{\frac{\epsilon_i^k}{\gamma_i^k}\}_{k=1}^{\infty}$ has a limit point $\bar{\epsilon}$.

Let us replace $\{\frac{\epsilon_i^k}{\gamma_i^k}\}_{k=1}^{\infty}$ with a subsequence converging to $\bar{\epsilon}$.

We then have that $\forall \delta \in [0, \gamma^*]$

$$f(z_i^{k_j}) = f(z_{i-1}^{k_j} + \gamma_i^k \frac{\epsilon_i^k}{\gamma_i^k}) \leq f(z_i^{k_j} + \delta \frac{\epsilon_i^k}{\gamma_i^k}), \tag{A.6}$$

since the BCD algorithm produces $z_i^{k_j}$ such that minimizes f with all components other than the i^{th} entry fixed.

Since $\forall \delta \in [0, \gamma^*]$, $z_{i-1}^{k_j} + \delta \frac{\epsilon_i^k}{\gamma_i^k}$ is a convex combination of $z_{i-1}^{k_j}$ and $z_i^{k_j}$, and f is convex in its i^{th} argument, we have that

$$f(z_{i-1}^{k_j} + \delta \frac{\epsilon_i^k}{\gamma_i^k}) \leq \max \left\{ f(z_{i-1}^{k_j}), f(z_i^{k_j}) \right\} = f(z_{i-1}^{k_j}). \tag{A.7}$$

Combining (A.6) and (A.7), we have that

$$f(z_i^{k_j}) \leq f(z_{i-1}^{k_j} + \delta \frac{\epsilon_i^k}{\gamma_i^k}) \leq f(z_{i-1}^{k_j}) \ \forall \delta \in [0, \gamma^*]. \tag{A.8}$$

Since $z_{i-1}^{k_j} \to \bar{x}$ and f is continuous, then $f(z_{i-1}^{k_j}) \to f(\bar{x})$. Since $f(z_{i-1}^{k_j}) > f(z_i^{k_j}) > f(z_{i-1}^{k_{j+1}})$, then we have that $f(z_i^{k_j}) \to f(\bar{x})$.

Letting $j \to \infty$ in (A.8), we have

$$f(\bar{x} + \delta \bar{\epsilon}) = f(\bar{x}) \ \forall \delta \in [0, \gamma^*],$$

which is a contradiction of the fact that f is a strictly convex function with respect to its i^{th} argument. □

A.4 Tables used to generate Figures 2.1 and 2.2

m	$\log_{10}(\frac{\|\hat{\mathbf{L}}-\mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F})$	$\log_{10}(\frac{\|\hat{\mathbf{S}}-\mathbf{S}^*\|_F}{\|\mathbf{S}^*\|_F})$	Time (s)	Iterations
500	-0.31 (2.20)	-0.33 (2.30)	2.93 (0.78)	2.92 (0.44)
1000	-0.31 (2.18)	-0.33 (2.30)	7.41 (1.22)	3.02 (0.54)
2000	-0.31 (2.15)	-0.33 (2.30)	32.03 (7.26)	2.97 (0.45)
3000	-0.31 (2.17)	-0.33 (2.30)	67.48 (11.03)	2.94 (0.48)

Table A.1. Algorithm 2, $r = 1$. Average over 100 trials, standard deviation in parentheses.

A.5 The Cyclical Coordinate Descent Algorithm for Subspace Prior Regression

There are a variety of techniques for solving the convex program (3.5), including interior-point methods [Boyd and Vandenberghe, 2004], LARs [Efron et al., 2004], iteratively re-weighted least squares [Huber, 1974], approximating the ℓ_1 term with

m	$\log_{10}(\frac{\|\hat{\mathbf{L}}-\mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F})$	$\log_{10}(\frac{\|\hat{\mathbf{S}}-\mathbf{S}^*\|_F}{\|\mathbf{S}^*\|_F})$	Time (s)	Iterations
500	-0.11 (0.77)	-0.15 (1.06)	7.65 (3.63)	7.86 (0.99)
1000	-0.09 (0.65)	-0.14 (0.98)	35.28 (8.00)	8.32 (1.17)
2000	-0.10 (0.71)	-0.15 (1.08)	302.93 (77.99)	8.84 (1.13)
3000	-0.11 (0.78)	-0.17 (1.18)	1085.03 (236.89)	9.82 (1.26)

Table A.2. ADMM solver for the convex program (2.2), $r = 1$. Average over 100 trials, standard deviation in parentheses.

m	$\log_{10}(\frac{\|\hat{\mathbf{L}}-\mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F})$	$\log_{10}(\frac{\|\hat{\mathbf{S}}-\mathbf{S}^*\|_F}{\|\mathbf{S}^*\|_F})$	Time (s)	Iterations
500	-0.11 (0.80)	-0.14 (0.98)	177.38 (33.29)	12.88 (2.39)
1000	-0.12 (0.85)	-0.15 (1.06)	1520.03 (259.61)	14.90 (2.46)
2000	-0.13 (0.94)	-0.17 (1.17)	13679.21 (2346.40)	16.61 (2.56)

Table A.3. Algorithm 2, $r = 0.05m$. Average over 100 trials, standard deviation in parentheses.

m	$\log_{10}(\frac{\|\hat{\mathbf{L}}-\mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F})$	$\log_{10}(\frac{\|\hat{\mathbf{S}}-\mathbf{S}^*\|_F}{\|\mathbf{S}^*\|_F})$	Time (s)	Iterations
500	-0.14 (0.95)	-0.16 (1.14)	9.64 (3.89)	8.84 (1.13)
1000	-0.13 (0.89)	-0.16 (1.10)	60.24 (11.60)	8.84 (1.13)
2000	-0.13 (0.93)	-0.17 (1.16)	452.13 (79.99)	8.62 (1.17)

Table A.4. ADMM solver for the convex program (2.2), $r = 0.05m$. Average over 100 trials, standard deviation in parentheses.

a smooth function [Lee et al., 2006] the sub-gradient method [Shor et al., 1985], and Nesterov's proximal gradient method [Nesterov, 2003].

Ultimately, we found experimentally that cyclical coordinate descent (CCD) [Friedman et al., 2007, Wu and Lange, 2008] was the fastest for our problem.

The CCD method works by repeatedly optimizing the objective function viewed as a function of each variable with the others fixed. This idea gives a CCD algorithm for SPR, Algorithm 4.

A.5.1 Convergence of Algorithm 4

The correctness of this algorithm for minimizing the objective function (3.4) follows from Lemma A.5.1.

Lemma A.5.1. *Let* $\alpha_{hca}{}^*, z_0^*, \boldsymbol{\beta}^*, \boldsymbol{z}^* \in \arg\min g(\alpha_{hca}, \boldsymbol{\beta}, z_0, \boldsymbol{z}; \vec{\lambda})$.

Algorithm 4 CCDSPR($\mathbf{X}$, $\mathbf{Y}$, $\mathbf{R}$, $\vec{\lambda}$, T)

$\alpha_{\text{hca}}(0) \leftarrow 0$, $z_0(0) \leftarrow 0$, $\boldsymbol{\beta}(0) \leftarrow \mathbf{0}_p$, $\mathbf{z}(0) \leftarrow \mathbf{0}_d$
for $i \in \{1, 2, \ldots, T\}$ **do**
  {Optimize α_{hca} with all other variables fixed}
  {Optimize z_0 with all other variables fixed}
  for $k \in \{1, 2, \ldots, p\}$ **do**
    {Optimize $\boldsymbol{\beta}_k$ with all other variables fixed}
  end for
  for $\ell \in \{1, 2, \ldots, d\}$ **do**
    {Optimize $\mathbf{z}_\ell$ with all other variables fixed}
  end for
end for
return $\alpha_{\text{hca}}(T), z_0(T), \boldsymbol{\beta}(T), \mathbf{z}(T)$

Then

$$\lim_{T\to\infty} g(\alpha_{hca}(T), z_0(T), \boldsymbol{\beta}(T), \boldsymbol{z}(T)) = g({\alpha_{hca}}^*, z_0^*, \boldsymbol{\beta}^*, \boldsymbol{z}^*).$$

Furthermore, when ${\alpha_{hca}}^*, z_0^*, \boldsymbol{\beta}^*, \boldsymbol{z}^*$ *is the unique global minimum of* g*,*

$$\lim_{T\to\infty} (\alpha_{hca}(T), z_0(T), \boldsymbol{\beta}(T), \boldsymbol{z}(T)) = ({\alpha_{hca}}^*, z_0^*, \boldsymbol{\beta}^*, \boldsymbol{z}^*).$$

Proof. This is a direct consequence of Proposition 5.1 of Tseng [2001]. In particular, identify f_0 and $f_i, i = 1, \ldots, p$ of Proposition 5.1 with $L_{\text{quadratic}}(\alpha_{\text{hca}}, \boldsymbol{\beta}) + \lambda_2||\boldsymbol{\beta} - z_0\mathbf{1}_p - \mathbf{R}\mathbf{z}||_2^2$ and $\lambda_1|\boldsymbol{\beta}_i|, i = 1, \ldots, p$, respectively. We observe that

- Assumption B1 of Tseng [2001] is satisfied, since f_0 is continuous.
- Assumption B2 of Tseng [2001] is satisfied, since f is convex and non-constant on line segments.
- Assumption B3 is satisfied, since $f_i, i = 1, \ldots, p$ are continuous.
- Assumption C2 is trivially satisfied.

Therefore, the conditions of Proposition 5.1 of Tseng [2001] are satisfied for Algorithm 4 on the objective function (3.3).

Since (3.4) has at least one global minimum and is convex, then we further conclude that the limit points of Algorithm 4 are global minima.

□

A.5.2 Computing the updates for Algorithm 4

The updates for $\alpha_{\text{hca}}(i), z_0(i), \boldsymbol{\beta}(i), \mathbf{z}(i)$ can be computed in closed form.

To compute $\alpha_{\text{hca}}(i)$, we can optimize the objective function g viewed as a function only of the decision variable α_{hca} by taking the derivative and setting it to zero.

This yields the update

$$\alpha_{\text{hca}} \leftarrow \frac{1_n^T \text{Diag}(w) \left[\mathbf{Y} - \mathbf{X}\beta\right]}{1_n^T w}.$$

Similarly, for $z_0(i)$ we get the update

$$z_0 \leftarrow \frac{1}{p} \mathbf{1}_p^T \left[\beta - \mathbf{R}z\right].$$

For $\mathbf{z}$, we simply get the least squares updates:

$$\mathbf{z} \leftarrow (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T[\boldsymbol{\beta} - z_0\mathbf{1}_p].$$

Updates for $\boldsymbol{\beta}$

We next derive a closed-form expression for the updates for $\boldsymbol{\beta}_i$. To do so, we need Lemma A.5.2.

Lemma A.5.2 (One-variable lasso is soft-thresholding)**.** *Let* $h(x) := \frac{1}{2}Ax^2 - Bx + C + \tau|x|, x \in \mathbb{R}$. *Suppose that* $A > 0$. *The solution of*

$$\min_{x \in \mathbb{R}} h(x) \tag{A.9}$$

is

$$x^* = \frac{S_\tau(B)}{A}$$

where

$$S_\tau(x) := \begin{cases} 0 & \text{if } |x| \leq \tau \\ x - \tau & \text{if } x > 0 \text{ and } |x| > \tau \\ x + \tau & \text{if } x < 0 \text{ and } |x| > \tau, \end{cases}$$

is the soft-thresholding function with threshold τ.

See Section A.5.2 for a proof of this.
This lemma is useful, as it allows us to immediately write the updates for $\boldsymbol{\beta}_k(i)$. First, let us identify A and B for $\boldsymbol{\beta}_k(i)$. Differentiating g_s, we get

$$\begin{aligned}
\partial_{\boldsymbol{\beta}_t} g_s &= \partial_{\boldsymbol{\beta}_t}(L_{\text{quadratic}}(\alpha_{\text{hca}}, \boldsymbol{\beta}) + \lambda_2 ||\boldsymbol{\beta} - z_0 \mathbf{1}_p - \mathbf{R}\mathbf{z}||_2^2) \\
&= \partial_{\boldsymbol{\beta}_t} \left[\frac{1}{\mathbf{1}_n^T \vec{w}} \sum_i w_i (Y_i - \alpha_{\text{hca}} - \mathbf{X}_i^T \boldsymbol{\beta})^2 \right] + \lambda_2 \partial_{\boldsymbol{\beta}_t} \sum_j (\boldsymbol{\beta}_j - z_0 - \mathbf{R}_j^T z)^2 \\
&= (\frac{1}{\mathbf{1}_n^T \vec{w}} \sum_i w_i \partial_{\boldsymbol{\beta}_t} (Y_i - \alpha_{\text{hca}} - \mathbf{X}_i^T \boldsymbol{\beta})^2 + \lambda_2 \sum_j \partial_{\boldsymbol{\beta}_t} (\boldsymbol{\beta}_j - z_0 - \mathbf{R}_j^T z)^2 \\
&= \frac{1}{\mathbf{1}_n^T \vec{w}} \sum_i 2 w_i \mathbf{X}_{it} (-Y_i + \alpha_{\text{hca}} + \mathbf{X}_i^T \boldsymbol{\beta}) + \lambda_2 2 (\boldsymbol{\beta}_t - z_0 - \mathbf{R}_t^T z) \\
&= C \sum_i w_i \mathbf{X}_{it} (-Y_i + \alpha_{\text{hca}} + \mathbf{X}_i^T \boldsymbol{\beta}) + \lambda_2 2 (\boldsymbol{\beta}_t - z_0 - \mathbf{R}_t^T z) \\
&= C (\mathbf{X} e_t)^T W (-Y + \alpha_{\text{hca}} \mathbf{1}_n + \mathbf{X}\boldsymbol{\beta}) + \lambda_2 2 (\boldsymbol{\beta}_t - z_0 - \mathbf{R}_t^T z) \\
&= C (\mathbf{X} e_t)^T W (-Y + \alpha_{\text{hca}} \mathbf{1}_n + \mathbf{X}[\boldsymbol{\beta} - e_t \boldsymbol{\beta}_t + e_t \boldsymbol{\beta}_t] + \lambda_2 2 (\boldsymbol{\beta}_t - \theta_t) \\
&= C (\mathbf{X} e_t)^T W (\kappa + \mathbf{X}[e_t \boldsymbol{\beta}_t]) + \lambda_2 2 (\boldsymbol{\beta}_t - \theta_t)
\end{aligned}$$

where

$$\begin{aligned}
C &:= \frac{2}{\mathbf{1}_n^T \vec{w}}, \\
\theta_t &:= (z_0 \mathbf{1}_p + \mathbf{R}\mathbf{z})^T e_t, \\
\kappa &:= -Y + \alpha_{\text{hca}} \mathbf{1}_n + \mathbf{X}[\boldsymbol{\beta} - e_t \boldsymbol{\beta}_t].
\end{aligned}$$

The constant term (with respect to $\boldsymbol{\beta}_t$) of the above expression is

$$D := C (\mathbf{X} e_t)^T W \kappa - 2\lambda_2 \theta_t.$$

The linear term is

$$\begin{aligned}
C (\mathbf{X} e_t)^T W \mathbf{X} e_t \boldsymbol{\beta}_t + 2\lambda_2 \boldsymbol{\beta}_t &= \left[C e_t^T \mathbf{X}^T W \mathbf{X} e_t + 2\lambda_2 \right] \boldsymbol{\beta}_t \\
&= E \boldsymbol{\beta}_t,
\end{aligned}$$

where

$$E := C e_t^T \mathbf{X}^T W \mathbf{X} e_t + 2\lambda_2.$$

From this we conclude that for $\boldsymbol{\beta}_k(t)$

$$\begin{aligned}
A &= E \\
B &:= -D.
\end{aligned}$$

So, we have the update equation

$$\boldsymbol{\beta}_k(i) \leftarrow \frac{\mathrm{S}_{\lambda_1}(B)}{A}.$$

Proof of Lemma A.5.2

Proof. The subdifferential of $h(x)$ [Rockafellar, 1970] is the set

$$\begin{aligned}\partial h(x) &:= \sum_{k=1}^{K} a_k(a_k x - b_k) + \tau\partial|x| \\ &= xA - B + \tau\partial|x|,\end{aligned}$$

where

$$\begin{aligned}A &:= \sum_{k=1}^{K} a_k^2 \\ B &:= \sum_{k=1}^{K} b_k \\ \partial|x| &:= \begin{cases}\{\text{sign}(x)\} & \text{if } x \neq 0 \\ [-1,1] & \text{otherwise.}\end{cases}\end{aligned}$$

From the theory of convex analysis [Rockafellar, 1970] x^* is the solution of (A.9) if and only if

$$0 \in \partial h(x^*). \tag{A.10}$$

The set $\partial h(x^*)$ behaves differently depending on the value of x^*. When $x^* \neq 0$, then

$$0 \in \partial h(x^*) = \{x^*A - B + \tau\,\text{sign}(x^*)\},$$

which is equivalent to $x^* = \frac{B-\tau\,\text{sign}(x^*)}{A}$. However, when $x^* = 0$, then

$$0 \in \partial h(x^*) = \{x^*A - B + \tau[-1,1]\}.$$

We use this observation to deal with the following two cases:

1. Suppose that $\tau \geq |B|$. Then

   (a) If $x^* \neq 0$, then

   $$\begin{aligned}\tau \geq |B| &= |x^*A + \tau\,\text{sign}(x^*)| \\ &= x^*A + \tau,\end{aligned}$$

   since at least one $a_k \neq 0$, then $A > 0$. This is a contradiction. Therefore $x^* \neq 0$ cannot be a solution when $\tau \geq |B|$.

(b) If $x^* = 0$, then

$$0 \in -B + \tau[-1, 1] = [-\tau - B, \tau - B],$$

which is true.

2. Suppose that $\tau < |B|$. Then

   (a) If $x^* \neq 0$, then $B - \tau \operatorname{sign}(x)$ has the same sign as B. Since A is positive, then x^* has the same sign as B. So the choice of $x^* = \frac{B - \tau \operatorname{sign}(B)}{A}$ satisfies the required sub-gradient optimality condition (A.10) without contradiction.

   (b) If $x^* = 0$, then $0 \in -B + \tau[-1, 1] = [-\tau - B, \tau - B]$, which is a contradiction.

Thus,

1. $\tau \geq |B| \implies x^* = 0$,

2. $\tau < |B| \implies x^* = \frac{\mathrm{S}_\tau(B)}{A}$.

These two cases can be summarized by $x^* = \frac{\mathrm{S}_\tau(B)}{A}$, as desired. □