Best Subset Selection in Linear Regression Analysis

Ronald de Boer

August 3, 2015

Bachelor thesis
Supervisor: dr. Neil Walton
Abstract

For the sake of interpretation we often want to reduce the number of predictors in a linear regression model. This thesis presents some of the techniques used to select a best subset of size $k$ from a set of $p$ predictors with $n$ observations. In the last chapter we review an article of Dimitris Bertsimas, which presents the best subset problem as a mixed integer optimization (MIO) problem. We will treat one of the MIO formulations and we examine one of the algorithms in full detail. In the optimization chapter we will develop some theory around integer optimization needed for the MIO formulation and in the linear regression chapter we will develop some theory on linear regression and especially on techniques to solve the best subset selection problem.
Contents

1. Introduction 4

2. Optimization 5
   2.1. Convex Sets, Polyhedra and Polytopes .......................... 5
   2.2. Gradient Descent Method ........................................... 10
       2.2.1. Projected Gradient Descent .................................. 13
   2.3. Linear Programming .................................................. 14
   2.4. Integer Programming .................................................. 15
   2.5. Mixed Integer Programming ......................................... 16

3. Linear Regression Analysis 18
   3.1. Estimation of Regression Coefficients .............................. 18
       3.1.1. The Gauss-Markov Theorem ..................................... 21
   3.2. Best Subset Selection ................................................ 22
       3.2.1. The Determination Coefficient .................................. 22
       3.2.2. t-test ............................................................. 23
       3.2.3. F-test ............................................................. 23
       3.2.4. Forward-Stepwise Selection ..................................... 24
       3.2.5. Regression Shrinkage and Selection via the Lasso .......... 25

4. Best Subset Problem via Mixed Integer Optimization 27
   4.1. Formulation ............................................................ 27
   4.2. The Algorithm ........................................................ 29
   4.3. Convergence Analysis ................................................ 32
   4.4. Application ........................................................... 36

5. Conclusion 37

A. Popular Summary 39
1. Introduction

Linear regression is about predicting an quantitative outcome, such as body fat percentage or stock prices, based on predictors (in the body fat percentage case, the predictors would be for example waist circumference). We usually start with a set of training data and we build a linear model using this dataset.

An important problem is the selection of subsets of regression variables, see [9]. From the set of predictors we would like to select only those which exhibits the strongest effect on the model. This often improves the prediction accuracy and the interpretability.

This thesis is about selecting the ‘best’ predictors for the data based on an article Best Subset Selection via a Modern Optimization Lens by Dimitris Bertsimas [1]. In this article the best subset problem is reformulated as a mixed integer optimization (MIO) problem, which can then be solved using a MIO solver. Before we can address this paper we will need to develop some theory. This is done in the Optimization and the Linear Regression Analysis chapters.

In the Optimization chapter we will give some basic definitions and theorems about combinatorial optimization, mostly based on the definition of convexity. Then we will develop the gradient descent algorithm, which minimizes a convex function by going in the negative direction of the gradient. We also show when the gradient descent algorithm converges and how to extend this to a projected gradient descent method. After that we will delve the theory of linear and integer programming and give the definitions of mixed integer programming problems needed in the last chapter.

In the Linear Regression Analysis chapter we develop the linear regression model and show how to estimate the regression coefficients using least squares estimation. Then we prove the Gauss-Markov Theorem from which we derive that the least squares estimator is the best linear unbiased estimator. After that we will start developing the theory about the best subset selection problem in linear regression analysis. We propose two methods; a discrete method called Forward-Stepwise selection and a continuous method called the lasso.

Central to the last chapter is the paper by Bertsimas. From this paper we have selected one of the five MIO formulations and show how to calculate the bound for it. Bertsimas also proposes an algorithm which we will examine and do a convergence analysis of it. Lastly we will give a brief overview of the rest of the paper and the performance claims in it.
2. Optimization

In this chapter we will develop some theory around mathematical optimization and in particular convex and combinatorial optimization.

A mathematical optimization problem has the form

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \quad \text{subject to} \quad f_i(x) & \leq b_i, i = 1, \ldots, m \quad \text{over} \quad x \in \mathbb{R}^n
\end{align*}
\]

(2.1)

Here the function \( f_0 : \mathbb{R}^n \to \mathbb{R} \) is the objective function, \( x = (x_0, \ldots, x_n) \) the optimization variable, the functions \( f_i : \mathbb{R}^n \to \mathbb{R} \) are the constraint functions and the constants \( b_1, \ldots, b_n \) are the bounds for the constraints. The domain of the optimization problem is defined as

\[ D = \bigcap_{i=0}^{m} \text{dom} f_i. \]

A point \( x \in D \) is called a feasible point if it satisfies the constraints \( f_i(x) \leq b_i \) for all \( i = 1, \ldots, m \). The optimal value \( p^* \) of our optimization problem is defined as

\[ p^* = \inf \{ f_0(x) \mid f_i(x) \leq b_i, i = 1, \ldots, m \}. \]

A solution vector \( x^* \) to problem (2.1) is called optimal if it is feasible and \( f_0(x^*) = p^* \). The optimization problem is called a linear program if the objective and constraint functions are linear. If the optimization problem is not linear, it is called a nonlinear program. If the objective function is a quadratic form, then the nonlinear program is called a quadratic program.

A large class of optimization problems are called convex optimization problems. This is a problem in which the objective and the constraint functions are convex, meaning that they satisfy the inequality

\[ \forall x_1, x_2 \in X \subseteq \text{dom} f, \forall t \in [0, 1] : f(t x_1 + (1-t) x_2) \leq t f(x_1) + (1-t) f(x_2). \]

Comparing this inequality with the definition of linear functions, we see that convexity applies to a more general class of functions, thus every linear program is a convex optimization problem.

2.1. Convex Sets, Polyhedra and Polytopes

In this section we will define convex sets, polyhedra and polytopes. One of the basic properties of closed convex sets is that any point outside the convex set can be separated by a hyperplane.
Definition 2.1 (Convex set). A set $C \subseteq \mathbb{R}^n$ is called convex if for all $x, y \in C$ and for any $0 \leq \lambda \leq 1$, the convex combination $\lambda x + (1 - \lambda)y$ is again in $C$.

Theorem 2.2. The intersection of arbitrary many convex sets is again convex.

Proof. Let $\{C_i\}_{i \in \alpha}$ be collection of convex subsets of $\mathbb{R}^n$ and define $C = \bigcap_{i \in \alpha} C_i$. Let $x, y \in C$, then $x, y \in C_i$ for all $i$ and for all $0 \leq \lambda \leq 1$ the convex combination $\lambda x + (1 - \lambda)y \in C_i$ and hence also in $C$. □

Definition 2.3 (Convex hull). Let $X \subseteq \mathbb{R}^n$ be a set. The convex hull of $X$, denoted $\text{conv. hull}(X)$, is the smallest convex set containing $X$.

Since $\mathbb{R}^n$ itself is convex, for a subset $X \subseteq \mathbb{R}^n$ it follows from Theorem 2.2 that the convex hull of $X$ always exists.

Definition 2.4 (Hyperplane). A set $H \subseteq \mathbb{R}^n$ is called a hyperplane if there exists a nonzero vector $c \in \mathbb{R}^n$ and a scalar $\delta$ such that

$$H = \{x \in \mathbb{R}^n \mid c^T x = \delta\}.$$ 

The hyperplane $H$ separates the point $x$ and the set $C$ if $x$ and $C$ are in different components of $\mathbb{R}^n/H$.

Theorem 2.5 (Hyperplane Separation Theorem). Let $C \subseteq \mathbb{R}^n$ be a nonempty, closed and convex set and let $x \notin C$. Then there exists a hyperplane separating $x$ and $C$.

Proof. Since $C$ is a nonempty, closed and convex subset of the Hilbert space $\mathbb{R}^n$, by Theorem 3.32 from [12], there exists a unique $y \in C$ that minimizes $\|x - y\|$. Now define $c := x - y$ and $\delta := \frac{1}{2}(|x|^2 - |y|^2)$. We will prove that $c^T x > \delta$ and $c^T z < \delta$ for all $z \in C$.

First we have that

$$c^T x = (x - y)^T x > (x - y)^T x - \frac{1}{2}\|x - y\|^2$$

$$= \|x\|^2 - y^T x - \frac{1}{2}\|x\|^2 + y^T x - \frac{1}{2}\|y\|^2$$

$$= \frac{1}{2}\|x\|^2 - \frac{1}{2}\|y\|^2 = \delta.$$

This proves that $c^T x > \delta$.

Now suppose that there exists a $z \in C$ such that $c^T z \geq \delta$. We have that $c^T y < c^T y + \frac{1}{2}\|c\|^2 = \delta$, so

$$c^T (z - y) = c^T z - c^T y > \delta - \delta = 0.$$ 

Also $\|z - y\|^2$ is positive ($z \neq y$, because $c^T z > c^T y$), so there exists a $\lambda$ with

$$0 < \lambda < \min \left(1, \frac{2\lambda}{\|z - y\|^2}\right).$$
Now define $w := \lambda z + (1 - \lambda)y$. This is a convex combination of points from $C$, so it belongs to $C$. Then it follows that
\[
\|w - x\|^2 = \|\lambda(z - y) + (y - x)\|^2 = \|\lambda(z - y) - c\|^2 = \lambda^2\|z - y\|^2 - 2\lambda c^T(z - y) + \|c\|^2 \\
< \lambda(2c^T(z - y) - 2c^T(z - y)) + \|c\|^2 = \|c\|^2 = \|z - y\|^2.
\]

This implies that the point $w$ has smaller distance to $x$ than $y$. This is a contradiction and hence there does not exist a $z \in C$ such that $c^Tz \geq \delta$.

**Definition 2.6** (Halfspace). A set $F \subseteq \mathbb{R}^n$ is called a halfspace if there exists a nonzero vector $c \in \mathbb{R}^n$ and a scalar $\delta$ such that
\[
F = \{x \in \mathbb{R}^n \mid c^T x \leq \delta\}.
\]

**Theorem 2.7.** A halfspace $F$ is a closed convex set.

**Proof.** We first prove that a halfspace is convex. Let $x, y \in F$, then $c^T x \leq \delta$ and $c^T y \leq \delta$. For $0 \leq \lambda \leq 1$ we have that $\lambda c^T x + (1 - \lambda)c^T y \leq \lambda \delta + (1 - \lambda)\delta \leq \delta$. So $\lambda x, (1 - \lambda)y \in F$ and hence
\[
c^T(\lambda x + (1 - \lambda)y) = \lambda c^T x + (1 - \lambda)c^T y \leq \lambda \delta + (1 - \lambda)\delta = \delta,
\]
so $\lambda x + (1 - \lambda)y \in F$ which proves that $F$ is a convex set.

To show that $F$ is closed, consider $(x_i)_{i \in \mathbb{N}} \subset F$ such that $\lim_{i \to \infty} x_i = x \in \mathbb{R}^n$. By continuity of the inner product we have
\[
c^T x = c^T \left( \lim_{i \to \infty} x_i \right) = \lim_{i \to \infty} c^T x_i \leq \left( \lim_{i \to \infty} \delta \right) = \delta.
\]
This shows that $x \in F$ and hence $F$ is closed.

**Theorem 2.8.** Let $C \subseteq \mathbb{R}^n$. Then $C$ is a closed convex set if and only if $C = \cap \mathcal{F}$ for some collection $\mathcal{F} := \{F_i\}_{i \in \alpha}$ of halfspaces.

**Proof.** Suppose that $C$ is a closed convex set. Take a point $x_1 \notin C$, then the Hyperplane Separation Theorem implies that there exists a halfspace $F_1$ such that $C \subseteq F_1$ and $x_1 \notin F_1$. For a point $x_2$ different from $x_1$ we obtain a halfspace $F_2$ with $C \subseteq F_2$ and $x_2 \notin F_2$. Then $C \subseteq F_1 \cap F_2$ and $x_1, x_2 \notin F_1 \cap F_2$. If we do this for all points $x_i \notin C$ we get a collection $\mathcal{F}$ of halfspaces such that $x_i \notin \cap \mathcal{F}$ for all $x_i \notin C$ and hence $C = \cap \mathcal{F}$.

Now suppose that $C = \cap \mathcal{F}$ for some collection of halfspaces $F_i$. Then $C \subseteq F_i$ for all $i$. Since the intersection of convex sets is again a convex set and all halfspaces are convex, the set $C$ is convex. The same argument proves that $C$ is closed.

**Definition 2.9** (Polyhedron). Let $P \subseteq \mathbb{R}^n$ be a closed convex set. Then $P = \cap \mathcal{F}$ for some collection $\mathcal{F}$ of halfspaces. If $P$ equals the intersection of a finite number of halfspaces, then $P$ is called a polyhedron.
Theorem 2.10. Let \( P \subseteq \mathbb{R}^n \). Then \( P \) is a polyhedron if and only if there exists a \( m \times n \) matrix \( A \) and a vector \( b \in \mathbb{R}^m \) such that \( P = \{ x \in \mathbb{R}^n \mid Ax \leq b \} \).

Proof. Let \( P \) be a polyhedron. Then \( P = F_1 \cap \ldots \cap F_m \) for some \( m \) and halfspaces \( F_i \).

Per definition of a halfspace

\[
F_i = \{ x \in \mathbb{R}^n \mid a_i^T x \leq b_i \}
\]

for some vector \( a_i \) and a scalar \( b_i \). Define the matrix \( A := (a_1 \mid \ldots \mid a_m) \) and the vector \( b := (b_1 \ldots b_m)^T \), then \( P = \{ x \in \mathbb{R}^n \mid Ax \leq b \} \).

Now suppose that there exists a \( m \times n \) matrix \( A \) and a vector \( b \in \mathbb{R}^m \) such that \( P = \{ x \in \mathbb{R}^n \mid Ax \leq b \} \). For each column vector \( a_i \) from \( A \) we have that \( a_i^T x \leq b_i \). Then define \( F_i := \{ x \in \mathbb{R}^n \mid a_i^T x \leq b_i \} \). This gives us a finite collection of halfspaces. Then set \( P := F_1 \cap \ldots \cap F_m \) and the result follows. \( \square \)

Definition 2.11 (Polytope). A polyhedron \( P \) is called a polytope if it is bounded.

Definition 2.12 (Vertex). Let \( C \) be a convex set. A point \( v \in C \) is called a vertex of \( C \) if \( v \) is not a convex combination of two other points in \( C \).

Definition 2.13 (Convex cone, polyhedral cone). A set \( C \subseteq \mathbb{R}^n \) is a convex cone, or cone, if for any \( p, q \in C \) and any scalars \( \lambda, \mu \geq 0 \) that \( \lambda p + \mu q \in C \). For any subset \( X \subseteq \mathbb{R}^n \), \( \text{cone}(X) \) is the smallest cone containing \( X \). A cone is finitely generated if there exists some finite set of vectors \( p_1, \ldots, p_n \in C \) such that for any \( p \in C \), there exists scalars \( \lambda_1, \ldots, \lambda_n \geq 0 \) with \( p = \sum_{i=1}^n \lambda_i p_i \). A polyhedral cone is a polyhedron of type \( \{ x \in \mathbb{R}^n \mid Ax \leq 0 \} \).

The following theorem shows that a polyhedral cone is finitely generated.

Theorem 2.14. Let \( C = \{ x \in \mathbb{R}^n \mid Ax \leq 0 \} \) be a nonempty polyhedral cone, then \( C \) is generated by a subset of the set of solutions to the system \( My = b \), where \( M \) consists of \( n \) linearly independent rows of \( (A^T) \) for some indentity matrix \( I \) and \( b = \pm e_j \) for some unit vector \( e_j \).

Proof. Let \( A \) be a \( m \times n \) matrix and consider the system \( My = b \) mentioned above. Let \( y_1, \ldots, y_t \in C \) be a subset of solutions to this system. We will prove that \( C \) is generated by \( y_1, \ldots, y_t \).

Suppose that \( C = \{ x \in \mathbb{R}^n \mid Ax = 0 \} \), which means that \( C \) is a linear subspace. Denote \( A' \) as the maximal set of linearly independent row vectors of \( A \). Let \( I' \) consist of some rows of \( I \) such that \( (A') \) is nonsingular and square. It follows that \( C \) is generated by the solutions of \( (A')^T x = (0)_j \), for some \( b = \pm e_j \) and \( j = 1, \ldots, \dim(C) \).

We will use induction on the dimension of \( C \) to prove the theorem. If \( C \) is not a linear subspace, choose a row \( a \) from \( A \) and a submatrix \( A' \) such that the rows of \( (A')^T \) are linearly independent and \( \{ x \in \mathbb{R}^n \mid A'x = 0, ax < 0 \} \subseteq C \). By the construction in the first part of the proof there is an index \( s \in \{1, \ldots, t\} \) such that \( A'y_s = 0 \) and \( ay_s = -1 \). Now given an arbitrary \( p \in C \), let \( a_1, \ldots, a_m \) be the rows of \( A \) and define

\[
\mu := \max \left\{ \frac{a_i p}{a_i y_s} \mid a_i y_s < 0 \right\}.
\]
Since $p \in C = \{x \mid Ax \leq 0\}$, $a_i p \leq 0$ and $a_i y_s < 0$ so $\mu \geq 0$. Let $k$ be an index where the maximum is attained and consider $p' := p - \mu y_s$. If we set $C' := \{x \in C \mid a_k x = 0\}$ we see that $a_k p' = a_k(p - \mu y_s) = 0$ and hence $p' \in C'$.

Now because $a_k y_s < 0$ and $y_s \in C$, $y_s \not\in C'$ so $\dim(C') = \dim(C) - 1$. By induction, $C'$ is generated by a subset of $y_1, \ldots, y_t$, implying that $p' = \sum_{i=1}^t \lambda_i y_i$ for nonnegative scalars $\lambda_1, \ldots, \lambda_t$. By setting $\lambda'_i := \lambda_s + \mu$ and $\lambda'_i := \lambda_i$ for $i \neq s$, we have that $p = p' + \mu y_s = \sum_{i=1}^t \lambda'_i y_i$. \(\square\)

**Definition 2.15 (Linear halfspace).** Let $F \subseteq \mathbb{R}^n$. Then $F$ is called a linear halfspace if $F = \{x \in \mathbb{R}^n \mid c^T x \leq 0\}$ for some nonzero vector $c$.

**Lemma 2.16.** Let $C \subseteq \mathbb{R}^n$.

1. If $C$ is a finitely generated convex cone, then it is closed.

2. $C$ is a closed convex cone if and only if $C$ equals the intersection of some arbitrary collection of linear halfspaces $\mathcal{F}$.

**Theorem 2.17 (Farkas’ Lemma).** The system $Ax = b$ has a nonnegative solution if and only if there is no vector $y$ satisfying $y^T A \geq 0$ and $y^T b < 0$

**Proof.** First, suppose that $Ax = b$ has a nonnegative solution $x_0$ and suppose that there does exist a vector $y$ such that $y^T A \geq 0$ and $y^T b < 0$. Then we have that

$$0 > y^T b = y^T (Ax_0) = (y^T A) x_0 \geq 0,$$

which is a contradiction. So no such $y$ exists.

We will prove the other direction by contraposition. Let $a_1, \ldots, a_n$ be the columns of $A$ and suppose that $Ax = b$ does not have a nonnegative solution. Let $C := \text{cone}\{a_1, \ldots, a_n\}$. Then $b \not\in C$, since this implies that there do not exist scalars $x_1, \ldots, x_n \geq 0$ such that $b = \sum_{i=1}^n a_i x_i$. By Lemma 2.16.1. $C$ is a closed cone, so by 2.16.2. there exists a linear halfspace $F$ such that $C \subseteq F$ and $b \not\in F$. Looking at the complement of $F$, there exists a vector $y$ such that $y^T b < 0$ and $y^T x' \geq 0$ for all $x' \in C$ and in particular $c^T a_i \geq 0$ for all $i$. This proves that if no such vector $y$ exists, the system $Ax = b$ has a nonnegative solution. \(\square\)

The next two theorems characterize a polyhedron and a polytope.

**Theorem 2.18.** A cone $P$ is polyhedral if and only if it is finitely generated.

**Proof.** Let $P$ be a polyhedral cone. If it is empty, then it surely is finitely generated. If it is not empty, it follows from Theorem 2.14 that it is finitely generated.

Now let $P$ be a cone which is generated by a set $a_1, \ldots, a_t$. Let $A$ be a matrix with rows $a_1, \ldots, a_t$ and define the cone $D := \{x \in \mathbb{R}^n \mid Ax \leq 0\}$. By Theorem 2.14, $D$ is generated by some set $b_1, \ldots, b_s$. We will prove that $P = \{x \in \mathbb{R}^n \mid Bx \leq 0\}$, where $B$ has rows $b_1, \ldots, b_s$. Then $P$ is a polyhedral cone by definition.
As $Ab_j \leq 0$ for all $j$, we have $a_i^T b_j \leq 0$ for all $i$ and $j$. By symmetry of the inner product we also have that $b_i^T a_i \leq 0$, so $Ba_i \leq 0$. Now let $x \in P = \text{cone}(\{a_1, \ldots, a_t\})$, then $x = \sum_{k=1}^t \lambda_k a_k$ for $\lambda_i \geq 0$. So

$$Bx = B \sum_{k=1}^t \lambda_k a_k = \sum_{k=1}^t \lambda_k Ba_k \leq 0,$$

so $P \subseteq \{ x \in \mathbb{R}^n \mid Bx \leq 0 \}$. Now suppose that there exists a vector $y \notin P$ such that $By \leq 0$. If $y \notin P$, then there does not exist a vector $\lambda \geq 0$ such that $A^T \lambda = y$. By Farkas’ Lemma 2.17 this means that there exists a vector $z$ such that $z^T y < 0$ and $z^T A = Az \geq 0$. So $-z \in D = \text{cone}(\{b_1, \ldots, b_s\})$. Then we have that $-z = B\gamma$ for some $\gamma \geq 0$. So $z^T y < 0$ thus $-z^T y = B\gamma y \leq 0$. Contradiction! So $\{ x \in \mathbb{R}^n \mid Bx \leq 0 \} \subseteq P$ and hence $P = \{ x \in \mathbb{R}^n \mid Bx \leq 0 \}$.

\[ \square \]

**Theorem 2.19.** A set $P$ is a polytope if and only if it is the convex hull of a finite set of points.

**Proof.** Let $P = \{ x \in \mathbb{R}^n \mid Ax \leq b \}$ be a nonempty polytope. Then

$$P = \left\{ x \mid \left( \begin{array}{c} x \\ 1 \end{array} \right) \in Q \right\}, \text{ where } Q := \left\{ \left( \begin{array}{c} x \\ \lambda \end{array} \right) \in \mathbb{R}^{n+1} \mid \lambda \geq 0, Ax - \lambda b \leq 0 \right\}.$$

Then $Q$ is a polyhedral cone, so by Theorem 2.18 it is generated by a finite set of nonzero vectors $\left( \begin{array}{c} x_1 \\ \lambda_1 \end{array} \right), \ldots, \left( \begin{array}{c} x_k \\ \lambda_k \end{array} \right)$. Since $P$ is bounded, all $\lambda_i$ are nonzero. Without loss of generality, assume that all $\lambda_i = 1$. It now follows that $x \in P$ if and only if

$$\left( \begin{array}{c} x \\ 1 \end{array} \right) = \mu_1 \left( \begin{array}{c} x_1 \\ 1 \end{array} \right) + \cdots + \mu_k \left( \begin{array}{c} x_k \\ 1 \end{array} \right)$$

for some nonnegative scalars. This implies that $P$ is the convex hull of $x_1, \ldots, x_k$.

Now suppose that $P$ is a convex hull of a finite set of points $x_1, \ldots, x_k \in \mathbb{R}^n$. Then $x \in P$ if and only if $\left( \begin{array}{c} x \\ 1 \end{array} \right) \in Q$, where $Q$ is the cone generated by $\left( \begin{array}{c} x_1 \\ 1 \end{array} \right), \ldots, \left( \begin{array}{c} x_k \\ 1 \end{array} \right)$. By Theorem 2.18 $Q$ is a polyhedral cone, so

$$Q = \{ \left( \begin{array}{c} x_1 \\ \lambda \end{array} \right) \mid Ax + \lambda b \leq 0 \}.$$

This implies that $P = \{ x \in \mathbb{R}^n \mid Ax + b \leq 0 \}$.

\[ \square \]

### 2.2. Gradient Descent Method

The gradient descent method is a method which minimizes a function by going in the negative direction of the gradient. In this section we will look at the assumptions needed to ensure that the method converges to a global minimum.
Definition 2.20 (Strongly convex function). Suppose $f$ is a twice continuously differentiable function. We call $f$ strongly convex on a set $X$ if there exists an $m > 0$ such that for all $x \in X$

$$\nabla^2 f(x) \succeq mI,$$  \hspace{1cm} (2.2)

where $\succeq$ means that $\nabla^2 f(x) - mI$ is positive semidefinite.

Strong convexity has some interesting consequences as can be seen in the theorem below.

Theorem 2.21. Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is a strongly convex function on $S$. Then for all $x, y \in X$

1. $f(y) \geq f(x) + \langle \nabla f(x), (y-x) \rangle + \frac{m}{2}\|y-x\|^2.$
2. $mI \preceq \nabla^2 f(x) \preceq MI$ for some constant $M$.
3. $f(y) \leq f(x) + \langle \nabla f(x), (y-x) \rangle + \frac{M}{2}\|y-x\|^2.$
4. $f(x) - \frac{1}{2m}\|\nabla f(x)\|^2 \leq p^* \leq f(x) - \frac{1}{2M}\|\nabla f(x)\|^2.$

Also the optimal point $x^*$ to the minimization problem (2.1) is unique.

Proof. See chapter 9 of [5]. \hfill \Box

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ which is strongly convex on $X$ and suppose that we want to solve the unconstrained optimization problem

$$\text{minimize } f(x).$$ \hspace{1cm} (2.3)

A direct consequence of convexity is that if a minimum exists, then it is unique. But there may not exist a stationary point in a normal convex function. This is why we demand the function to be strongly convex, there always exists a minimum and hence it is unique. So for strongly convex functions, a sufficient condition for a point $x^* \in X$ to be optimal is

$$\nabla f(x^*) = 0.$$ \hspace{1cm} (2.4)

So finding a solution to an unconstrained minimization problem (2.3) is the same as finding a solution to (2.4). These problems generally are solved using an iterative algorithm; an algorithm that computes points $x_0, x_1, \ldots$ such that $\lim_{m \to \infty} f(x_m) = x^*$.

We take a starting point $x_0 \in \text{dom } f$ and the gradient $\nabla f(x_0)$ of the function at that point. We will then go in the negative direction of the gradient to obtain a new point $x_1$ and so on. Because of convexity, this algorithm will converge to the point where the gradient is zero, and thus at a minimum. We can search for the minimum as follows

$$x_{m+1} = x_m - \eta \nabla f(x_m).$$

For certain choices of $\eta > 0$, it guarantees that $f(x_{m+1}) < f(x_m)$. But what step size do we need to choose? Too large and it will diverge and for too small choices it will take a
very long time to converge. To determine the step size we will use the backtracking line search method. The backtracking line search algorithm is as follows

<table>
<thead>
<tr>
<th>Data:</th>
<th>Descent direction $-\nabla f$ and $x \in \text{dom } f$, $a \in (0, 0.5)$, $\beta \in (0, 1)$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t := 1$.</td>
<td></td>
</tr>
<tr>
<td>while $f(x + t(-\nabla f)(x)) &gt; f(x) + at\nabla f(x)^T(-\nabla f(x))$ do</td>
<td></td>
</tr>
<tr>
<td>$t := \beta t$.</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

**Algorithm 1:** Backtracking Line Search Algorithm

The gradient descent algorithm is now defined as

<table>
<thead>
<tr>
<th>Data:</th>
<th>a starting point $x \in \text{dom } f$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>while $|\nabla f(x)|_2 &gt; \epsilon$ do</td>
<td></td>
</tr>
<tr>
<td>Choose a step size $\eta$ using the backtracking line search method.</td>
<td></td>
</tr>
<tr>
<td>Update $x := x - \eta\nabla f(x)$.</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
</tbody>
</table>

**Algorithm 2:** Gradient Descent

The next theorem shows when the gradient descent algorithm converges to a unique minimum.

**Theorem 2.22.** Let $f$ be a strongly convex function on $X$. For $\eta < \frac{1}{M}$ the gradient descent algorithm converges.

**Proof.** Define the function $\tilde{f}(\eta) := f(x - \eta\nabla f(x))$. We will only consider $\eta$ such that $x - \eta\nabla f(x) \in X$. From Theorem 2.21.3 with $y = x - \eta\nabla f(x)$, we obtain a quadratic upper bound on $\tilde{f}$:

$$\tilde{f}(\eta) \leq f(x) - \eta\|\nabla f(x)\|_2^2 + \frac{M\eta^2}{2}\|\nabla f(x)\|_2^2.$$  

Note that

$$\eta \leq \frac{1}{M} \implies \eta \cdot \frac{M\eta}{2} \leq \frac{1}{M} \cdot \frac{M\eta}{2} \implies -\eta + \frac{M\eta^2}{2} \leq -\frac{\eta}{2}.$$  

Using these two bounds and from the fact that $\alpha$ from the backtracking line search is less than 1/2, we have

$$\tilde{f}(\eta) \leq f(x) - \eta\|\nabla f(x)\|_2^2 + \frac{M\eta^2}{2}\|\nabla f(x)\|_2^2$$

$$\leq f(x) - \frac{\eta}{2}\|\nabla f(x)\|_2^2$$

$$< f(x) - \alpha\eta\|\nabla f(x)\|_2^2.$$  

It can be shown that the backtracking line search either stops with $\eta = 1$ or $\eta \geq \beta/M$. For $\eta = 1$ we have

$$f(x_{n+1}) \leq f(x_n) - \alpha\|\nabla f(x_n)\|_2^2.$$
and for $\eta \geq \beta/M$ we have
\[ f(x_{n+1}) \leq f(x_n) - \frac{\alpha \beta}{M} \|
abla f(x_n)\|_2^2, \]
which in turn, bounds
\[ f(x_{n+1}) \leq f(x_n) - \min\{\alpha, \frac{\alpha \beta}{M}\} \|
abla f(x_n)\|_2^2. \]
Subtract $p^*$ from both sides to get
\[ f(x_{n+1}) - p^* \leq f(x_n) - p^* - \min\{\alpha, \frac{\alpha \beta}{M}\} \|
abla f(x_n)\|_2^2, \]
and combine this with $\|
abla f(x)\|_2^2 \geq 2m(f(x) - p^*)$, which follows from Theorem 2.21.4, to obtain
\[ f(x_{n+1}) - p^* \leq f(x_n) - p^* - \min\{\alpha, \frac{\alpha \beta}{M}\} (2m(f(x) - p^*)) \]
\[ = (1 - \min\{2ma, \frac{2ma\beta}{M}\})(f(x_n) - p^*). \]
Applying this inequality recursively on $n$, we find that
\[ f(x_n) - p^* \leq c^n(f(x_0) - p^*), \]
where $c = 1 - \min\{2ma, \frac{2ma\beta}{M}\} < 1$. This shows that $f(x_n)$ converges to $p^*$ as $n \to \infty$. \hfill \Box

2.2.1. Projected Gradient Descent

Now consider the following constrained optimization problem
\[ \text{minimize } f(x) \text{ over } x \in C \quad (2.5) \]
where $f$ is convex and twice differentiable and where $C$ is a non-empty closed convex set. We again want to solve this by following the steepest descent direction $x_{n+1} = x_n - \eta \nabla f(x_n)$. The only problem is that the new $x_{n+1}$ need not belong to $C$. Now define the projection of $x \in \mathbb{R}^n$ onto $C$ as
\[ P_C(x) = \text{argmin}_{y \in C} \|x - y\|^2_2. \]
We can now define the projected gradient descent algorithm as follows

<table>
<thead>
<tr>
<th>Data:</th>
<th>a starting point $x \in C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>while $|\nabla f(x)|_2 &gt; \epsilon$ do</td>
</tr>
<tr>
<td>2</td>
<td>Choose a step size $\eta$ using the backtracking line search method.</td>
</tr>
<tr>
<td>3</td>
<td>Update $x := P_C(x - \eta \nabla f(x))$.</td>
</tr>
<tr>
<td>4</td>
<td>end</td>
</tr>
</tbody>
</table>

Algorithm 3: Projected Gradient Descent

We will not do the convergence analysis of algorithm 3 here, see for example [6].
2.3. Linear Programming

As already stated above a linear program (LP) is an optimization problem of the form

\[
\text{minimize } \mathbf{c}^T x \quad \text{subject to } \mathbf{A}x \leq \mathbf{b} \quad \text{over } \mathbf{x} \in \mathbb{R}^n.
\]

The standard form or canonical form of a LP is

\[
\text{maximize } \mathbf{d}^T x \quad \text{subject to } \mathbf{A}x \leq \mathbf{b} \quad \text{over } \mathbf{x} \in \mathbb{R}^n.
\]

Since \( \min\{\mathbf{c}^T x \mid \mathbf{A}x \leq \mathbf{b}\} = -\max\{-\mathbf{c}^T x \mid \mathbf{A}x \leq \mathbf{b}\} \) we can transform every minimization into a maximization problem and vice versa.

Both forms can be interpreted as minimizing/maximizing a linear function over a polyhedron \( P = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}x \leq \mathbf{b}\} \).

**Theorem 2.23.** If \( P = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}x \leq \mathbf{b}\} \) is a nonempty polytope, then the maximum of the linear function \( \mathbf{d}^T x \) subject to \( \mathbf{A}x \leq \mathbf{b} \) is attained by a vertex of \( P \).

**Proof.** The existence of the maximum and the fact that it is finite follows from the fact that a polytope is closed and bounded. Now suppose that the maximum \( m \) is not attained by a vertex \( v \) of \( P \), then \( m \) is a convex combination of vertices \( v_1, \ldots, v_k \) of \( P \):

\[
m = \sum_{i=1}^{k} \lambda_i v_i, \quad \text{where } \sum_{i=1}^{k} \lambda_i = 1.
\]

Hence

\[
d^T m = d^T \left( \sum_{i=1}^{k} \lambda_i v_i \right) = \sum_{i=1}^{k} \lambda_i d^T v_i \leq \sum_{i=1}^{k} \lambda_i \max_i (d^T v_i) = \max_i (d^T v_i),
\]

which contradicts the fact that \( m \) was maximal. \( \square \)

A small alteration of the above proof gives the same result for the minimum of a linear function on a polytope.

**Theorem 2.24.** Let \( P := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}x \leq \mathbf{b}\} \) be a nonempty polyhedron and let \( \mathbf{c} \) be a vector in \( \mathbb{R}^n \). If the supremum \( \delta := \sup\{\mathbf{c}^T x \mid x \in P\} \) is finite, then there exists a vector \( \mathbf{y} \in P \) such that \( \mathbf{c}^T \mathbf{y} = \delta \).

**Proof.** Let \( \{\mathbf{y}_n\} \) be a sequence in \( P \) such that \( \lim_{n \to \infty} \mathbf{c}^T \mathbf{y}_n = \delta \in \mathbb{R}^n \). By continuity of the inner product we have that \( \mathbf{c}^T (\lim_{n \to \infty} \mathbf{y}_n) = \delta \). Define \( \mathbf{y} := \lim_{n \to \infty} \mathbf{y}_n \) and since \( P \) is closed, \( \mathbf{y} \in P \). \( \square \)
2.4. Integer Programming

A lot of optimization problems can be described as minimizing a linear function $c^T x$ over the integer vectors in some polyhedron $P$, see the travelling salesman example 2.27 below. The feasible solution set to these problems are of the form $P \cap \mathbb{Z}^n$, where $P$ is a polyhedron.

**Definition 2.25.** Let $P$ be a polyhedron. Then the *integer hull* $P_I$ of $P$ is defined as the convex hull of all integer vectors in $P$, that is $P_I := \text{conv.hull}(\{P \cap \mathbb{Z}^n\})$.

![Figure 2.1.: A polyhedron $P$ and its integer hull $P_I$.](image)

Obviously $P_I \subseteq P$, since $P$ itself is convex.

**Theorem 2.26.** Let $P$ be a polyhedron and $P_I$ the integer hull of $P$. If $P$ is bounded then $P_I$ is a polytope.

**Proof.** If $P$ is bounded, then so is $P_I$ because $P_I \subseteq P$. Since $P_I$ consists of only integer vectors, the set $P_I$ is finite. So $P_I$ is convex hull of a finite set of points, and hence by Theorem 2.19 a polytope. \qed

Now the optimization problem takes the following form

$$\text{maximize } c^T x \quad \text{subject to } Ax \leq b \quad \text{over } x \in \mathbb{Z}^n.$$ 

This type of linear program is called a *integer linear program* (ILP).

**Example 2.27** (Travelling Salesman). Given a list of cities and a list of distances between each pair of cities, what is the shortest route such that we visit each city exactly once and then return at the origin city? This problem is called the travelling salesman problem and we can formulate this as an integer linear program. Define

$$x_{ij} = \begin{cases} 
1 & \text{the path from city } i \text{ to city } j \text{ is in the tour}, \\
0 & \text{otherwise}.
\end{cases}$$
Let $d_{ij}$ be the distance from $i$ to city $j$. Now consider the following integer program

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \\
\text{subject to} & \quad \sum_{i=1}^{n} x_{ij} = 1 \quad 1 \leq j \leq n \\
& \quad \sum_{j=1}^{n} x_{ji} = 1 \quad 1 \leq i \leq n \\
& \quad x_{ij} \in \{0, 1\} \quad \forall i, j.
\end{align*}
\]

Feasible solutions to this program may contain several subtours, see Figure 2.2. Since we want to start and end at the origin city and visit every city exactly once, those feasible solutions containing subtours are no good solutions for our problem. We need some additional constraints to exclude those. For this we also need extra variables $u_i$.

\[
\begin{align*}
& u_1 = 1, \\
& 2 \leq u_i \leq n, \\
& u_i - u_j + 1 \leq (n - 1)(1 - x_{ij}) \quad \forall i, j \neq 1.
\end{align*}
\]

This full formulation of the travelling salesman problem is called the Miller-Tucker-Zemlin (MTZ) formulation, see [10]. This definition excludes all subtours of less than $n$ vertices. To see this, suppose we have a subtour $(i_1, i_2, \ldots, i_k, i_1)$ with $k < n$, then the last inequality of 2.7 gives

\[
\begin{align*}
& u_{i_1} - u_{i_2} + 1 \leq 0 \\
& u_{i_2} - u_{i_3} + 1 \leq 0 \\
& \vdots \\
& u_{i_k} - u_{i_1} + 1 \leq 0.
\end{align*}
\]

Now when we sum over both sides, we find that $k \leq 0$, which is a contradiction. \qed

### 2.5. Mixed Integer Programming

In this section we will formulate the mixed integer linear programming problem and the mixed integer quadratic programming problem. The general form of a linear program is

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \in \mathbb{R}^n.
\end{align*}
\]

An integer linear program demands that the only feasible solutions are integer vectors, but a mixed integer linear program demands that only some of the coefficients are integers. So for some $\mathcal{I} \subset \{1, \ldots, n\}$ we can formulate a mixed integer linear program as

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x_i \in \mathbb{Z}, \quad \forall i \in \mathcal{I}, \\
& \quad x_i \in \mathbb{R}_+, \quad \forall i \notin \mathcal{I}.
\end{align*}
\]
The general form of a quadratic optimization program is

\[
\text{minimize } x^T Q x + c^T x \text{ subject to } Ax \leq b \text{ over } x \in \mathbb{R}^n,
\]

where \(c \in \mathbb{R}^n\), \(b \in \mathbb{R}^k\), \(A \in \mathbb{R}^{k \times n}\) and \(Q \in \mathbb{R}^{n \times n}\) positive semidefinite. A mixed integer quadratic program can now be formulated as

\[
\text{minimize } x^T Q x + c^T x \text{ subject to } Ax \leq b \text{ over } \begin{cases} x_i \in \mathbb{Z}, & \forall i \in I, \\ x_i \in \mathbb{R}^+, & \forall i \notin I. \end{cases}
\]

There is a lot of literature on linear and integer programming, see for example [2] or [3].
3. Linear Regression Analysis

Linear regression is fitting a model for a dependent, or response, variable \( Y \) to experimental data, or explanatory variables, denoted \( X \). In this model it is assumed that the response variable depends linearly on the explanatory variables.

**Definition 3.1** (Linear Regression Model). Given \( n \) observations of the response variable \( Y \) and given \( p \) independent explanatory variables \( x_1, \ldots, x_p \), then for \( i, j = 1, \ldots, n \),

\[
Y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + \epsilon_i, \quad \mathbb{E}\epsilon_i = 0, \quad \mathbb{E}\epsilon_i\epsilon_j = \begin{cases} \sigma^2, & i = k, \\ 0, & i \neq k. \end{cases} \tag{3.1}
\]

Here \( \beta_0, \ldots, \beta_p \) and \( \sigma^2 \) are unknown constants, called *regression coefficients*, \( \beta_0 \) is called the *intercept* and \( \epsilon_i \) is the unknown stochastic measurement error of the model.

We can also write this in matrix notation as

\[
Y = X\beta + \epsilon, \quad \mathbb{E}\epsilon = 0, \quad \text{cov}(\epsilon) = \sigma^2 I_n
\]

Here \( X \) is the \( n \times (p+1) \)-matrix with the \( i \)-th row \( x_i = (1, x_{i1}, \ldots, x_{ip}) \).

Our first goal is to estimate \( \beta \) and \( \sigma^2 \).

### 3.1. Estimation of Regression Coefficients

The most common way to estimate \( \beta \) is using the least squares method, which we can formulate in the following way

\[
\text{minimize} \quad \|Y - X\beta\|_2^2 \quad \text{over} \quad \beta \in \mathbb{R}^n. \tag{3.2}
\]

Writing out

\[
\|Y - X\beta\|_2^2 = (Y - X\beta)^T(Y - X\beta) = Y^T Y - Y^T X\beta - \beta^T X^T Y + \beta^T X^T X \beta.
\]

We see that both \( Y^T X \beta \) and \( \beta^T X^T Y \) have dimension \( 1 \times 1 \), so

\[
\beta^T X^T Y = (Y^T X \beta)^T = Y^T X \beta,
\]
which gives
\[ \|Y - X\beta\|^2 = Y^T Y - 2\beta^T X^T Y + \beta^T X^T X \beta. \]
Differentiating this with respect to \(\beta\) and setting it to zero gives \(-2X^T Y + 2X^T X \beta = 0\), so
\[ -X^T Y + X^T X \beta = 0, \]
and hence \(X^T X \beta = X^T Y\). We can now estimate \(\beta\) with
\[ \hat{\beta} = (X^T X)^{-1} X^T Y. \]  
(3.3)

Note that if \(X\) has full rank, then this estimation is unique. Also we have
\[ E\hat{\beta} = E[(X^T X)^{-1} X^T Y] = (X^T X)^{-1} X^T EY \]
\[ = (X^T X)^{-1} X^T E(X\beta + \epsilon) = (X^T X)^{-1} X^T X \beta + (X^T X)^{-1} X^T E\epsilon \]
\[ = \beta, \]
so \(\hat{\beta}\) is an unbiased estimator of \(\beta\). And
\[ \text{var}(\hat{\beta}) = \text{var}[(X^T X)^{-1} X^T (X\beta + \epsilon)] \]
\[ = \{(X^T X)^{-1} X^T\} \text{var}[X\beta + \epsilon]\{(X^T X)^{-1} X^T\}^T \]
\[ = \sigma^2(X^T X)^{-1} X^T X(X^T X)^{-T} \]
\[ = \sigma^2(X^T X)^{-T} = \sigma^2(X^T X)^{-1}. \]

This variance in fact can be very large, and is one of the downsides of least squares estimation. Also for the estimation of \(\hat{\beta}\) we need to invert the matrix \(X^T X\), and in the case that \(X\) is large and dense the algorithms are highly inefficient. A somewhat more efficient method is to estimate \(\beta\) with the gradient descent method, see section 2.2.

Example 3.2 (Lung cancer from smoking). Consider the data from a smoking-cancer study\(^1\), which is shown as a scatterplot in Figure 3.1.

We want to model the dataset with a model \(Y = \beta_0 + X_1 \beta_1\). We will minimize the following (convex) function, see Figure 3.2:

\[ J(\beta_0, \beta_1) = \sum_{i=1}^{n} (Y_i - (\beta_0 + X_i \beta_1))^2 = \|Y - X \beta\|^2. \]

The gradient of this function is as follows
\[ \nabla J(\beta_0, \beta_1) = \begin{pmatrix} \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_0} \\ \frac{\partial J(\beta_0, \beta_1)}{\partial \beta_1} \end{pmatrix} = \begin{pmatrix} 2 \sum_{i=1}^{n} (Y_i - (\beta_0 + X_i \beta_1)) \\ 2 \sum_{i=1}^{n} (-X_i(Y_i - (\beta_0 + X_i \beta_1)) \end{pmatrix} \]

Instead of inverting the matrix \(X^T X\) we will use the gradient descent algorithm. We take (arbitrary) \(\beta_0^{(0)} = 0\) and \(\beta_1^{(0)} = 1\). We then go in the negative direction of the gradient, so that after a few steps we find a \(\hat{\beta}\) such that \(\nabla J(\hat{\beta}) = 0\). After a few steps we obtain \(\beta_0^* = 6.4706\) and \(\beta_1^* = 0.5291\) as best candidates. The model is shown in Figure 3.1.

\(^1\)J.F. Fraumeni, “Cigarette Smoking and Cancers of the Urinary Tract: Geographic Variations in the United States,” Journal of the National Cancer Institute, 41, 1205-1211.
It might happen that $X$ is not full rank, which means that not all columns of $X$ are linearly independent. This happens if, for example, two of the explanatory variables are perfectly correlated ($X_2 = 3X_3$ for example). As noted in the derivation of $\hat{\beta}$ above, this means that $\hat{\beta}$ is not unique.

Once our $\hat{\beta}$ is computed we can define the vector of predicted responses $\hat{Y} = X\hat{\beta}$ and the vector of residuals $R$ as

$$R = Y - \hat{Y}.$$  

Also, the residual sum of squares $RSS$ is defined as

$$RSS = \|Y - X\hat{\beta}\|^2_2 = (Y - X\hat{\beta})^T(Y - X\hat{\beta}) = (Y - \hat{Y})^T(Y - \hat{Y}) = R^T R,$$

or equivalently

$$RSS = \sum_{i=1}^{n} R_i^2 = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2.$$  

The estimator

$$\hat{\sigma}^2 = \frac{RSS}{n - p - 1}.$$
is used to estimate $\sigma^2$. Under (3.1) we see that $\epsilon \sim N(0, \sigma^2)$, so that $\frac{RSS}{n-p-1}$ has chi-squared distribution with $(n-p-1)$ degrees of freedom and $E\hat{\sigma}^2 = \sigma^2$. Furthermore, from (3.1) we also see that $Y \sim N(X\beta, \sigma^2)$ and hence $\hat{\beta} \sim N(\beta, \sigma^2(X^TX)^{-1})$.

### 3.1.1. The Gauss-Markov Theorem

An estimator $\hat{\theta}$ of $\beta$ is called linear if it is a linear combination of observable data. For example $\hat{\beta} = (X^TX)^{-1}X^TY$ is a linear estimator, since $(X^TX)^{-1}X^T$ is a linear and consistst of observable data.

**Theorem 3.3** (Gauss-Markov). For the regression model 3.1, the ordinary least squares estimator $\hat{\beta}$ of $\beta$ is the Best Linear Unbiased Estimator (BLUE), meaning that if $\hat{\gamma}$ is another unbiased linear estimator of $\beta$, then $\text{var}(\hat{\gamma}) \geq \text{var}(\hat{\beta})$.

**Proof.** Since $\hat{\gamma} = BY$ is another unbiased estimator, it follows that

$$\beta = E(\hat{\gamma}) = E(BY) = B\text{E}(Y) = BX \beta.$$ 

This implies $BX = I$, where $I$ is the identity matrix. Now write $B = (X^TX)^{-1}X^T + C$ for some nonzero matrix $C$. Then from $BX = I$

$$I = BX = ((X^TX)^{-1}X^T + C)X = I + CX \implies CX = 0.$$
Now
\[
\text{var}(\hat{\gamma}) = \text{var}(BY) = B \text{var}(Y)B^T = \sigma^2 BB^T
\]
\[
= \sigma^2 \left[(X^T X)^{-1}X^T + C((X^T X)^{-1}X^T + C)^T\right]
\]
\[
= \sigma^2 \left[(X^T X)^{-1}X^T + C(X(X^T X)^{-1} + C)^T\right]
\]
\[
= \sigma^2 \left[(X^T X)^{-1}X^T + C(X(X^T X)^{-1} + CC^T)^T\right]
\]
\[
= \sigma^2 \left[(X^T X)^{-1} + CC^T\right] = \sigma^2 (X^T X)^{-1} + \sigma^2 CC^T
\]
\[
= \text{var}(\hat{\beta}) + \sigma^2 CC^T.
\]

Because \( C \) is a nonsingular matrix, \( CC^T \) is a positive semi-definite matrix, \( \sigma^2 CC^T \) is positive and hence \( \hat{\beta} \) has smallest variance among linear estimators.

The Gauss-Markov Theorem implies that the least squares estimator has the smallest mean squared error of all the unbiased linear estimators. It may well be possible that there exists an estimator with smaller mean squared error but with a little bit of bias. In this case we might decide to trade a little bit of bias for a reduction in variance. As we will see later with the shrinkage methods, the estimator we obtain using these methods is biased but have less variance than the least squares estimator.

3.2. Best Subset Selection

There are two main reasons why we are a lot of times not satisfied with the least squares estimate \( \hat{\beta} = (X^T X)^{-1}X^T Y \). Remember that \( \hat{\beta} \) is an unbiased estimator of \( \beta \) and that the variance is \( \sigma^2 (X^T X)^{-1} \). This variance can in fact be very large. We often can improve this variance by shrinking or setting some regression coefficients to zero. This way we obtain a little bit of bias, but we may improve the variance. The second reason is that if we have a large set of explanatory variables, a lot of them have very little effect on the model, and a lot of times we are willing to sacrifice those variables for having a smaller model.

For obtaining this 'best' smaller model, we can of course search through each possible subset, but this can quickly become unfeasible. There are more efficient methods and we will describe a few of them. But before we can, we need to define what makes one model 'better' than the other.

3.2.1. The Determination Coefficient

To determine the best fit of a linear regression model we will use the determination coefficient \( R^2 \) as a measure. This measure compares the residual sum of squares \( RSS \) of the given linear regression model to the of the empty model \( Y = \beta_0 1 + \epsilon \). We will denote the residual sum of squares of the empty model as \( SSY \). In this case we can estimate \( \beta = \beta_0 \) with \( \hat{\beta} = \bar{Y} \), such that the predicted responses can be expressed as
\[ \hat{Y} = X\hat{\beta} = \beta_0 1 = Y1. \] It follows that \[ \hat{Y}_i = Y \] for every \( i \), so
\[ SSY = \sum_{i=1}^{n} (Y_i - Y)^2. \]

If we look at the difference \( SSY - RSS \), we see the explanatory power of the part that is in the larger model, but not in the smaller model. Meaning that the size of the difference is a measure of the usefulness of our regression model. If we scale this difference in the following way, we obtain the coefficient of determination \( R^2 \)
\[ R^2 = \frac{SSY - RSS}{SSY} = 1 - \frac{RSS}{SSY}. \]

For linear regression models it holds that \( 0 \leq R^2 \leq 1 \), and the bigger \( R^2 \), the better the model.

### 3.2.2. \( t \)-test

Our goal is to obtain the best possible model with the smallest possible numbers of explanatory variables. For this we will test the hypothesis that a particular regression coefficient \( \beta_j = 0 \). For this we will for the Z-score
\[ z_j = \frac{\hat{\beta}}{\hat{\sigma}/\sqrt{v_j}}, \]
where \( v_j \) is the \( j \)-th diagonal element of \((X^TX)^{-1}\). Our null hypothesis will be \( \beta_j = 0 \), so that \( z_j \) has a \( t \) distribution with \( n - p - 1 \) degrees of freedom, e.g. \( z_j \sim t_{n-p-1} \). Now for significance level \( \alpha \), our null hypothesis is rejected if
\[ |z_j| \geq t_{(n-p-1);1-\alpha/2}, \]
where \( t_{(n-p-1);1-\alpha/2} \) is the \((1 - \alpha/2)\) quantile. Note that if \( \sigma \) is known and not estimated, then \( z_j \) has a standard distribution.

### 3.2.3. \( F \)-test

Instead of testing for the significance of every regression coefficient individually, we often need to test for the significance of groups of coefficients simultaneously. Let’s say that besides the explanatory variables \( X_1, \ldots, X_q \), we are interested if \( X_{q+1}, \ldots, X_r \) should be included in the model. For this we will consider the following test
\[
\begin{align*}
H_0 & : \beta_{q+1} = \ldots = \beta_r = 0, \\
H_1 & : \beta_j \neq 0 \text{ for some } j, (q + 1) \leq j \leq r.
\end{align*}
\]
To do this, fit the model with only \( X_1, \ldots, X_q \) and determine the residual sum of squares \( RSS_q \). Next fit the model with \( X_1, \ldots, X_q, X_{q+1}, \ldots, X_r \) and also determine \( RSS_r \). Now define
\[
F = \frac{(n - r - 1)(RSS_q - RSS_r)}{(r - q)RSS_r}.
\]
By our assumptions that $\epsilon_i$ is normally distributed, then the distribution of $F$ under the null hypothesis is an $F$-distribution with $(r - q)$ and $(n - r - 1)$ degrees of freedom. We will reject $H_0$ with significance $\alpha$ if

$$F \geq F_{(r-q),(n-r-1);1-\alpha}$$

Now that we can test for the significance of a regressor or a set of the regressors, we can start investigating some methods for selecting a 'best' subset from a set of variables. The first method we will investigate will be the forward-stepwise selection.

### 3.2.4. Forward-Stepwise Selection

Forward-stepwise selection starts with the model $Y = \beta_0 + \epsilon$ and stepwise adds the explanatory variable that most improves the fit. We can do this in the following way

1. Start with the empty model $Y = \beta_0 + \epsilon$.
2. Add the variable that yields the best increase in $R^2$.
3. Test the significance of the newly added variable using the $t$-test. If significant, go back to step 2, else stop.

**Example 3.4** (Predicting body fat percentage). Accurate measurement of body fat can be done by using expensive machines and can be quite inconvenient. It is therefore desirable to have an easy method of estimating the body fat percentage. Suppose we have the following data from 252 men:

\[
\begin{align*}
Y &= \text{body fat(\%)} \\
X_1 &= \text{age (years)} \\
X_2 &= \text{weight (kg)} \\
X_3 &= \text{height (cm)} \\
X_4 &= \text{neck circumference (cm)} \\
X_5 &= \text{chest circumference (cm)} \\
X_6 &= \text{abdomen circumference (cm)} \\
X_7 &= \text{hip circumference (cm)} \\
X_8 &= \text{thigh circumference (cm)} \\
X_9 &= \text{knee circumference (cm)} \\
X_{10} &= \text{ankle circumference (cm)} \\
X_{11} &= \text{upper arm circumference (cm)} \\
X_{12} &= \text{forearm circumference (cm)} \\
X_{13} &= \text{wrist circumference (cm)}
\end{align*}
\]

We want to estimate the body fat percentage with a linear model (3.1). Using the least squares estimator (3.3) on the standardized data (so the columns have zero mean and a standard deviation of one), we obtain the following model.

\[
Y = 0.0935X_1 - 0.3106X_2 - 0.0305X_3 - 0.1367X_4 - 0.0240X_5 + 1.2302X_6 \\
- 0.1777X_7 + 0.1481X_8 + 0.0044X_9 + 0.0352X_{10} + 0.0656X_{11} \\
+ 0.1091X_{12} - 0.1802X_{13}.
\]

This model has an $R^2$ value of 0.749 and mean squared error 0.2642. Measuring all the 13 variables is a rather tedious task and probably isn't more convenient than measuring the body fat percentage using a machine. And also some variables will not have any
correlation with the response variable, so we would like to remove those for the sake of interpretability. Using the forward-stepwise algorithm described above we obtain the following smaller model

\[ Y = -0.4763X_2 + 1.2830X_6 + 0.1142X_{12} - 0.1680X_{13}. \]

The \( R^2 \) value of this model is 0.735 and mean squared error of 0.2693. Stepwise regression left us with a model of only 4 variables, a reduction of 9. The \( R^2 \) value is only slightly less and our MSE increased just a little bit. We sacrificed a little accuracy for a big gain in interpretability.

For a model with \( n \) variables, searching through all the possible models and select the one with lowest expected prediction error quickly becomes unfeasible, as we need to check \( 2^n \) models. With the forward-stepwise selection method we 'only' need to check at most \( 1 + \sum_{i=1}^{n} i \) models. However, the forward-stepwise method is a greedy algorithm. A greedy algorithm selects at each step the local optimum hoping that it finds a global optimum, but this may not be the case. This is the reason why we search for other methods, like the Lasso.

### 3.2.5. Regression Shrinkage and Selection via the Lasso

Shrinkage methods shrink the regression coefficients by demanding their size is smaller than some value. In this section we will consider the Lasso method. The lasso estimate is defined as the solution to the optimization problem

\[
\text{minimize} \quad \| Y - X\beta \|_2^2 \quad \text{subject to} \quad \sum_{j=1}^{p} |\beta_j| \leq t \quad \text{over} \quad \beta \in \mathbb{R}^p,
\]

or in Lagrangian form

\[
\hat{\beta}^{\text{lasso}} = \arg\min_{\beta} \left\{ \frac{1}{2}(Y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.
\]

Here \( \lambda \) is called the tuning parameter. Computing the lasso estimate is a quadratic programming problem. For some efficient and stable algorithms see section 6 of [14].

Because of the nature of the constraint, decreasing \( t \) will force some of the coefficients to be exactly zero. So computing the lasso estimate is a kind of continuous subset selection. But why will some coefficients be exactly zero? It just imposes a penalty on the size of the least squares solution. For the case \( p = 2 \), Figure 3.3 shows why this happens. The red lines are the contours of the function \( \| Y - X\beta \|_2^2 \), the blue diamond is the constraint region. The lasso solution is the first place where the contours touch the square. Sometimes this will happen at a corner of the diamond, which means that the other coefficient will be exactly zero. If we would change our constraint to, for example, \( \sum_{j=1}^{p} \beta_j^2 \leq t \), the blue diamond would change to a blue circle without corners.
and so zero coefficients will rarely result. The method which uses $\sum_{j=1}^{p} \beta_j^2 \leq t$ instead of $\sum_{j=1}^{p} |\beta_j| \leq t$ is called ridge regression.

But how do we select the ’best’ $t$ or $\lambda$ for our model? We will show how to select the tuning parameter for best prediction accuracy using the 2−fold cross-validation method.

Define the mean-squared error of an estimate $\hat{X}\beta$ as

$$ME = \mathbb{E} \left[ (\hat{X}\beta - X\beta)^2 \right]$$

and the prediction error of $\hat{X}\beta$ as

$$PE = \mathbb{E} \left[ (Y - \hat{X}\beta)^2 \right] = ME + \sigma^2,$$

In 2−fold cross-validation, we randomly partition the original sample into 2 equal sized subsamples $S_1$ and $S_2$. Then choose a set of $\lambda$ values and for each of this $\lambda$, fit the model on $S_1$ and calculate it’s prediction error $PE_1$ on $S_2$. Then fit on $S_2$ and calculate the $PE_2$ on $S_1$. The prediction error $PE_\lambda$ of the model with tuning parameter $\lambda$ is the average of both $PE_1$ and $PE_2$. Then the model with lowest $PE_\lambda$ is the best for prediction accuracy.

**Example 3.5** (Predicting body fat percentage continued). Again we want to select the best subset, but now we will use the lasso method. Using 2−fold vross-validation, the model with the lowest MSE had $\lambda = 0.0216$ and yields the following model:

$$Y = 0.0970X_1 - 0.0696X_3 - 0.0723X_4 + 0.9100X_6 + 0.0506X_{12} - 0.1251X_{13}.$$  

This model has mean squared error of 0.2893, which is higher than the MSE of the forward-stepwise method. This is because the amount of observations $n$ exceeds the amount of variables $x_1, \ldots, x_p$ by a large amount.
4. Best Subset Problem via Mixed Integer Optimization

In the previous section we have defined the linear regression model and calculated the optimal solution of the regression coefficients. If we have a large number of explanatory variables, we often want a smaller subset which has the strongest effect on the model. Consider the case that from the $p$ explanatory variables, we want the best $k$. Then (3.2) becomes

$$\text{minimize} \; \|Y - X\beta\|_2 \; \text{subject to} \; \|\beta\|_0 \leq k \; \text{over} \; \beta,$$

where $\|\cdot\|_0$ counts the number of nonzero elements. Problem (4.1) is known as the Best Subset Problem and the cardinality constraint makes this problem NP-hard[11].

In this chapter we will use one of the algorithms described in the paper *Best Subset Selection via a Modern Optimization Lens* by Dimitris Bertsimas, see [1]. In his paper he gives a few reformulations of the best subset problem as mixed integer optimization problems. He has different formulations because one is more useful in the $n > p$ regime, and the other in the $p \gg n$. Because of the reformulation, it is possible to solve this problem with efficient MIO solvers such as Gurobi. He also proposes two discrete first order algorithms and a combination of the discrete first order algorithm and the MIO solver.

In the first section we have chosen one of his MIO formulations and we show how to find the bound for the constraint. In the second and the third section we have selected one of his algorithms and show that it converges to an optimal solution. In the last section we show how to apply the algorithm to our problem and we give a brief review of the rest of his paper and his claims.

4.1. Formulation

In this section we will reformulate problem (4.1) to a mixed integer optimization (MIO) problem, so that we can use techniques to solve MIO problems using a MIO solver such as Gurobi.

As already mentioned, the best subset problem is NP-hard. So a polynomial time algorithm is not yet known, which means that the best subset problem is computational very difficult. As we have seen in the previous chapters, there are some methods like Lasso which can solve this type of problem. But Lasso has some shortcomings, both Lasso and Ridge lead to biased estimates, since the constraint penalizes the large and small coefficients uniformly. Also when $p \gg n$, the Lasso selects at most $n$ variables. So the number of predictors is bounded by the number of observations.
Problem (4.1) can be reformulated as a mixed integer quadratic optimization problem as follows
\[
\min_{\beta, z} \frac{1}{2} \| Y - X\beta \|_2^2 \quad \text{subject to} \quad \begin{cases}
-MU z_i \leq \beta_i \leq MU z_i, & i = 1, \ldots, p, \\
z_i \in \{0, 1\}, & i = 1, \ldots, p, \\
\sum_{i=1}^p z_i \leq k.
\end{cases}
\] (4.2)

Here \( z \in \{0, 1\}^p \) is a binary variable and \( MU \) is a constant such that if \( \hat{\beta} \) is a minimizer of problem (4.2) then \( MU \geq \| \hat{\beta} \|_\infty \). Notice that \( \sum_{i=1}^p z_i \) is an indicator of the number of zeros in \( \beta \).

Before we can show which values for \( MU \) are appropriate, we need some notation. For \( j = 1, \ldots, p \), let \( X(j) \) represent the \( j \)-th column of \( X \). Given a set \( I \subset \{1, \ldots, p\} \) with \( \#I = k \), let \( \hat{\beta}_I \) denote the least squares solution of regressing \( Y \) on \( X_I \). To obtain \( \hat{\beta} \) we append \( \hat{\beta}_I \) with zeros in the remaining coefficients, so
\[
\hat{\beta} = \arg\min_{\beta: \hat{\beta}_I = 0, \beta \in I} \| Y - X\beta \|_2^2.
\]

Furthermore, \( X(j) \) is notation for the order the correlations \( |\langle X_j, Y \rangle| \):
\[
|\langle X(1), Y \rangle| \geq |\langle X(2), Y \rangle| \geq \ldots \geq |\langle X(p), Y \rangle|
\]

**Definition 4.1** (Restricted Eigenvalue Condition). A matrix \( X \) satisfies the restricted eigenvalue condition if there exists a \( \eta_k \) such that
\[
\lambda_{\min}(X^T I X) \geq \eta_k \quad \text{for every} \quad I \subset \{1, \ldots, p\} \quad \text{such that} \quad \#I \leq k,
\] (4.3)
where \( \lambda_{\min} \) denotes the smallest eigenvalue.

Since \( MU \geq \| \hat{\beta} \|_\infty \), the next theorem gives an appropriate value for \( MU \).

**Theorem 4.2.** For any \( k \geq 1 \), any optimal solution \( \hat{\beta} \) to problem (4.1) satisfies
\[
\| \hat{\beta} \|_\infty \leq \min \left\{ \frac{1}{\eta_k} \left( \sum_{j=1}^k |\langle X(j), Y \rangle|^2, \frac{1}{\sqrt{\eta_k}} \| Y \|_2 \right) \right\}
\]

Proof. Let \( A = (X_J^T X_I)^{-1} X_J^T \), then \( \hat{\beta}_I = AY \). Let \( a_i \) denote the rows of \( A \) for \( i = 1, \ldots, k \). Then by the Cauchy-Schwarz inequality
\[
\| \hat{\beta}_I \|_\infty = \max_{i=1, \ldots, k} |\langle a_i, Y \rangle| \leq \max_{i=1, \ldots, k} \| a_i \|_2 \| Y \|_2.
\] (4.4)

Since the operator norm of a matrix is always less or equal than the largest eigenvalue, we have for every \( i = 1, \ldots, k \):
\[
\| a_i \|_2 \leq \max_{\| u \|_2 = 1} \| Au \|_2 = \max_{\| u \|_2 = 1} \| (X_I^T X_I)^{-1} X_I^T u \|_2 \leq \lambda_{\max} (X_I^T X_I)^{-1} \leq \max \left\{ \frac{1}{d_1}, \ldots, \frac{1}{d_k} \right\},
\]

28
where \(d_1, \ldots, d_k\) are the singular values of the matrix \(X_I\). For the last equality, let \(X_I = UDV^T\) denote the singular value decomposition of \(X_I\) where \(D = \text{diag}(d_1, \ldots, d_k)\). Then

\[
(X_I^T X_I)^{-1} X_I^T = (VD^2V^T)^{-1} (UDV^T)^T = (VD^{-2}V^T)(VDU^T) = VD^{-1}U^T,
\]

so \(D^{-1}\) is the diagonal matrix with singular values of \((X_I^T X_I)^{-1} X_I^T\), and hence the singular values are \(\frac{1}{d_1}, \ldots, \frac{1}{d_k}\).

The eigenvalues of \(X_I^T X_I\) are \(d_i^2\) and from the restricted eigenvalue condition (4.3) we have that \(d_i^2 = \lambda_i \geq \eta_k\). We now obtain

\[
\max_{i=1, \ldots, k} \|a_i\|_2 \leq \max \left\{ \frac{1}{d_1}, \ldots, \frac{1}{d_k} \right\} \leq \frac{1}{\sqrt{\eta_k}}.
\]

Substituting this into equation (4.4) gives the bound

\[
\|\hat{\beta}_I\|_\infty \leq \frac{1}{\sqrt{\eta_k}} \|Y\|_2.
\] (4.5)

Let \(\hat{A} = (X_I^T X_I)^{-1}\), then \(\hat{a}_i\) denotes the \(i\)-th row of \(\hat{A}\). Using this notation we have

\[
\|\hat{\beta}_I\|_\infty = \max_{i=1, \ldots, k} |\langle \hat{a}_i, X_I^T Y \rangle| \leq \max_{i=1, \ldots, k} \|\hat{a}_i\|_2 \|X_I^T Y\|_2 \leq \max \left\{ (X_I^T X_I)^{-1} \right\} \|X^T Y\|_2
\]

\[
= \max_{i=1, \ldots, k} \frac{1}{d_i^2} \sqrt{\sum_{j \in I} |\langle X_j, Y \rangle|^2} \leq \frac{1}{\eta_k} \sqrt{\sum_{j=1}^k |\langle X(j), Y \rangle|^2}
\] (4.6)

Combining (4.5) and (4.6) yields the result. \(\Box\)

We have examined one of the five MIO reformulations. The formulation we choose is a mixed integer quadratic optimization (MIQO) problem and can be solved with a MIO solver. His other four formulations involve more bounds and the notion of Specially Ordered Sets of Type 1. These formulations have more constraints but also have better statistical performance. See section 2.2 and 2.3 of [1].

4.2. The Algorithm

In the paper two algorithms are proposed. One without and one with line search. We will examine the algorithm without line search and show that it finds a local optimal solution. The algorithm is a first order method based on the projected gradient descent method.

We consider the following optimization problem

\[
\text{minimize } g(\beta) \text{ subject to } \|\beta\|_0 \leq k \text{ over } \beta \in \mathbb{R}^p,
\] (4.7)
where $g(\beta) \leq 0$ is convex and has Lipschitz continuous gradient
\[
\| \nabla g(\beta) - \nabla g(\hat{\beta}) \| \leq \ell \| \beta - \hat{\beta} \|.
\]  

(4.8)

We first show that $g(\beta) = \| \beta - \gamma \|_2^2$ for a given $\gamma$ satisfies the above properties. Then we show that it admits a closed form solution.

**Theorem 4.3.** The function $g(\beta) = \| \beta - \gamma \|_2^2$ for a given $\gamma$ satisfies

1. $g(\beta) \geq 0$;
2. $g$ is convex;
3. $g$ has Lipschitz continuous gradient.

**Proof.** The first follows from the definition of the $\ell_2$-norm. For the second fact, take $\beta_1, \beta_2 \in \mathbb{R}^p$ and $t \in [0, 1]$, and define $h(\beta) = \| \beta - \eta \|_2$, then
\[
h(t \beta_1 + (1 - t) \beta_2) = \| t \beta_1 + (1 - t) \beta_2 - \gamma \|_2 \\
= \| t \beta_1 + (1 - t) \beta_2 - \gamma + (1 - t) \gamma - (1 - t) \gamma \|_2 \\
\leq t \| \beta_1 - \gamma \|_2 + (1 - t) \| \beta_2 - \gamma \|_2 \\
= th(\beta_1) + (1 - t) h(\beta_2).
\]

Now let $f(x) = x^2$. Since squaring is nondecreasing we have
\[
f(h(t \beta_1 + (1 - t) \beta_2)) \leq f(th(\beta_1) + (1 - t) h(\beta_2))
\]
and by convexity of $f$
\[
f(th(\beta_1) + (1 - t) h(\beta_2)) \leq tf(h(\beta_1)) + (1 - t) f(h(\beta_2)).
\]

Now observe that $g(\beta) = f(h(\beta))$. This proves the second fact.

Now for 3. the gradient of $g(\beta)$ equals
\[
\nabla g(\beta) = \begin{pmatrix} 2(\beta_1 - \gamma_1) \\ \vdots \\ 2(\beta_p - \gamma_p) \end{pmatrix} = 2 \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} - \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_p \end{pmatrix},
\]
which gives
\[
\| \nabla g(\beta) - \nabla g(\hat{\beta}) \| = 2 \left\| \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} - \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_p \end{pmatrix} - 2 \left( \begin{pmatrix} \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{pmatrix} - \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_p \end{pmatrix} \right) \right\| = 2 \| \beta - \hat{\beta} \|.
\]

So the gradient of $g$ is Lipschitz continuous with Lipschitz constant 2. \qed
Theorem 4.4. An optimal solution $H_k(\gamma)$ to the problem

$$\text{minimize } \|\beta - \gamma\|_2^2 \text{ subject to } \|\beta\|_0 \leq k \text{ over } \beta \in \mathbb{R}^p$$

(4.9)
can be computed as follows: for $\gamma \in \mathbb{R}^p$

1. order the elements of $\gamma$ from largest to smallest in absolute value;
2. retain the first $k$ elements and set the rest to zero.

So if $|\gamma(1)| \geq |\gamma(2)| \geq \ldots \geq |\gamma(p)|$ are the ordered absolute values of $\gamma$ then

$$(H_k(\gamma))_i = \begin{cases} 
\gamma_i, & i \in \{(1), \ldots, (k)\} \\
0, & \text{otherwise}
\end{cases}$$

Proof. Let $\beta$ be an optimal solution to problem (4.9) and let $S := \{i \mid \beta_i \neq 0\}$. We can now rewrite our objective function as

$$\|\beta - \gamma\|_2^2 = \sum_{i \in S} |\beta_i - \gamma_i|^2 + \sum_{i \notin S} |\beta_i - \gamma_i|^2 = \sum_{i \in S} |\beta_i - \gamma_i|^2 + \sum_{i \notin S} |\gamma_i|^2.$$ 

Now if we set $\beta_i = \gamma_i$ for $i \in S$, the objective function becomes $\sum_{i \notin S} |\gamma_i|^2$. This function is minimized when $S$ corresponds to the indices of the largest $k$ values of $|\gamma_i|$. \qed

The next theorem says that if we have a current solution $\beta$, we have an upper bound for $g(\eta)$ around $g(\beta)$. This is useful for our algorithm in the sense that using the bound we can generate a decreasing sequence with as limit the solution.

Theorem 4.5. For a convex function $g(\beta)$ with Lipschitz continuous gradient and for any $L \geq \ell$ we have

$$g(\eta) \leq Q(\eta, \beta) := g(\beta) + \frac{L}{2} \|\eta - \beta\|_2^2 + \langle \nabla g(\beta), \eta - \beta \rangle,$$

for all $\eta, \beta$. Equality holds when $\beta = \eta$.

Proof. By Taylor’s Theorem we have

$$g(\eta) = g(\beta) + \nabla g(\beta)^T(\eta - \beta) + \frac{1}{2}(\eta - \beta)^T \nabla^2 g(\zeta)(\eta - \beta)$$

(4.10)

for some $\zeta$ on the line segment between $\eta$ and $\beta$. Now note that Lipschitz continuity of the gradient means $LI \succeq \nabla^2 g(\zeta)$, so

$$(\eta - \beta)^T \left( LI - \nabla^2 g(\zeta) \right)(\eta - \beta) \geq 0,$$

which gives

$$(\eta - \beta)^T LI(\eta - \beta) \geq (\eta - \beta)^T \nabla^2 g(\zeta)(\eta - \beta).$$

But the LHS also equals $L\|\eta - \beta\|_2^2$, so (4.10) becomes

$$g(\eta) \leq g(\beta) + \langle \nabla g(\beta), \eta - \beta \rangle + \frac{L}{2} \|\eta - \beta\|_2^2.$$ \qed
We can combine Theorem 4.4 with Theorem 4.5 to obtain

\[
\arg\min_{\|\eta\|_0 \leq k} \left( g(\beta) + \frac{L}{2} \|\eta - \beta\|_2^2 + \langle \nabla g(\beta), \eta - \beta \rangle \right)
\]

\[=
\arg\min_{\|\eta\|_0 \leq k} \left( \frac{L}{2} \|\eta - \left( \beta - \frac{1}{L} \nabla g(\beta) \right)\|_2^2 - \frac{1}{2L} \|\nabla g(\beta)\|_2^2 + g(\beta) \right)
\]  (4.11)

\[=
\arg\min_{\|\eta\|_0 \leq k} \|\eta - \left( \beta - \frac{1}{L} \nabla g(\beta) \right)\|_2^2
\]

\[= H_k \left( \beta - \frac{1}{L} \nabla g(\beta) \right). \]  (4.12)

The equality (4.11) is just an elementary property of inner products

\[
\|\alpha x + \beta y\|_2^2 = \langle \alpha x + \beta y, \alpha x + \beta y \rangle = |\alpha|^2 \langle x, x \rangle + 2\alpha\beta \langle x, y \rangle + |\beta|^2 \langle y, y \rangle
\]

Now we can present the algorithm to find a local optimal solution to problem (4.7).

| **Data:** \( g(\beta), L, \epsilon \) |
| **Result:** A local optimal solution \( \beta^* \) |
| **1** Initialize with \( \beta_1 \in \mathbb{R}^p \) such that \( \|\beta_1\|_0 \leq k; \) |
| **2** For \( m \geq 1 \), apply (4.12) with \( \beta = \beta_m \) to obtain \( \beta_{m+1} \) as |
| \( \beta_{m+1} = H_k \left( \beta_m - \frac{1}{L} \nabla g(\beta_m) \right); \) |
| **3** Repeat until \( \|\beta_{m+1} - \beta_m\|_2 < \epsilon; \) |
| **4** Let \( \beta_m := (\beta_{m1}, \ldots, \beta_{mp}) \) be the current estimate and let |
| \( I = \text{Supp}(\beta_m) := \{i \mid \beta_{mi} \neq 0\} \). Then solve the continuous optimization problem |
| minimize \( g(\beta) \) subject to \( b_i = 0 \ \forall i \notin I \) over \( \beta \) |
| and let \( \beta^* \) be a minimizer; |

**Algorithm 4:** The algorithm to find a local optimal solution to (4.7).

The paper also proposes a variant of the above algorithm. This algorithm uses line searches and has better empirical performance, see section 3.1 of the paper. The next section deals with the convergence analysis of the algorithm.

### 4.3. Convergence Analysis

We will now do the convergence analysis for algorithm 4. First we will define a first order stationary point, which is a necessary condition for optimality. After that we show that the algorithm either converges to a first order stationary point or to a global optimal solution.
Definition 4.6 (First order stationary point). Given $L \leq \ell$. The vector $\eta \in \mathbb{R}^p$ is called a first order stationary point of problem (4.7) if

1. $\|\eta\|_0 \leq k,$
2. $\eta = H_k \left( \eta - \frac{1}{L} \nabla g(\eta) \right).$

Definition 4.7 ($\epsilon$-Approximate first order stationary point). Given $\epsilon > 0$ and $L \geq \ell$. The vector $\eta \in \mathbb{R}^n$ satisfies the $\epsilon$-approximate first order optimality condition of problem (4.7) if

1. $\|\eta\|_0 \leq k,$
2. $\|\eta - H_k \left( \eta - \frac{1}{L} \nabla g(\eta) \right)\|_2 < \epsilon.$

Definition 4.8 (Sparsity pattern). For $\beta_m$ define $1_m = (e_1, \ldots, e_p)$ with

$$e_j = \begin{cases} 1, & \text{if } \beta_m j \neq 0 \\ 0, & \text{if } \beta_m j = 0. \end{cases}$$

This vector counts the number of nonzero elements of $\beta_m$ and is called the sparsity pattern of the support of $\beta_m$.

Theorem 4.9. Let $g(\beta)$ be a function satisfying (4.7) and (4.8). Let $(\beta_m)_{m \in \mathbb{N}}$ be the sequence generated by the algorithm from the previous section. Then

1. For any $L \geq \ell$, the sequence $(g(\beta_m))_{m \in \mathbb{N}}$ satisfies

$$g(\beta_m) - g(\beta_{m+1}) \geq \frac{L - \ell}{2} \|\beta_{m+1} - \beta_m\|_2^2,$$

decreases and converges.

2. If $L > \ell$, then $\lim_{m \to \infty} \beta_{m+1} - \beta_m = 0$.

3. If $L > \ell$ and $\|\lim \inf_{m \to \infty} \beta_m\|_0 = k$, then the sequence $(1_m)_{m \in \mathbb{N}}$ converges after finitely many iterations and the sequence $(\beta_m)_{m \in \mathbb{N}}$ is bounded and converges to a first order stationary point.

4. If $L > \ell$ and $\|\lim \inf_{m \to \infty} \beta_m\|_0 < k$, then $\lim_{m \to \infty} g(\beta_m) = g(\beta^*)$ where $\beta^* \in \arg\min g(\beta)$ is an unconstrained minimizer.

Proof. We will only prove 1. and 2. For the proof of 3. and 4. see [1], section 3.2.

1. Let $\beta$ be a vector such that $\|\beta\|_0 \leq k$. Using the bound from Theorem 4.5 we obtain

$$g(\beta) = Q(\beta, \beta) \geq \inf_{\|\eta\|_0 \leq k} Q(\eta, \beta) = \inf_{\|\eta\|_0 \leq k} \left( g(\beta) + \frac{L}{2} \|\eta - \beta\|_2^2 + \langle \nabla g(\beta), \eta - \beta \rangle \right)$$

$\overset{(4.1)}{=} \inf_{\|\eta\|_0 \leq k} \left( \frac{L}{2} \left\| \eta - \left( \beta - \frac{1}{L} \nabla g(\beta) \right) \right\|_2^2 - \frac{1}{2L} \|\nabla g(\beta)\|_2^2 + g(\beta) \right).$
Now let $\hat{\eta} = H_k(\beta - \frac{1}{L} \nabla g(\beta))$, then by (4.12) we have

$$\inf_{\|\eta\| \leq k} \left( \frac{L}{2} \left\| \eta - \left( \beta - \frac{1}{L} \nabla g(\beta) \right) \right\|_2^2 - \frac{1}{2L} \left\| \nabla g(\beta) \right\|_2^2 + g(\beta) \right)$$

$$= \frac{L}{2} \left\| \hat{\eta} - \left( \beta - \frac{1}{L} \nabla g(\beta) \right) \right\|_2^2 - \frac{1}{2L} \left\| \nabla g(\beta) \right\|_2^2 + g(\beta) \quad \text{ (by equation (4.12))}$$

$$= \frac{L}{2} \left\| \hat{\eta} - \beta \right\|_2^2 + \langle \nabla g(\beta), \hat{\eta} - \beta \rangle + g(\beta)$$

$$= \frac{L - \ell}{2} \left\| \hat{\eta} - \beta \right\|_2^2 + \left( \frac{\ell}{2} \left\| \hat{\eta} - \beta \right\|_2^2 + \langle \nabla g(\beta), \hat{\eta} - \beta \rangle + g(\beta) \right)_{Q(\hat{\eta},\beta)}$$

$$\geq \frac{L - \ell}{2} \left\| \hat{\eta} - \beta \right\|_2^2 + g(\hat{\eta}). \quad \text{ (by Theorem 4.5)}$$

We now have that

$$g(\beta) \geq \frac{L - \ell}{2} \left\| \hat{\eta} - \beta \right\|_2^2 + g(\hat{\eta}),$$

which leads to

$$g(\beta) - g(\hat{\eta}) \geq \frac{L - \ell}{2} \left\| \hat{\eta} - \beta \right\|_2^2.$$

If we set $\beta = \beta_m$, the algorithm gives $\beta_{m+1} = H_k \left( \beta_m - \frac{1}{L} \nabla g(\beta_m) \right) = \hat{\eta}$ and we obtain

$$g(\beta_m) - g(\beta_{m+1}) \geq \frac{L - \ell}{2} \left\| \beta_{m+1} - \beta_m \right\|_2^2.$$

Since all factors are positive, $g(\beta_m) \geq \frac{L - \ell}{2} \left\| \beta_{m+1} - \beta_m \right\|_2^2 + g(\beta_{m+1})$ implies $g(\beta_m) \geq g(\beta_{m+1})$, which in turn implies that the sequence $(g(\beta_m))_{m \in \mathbb{N}}$ is decreasing and since it is also bounded below by zero, by the Monotone Convergence Theorem it converges as $m \to \infty$.

2. Using 1. we know that the sequence $(g(\beta))_{m \in \mathbb{N}}$ converges, meaning that $g(\beta_m) - g(\beta_{m+1}) \to 0$ for $m \to \infty$. This implies that also $\frac{L - \ell}{2} \left\| \beta_{m+1} - \beta_m \right\|_2^2 \to 0$ for $m \to \infty$ and since $L > \ell$ we have that $\left\| \beta_{m+1} - \beta_m \right\|_2 \to 0$ and hence $(\beta_m)_{m \in \mathbb{N}}$ converges. \(\square\)

The above theorem says that the algorithm converges to a first order stationary point or that it converges to a globally optimal solution. The next theorem and corollary quantify the rate of convergence, the speed at which a convergent sequence approaches its limit.

**Theorem 4.10.** Let $\hat{\beta}$ be a first order stationary point of the algorithm and let $L > \ell$. After $M$ iterations the algorithm satisfies

$$\min_{m \in 1, \ldots, M} \left\| \beta_{m+1} - \beta_m \right\|_2^2 \leq \frac{2(g(\beta_1) - g(\beta))}{M(L - \ell)},$$

where $g(\beta_m)$ decreases to $g(\hat{\beta})$ as $m \to \infty$. 

34
Proof. Using the inequality from Theorem 4.9.1, we obtain by summing for $1 \leq m \leq M$
\[
\sum_{m=1}^{M} (g(\beta_m) - g(\beta_{m+1})) \geq \sum_{m=1}^{M} \frac{L - \ell}{2} \|\beta_{m+1} - \beta_m\|^2_2 = \frac{L - \ell}{2} \sum_{m=1}^{M} \|\beta_{m+1} - \beta_m\|^2_2.
\]

The LHS of the above equation yields
\[
\sum_{m=1}^{M} (g(\beta_m) - g(\beta_{m+1})) = g(\beta_1) - g(\beta_{M+1}).
\]

So
\[
g(\beta_1) - g(\beta_{M+1}) \geq \frac{L - \ell}{2} \sum_{m=1}^{M} \|\beta_{m+1} - \beta_m\|^2_2 \geq \frac{M(L - \ell)}{2} \min_{m=1, \ldots, M} \|\beta_{m+1} - \beta_m\|^2_2.
\]

By Theorem 4.9.1, the sequence $(g(\beta_m))_{m \in \mathbb{N}}$ is decreasing and converges and by part 3 of that theorem it converges to a first order stationary point $g(\hat{\beta})$. We have
\[
\sum_{m=1}^{\infty} (g(\beta_m) - g(\beta_{m+1})) \geq \sum_{m=1}^{M} (g(\beta_m) - g(\beta_{m+1}))
\]

which equals
\[
g(\beta_1) - g(\hat{\beta}) \geq g(\beta_1) - g(\beta_{M+1}).
\]

Now using (4.13) we obtain
\[
\frac{g(\beta_1) - g(\hat{\beta})}{M} \geq \frac{g(\beta_1) - g(\beta_{M+1})}{M} \geq \frac{(L - \ell)}{2} \min_{m=1, \ldots, M} \|\beta_{m+1} - \beta_m\|^2_2.
\]

Rewriting this gives
\[
\min_{m=1, \ldots, M} \|\beta_{m+1} - \beta_m\|^2_2 \leq \frac{2(g(\beta_1) - g(\hat{\beta}))}{M(L - \ell)}.
\]

**Corollary 4.11.** For any $\epsilon > 0$, there exists an $M$ such that for some $m^*$ with $1 \leq m^* \leq M$
\[
\|\beta_{m^*+1} - \beta_{m^*}\|^2_2 \leq \epsilon.
\]

**Proof.** Let $\epsilon > 0$. Since $L, \ell, g(\beta_1)$ and $g(\hat{\beta})$ are fixed, there exists an $M$ such that
\[
\frac{2(g(\beta_1) - g(\hat{\beta}))}{M(L - \ell)} \leq \epsilon.
\]

Now let $1 \leq m^* \leq M$ be the argument that minimizes $\|\beta_m - \beta_{m+1}\|^2_2$, then
\[
\|\beta_{m^*+1} - \beta_{m^*}\|^2_2 = \min_{m=1, \ldots, M} \|\beta_{m+1} - \beta_m\|^2_2 \leq \frac{2(g(\beta_1) - g(\hat{\beta}))}{M(L - \ell)} \leq \epsilon.
\]
4.4. Application

As shown in Theorem 4.9.1, the algorithm generates a decreasing sequence \( g(\beta_m)_{m \in \mathbb{N}} \) as a result of the upper bound of Theorem 4.5. It keeps lowering the upper bound until \( \|\beta_{m+1} - \beta_m\|_2 \leq \epsilon \) for some given \( \epsilon > 0 \). Then it solves the continuous optimization problem by keeping the zeros of \( \beta_m \) fixed.

Now if we take

\[
g(\beta) = \frac{1}{2}\|Y - X\beta\|_2^2,
\]

then

\[
\begin{align*}
\nabla g(\beta) &= -X^T(Y - X\beta), \\
\ell &= \lambda_{\max}(X^TX).
\end{align*}
\]

We have shown in Theorem 4.3 that \( g(\beta) \) satisfies the assumptions needed for the convergence of the algorithm. So we can use the algorithm to solve least squares problem.

**Example 4.12** (Predicting body fat percentage continued 2). An implementation of the above algorithm in Matlab yields the following model

\[
Y = 0.1347X_1 - 0.0783X_3 + 0.9151X_6 + 0.0935X_{12} - 0.2789X_{13}.
\]

Here \( \ell = 201.8 \), so we choose \( L = \ell + 1 \) and \( \epsilon = 0.0001 \). This model has an MSE of 0.2712, an improvement of the error lasso produces, which was 0.2893.

As already discussed, the paper of Bertsimas has five different formulations, two first order algorithms and three algorithms for solving the best subset problem. We have treated one formulation and one first order algorithm.

The paper compares three algorithms:

1. A variant of algorithm 4 using line searches,

2. MIO using the Gurobi solver with a cold start (random initialization) and time limit of 500 seconds,

3. MIO using the Gurobi solver with a warm start, so initializing with the solution found with the first algorithm, and time limit of 500 seconds.

For both the \( n > p \) and \( p \gg n \) regime the MIO with warm start performs the best, Lasso second and Stepwise regression has the worst performance. The paper claims that the MIO warm start approach solves problems with \( n \) in the thousands and \( p \) in the hundreds in minutes to optimality and for \( n \) in the hundreds and \( p \) in the thousands in minutes to near optimal solutions.
5. Conclusion

The main goal of this thesis was to review the article written by Dimitris Bertsimas. In the paper he proposed algorithms to solve the best subset problem more efficient. He did this using mixed integer optimization (MIO). This was the subject of the first chapter. There exists for both convex and discrete optimization a vast amount of literature, see [5] and [8]. In the second chapter we looked at linear regression and how to estimate the regression coefficient using the least squares method. We then motivated why we a lot of times only want to have a subset of the variables and we examined three popular different methods which are used to select those subsets. There also is a lot of literature on regression and on the methods used to solve regression problems, see for example [7] or [15].

In the last chapter we finally reviewed the paper by Bertsimas. We selected one of the MIO formulations and one of the algorithms he proposed and examined them extensively. The algorithms proposed in the paper had better computational and statistical performance compared to the Lasso and stepwise regression methods.
Bibliography


A. Popular Summary

One of the things on a lot of people’s minds is losing weight and getting fit. You can check if you have lost weight by standing on a scale and hopefully you will see a lower weight than before. But what is an healthy weight for one isn’t necessarily an healthy weight for someone else; bigger people tend to weigh more for example. Also, when you start working out you will grow muscle tissue and hence get heavier, so weight alone is not a good indicator for measuring an healthy bodyweight. The body fat percentage measures how much of your body consists of fat, so by reducing your body fat percentage you are losing fat, which for a lot of people is the main objective when ‘losing weight’.

Accurate measurement of the body fat percentage can be done by using expensive machines and those machines can also be quite inconvenient. So we want to a method which can predict the body fat percentage by measuring the circumference of different bodyparts. The formula which links the body fat percentage to the different circumferences of bodyparts is called a model. But the only way to make such a model is to estimate it using gathered data. So let’s say that of a group of 252 men we measure all of the following:

\[
Y = \text{body fat}(\%) \quad X_7 = \text{hip circumference (cm)}
\]
\[
X_1 = \text{age (years)} \quad X_8 = \text{thigh circumference (cm)}
\]
\[
X_2 = \text{weight (kg)} \quad X_9 = \text{knee circumference (cm)}
\]
\[
X_3 = \text{height (cm)} \quad X_{10} = \text{ankle circumference (cm)}
\]
\[
X_4 = \text{neck circumference (cm)} \quad X_{11} = \text{upper arm circumference (cm)}
\]
\[
X_5 = \text{chest circumference (cm)} \quad X_{12} = \text{forearm circumference (cm)}
\]
\[
X_6 = \text{abdomen circumference (cm)} \quad X_{13} = \text{wrist circumference (cm)}
\]

From this data we want to build a model which predicts the body fat percentage. This can easily be done with the least squares method, which gives the following formula, or model:

\[
Y = 0.0621X_1 - 0.1950X_2 - 0.0274X_3 - 0.4706X_4 - 0.0239X_5 + 0.9548X_6
\]
\[
- 0.2075X_7 + 0.2361X_8 + 0.0153X_9 + 0.1740X_{10} + 0.1816X_{11}
\]
\[
+ 0.4520X_{12} - 1.6206X_{13} - 18.1885.
\]

But who would want to measure every of the bodyparts and put it into a large formula above? Also, maybe some of the bodyparts have very little or even zero influence on the bodyfat percentage. We would like to make the above formula smaller by removing some of the bodyparts from the list above. So from \(X_1\) to \(X_{13}\) we would like to select only those which have the greatest influence on the bodyfat percentage. Of course the easiest way to do this is to try all the 8192 possible combinations of the above bodyparts, but
even for a small dataset, like the one above, this can be computational very difficult. It turns out that choosing a best subset of bodyparts is computation a very hard problem to solve. So we are looking for efficient methods to solve this problem. One of the methods described is the forward-stepwise method. This in general works as follows:

1. Start with the empty model.
2. For every of the thirteen bodyparts, select the one which improves the model the best and add it to the empty model.
3. Now from the remaining twelve, again add the bodypart which improves the model the best.
4. Do this until none of the remaining bodyparts improve the model any further.

This method gives the following model

\[ Y = -0.2990X_2 + 0.9958X_6 + 0.4729X_{12} - 1.5056X_{13} - 34.8541. \]

A great improvement over the big formula!

So now we have a method which succesfully selects a best subset, right? Not quite, the forward-stepwise method adds the variable which improves the previous model the best, but this doesn’t mean that the final model is the best model overall. The method described above is called a greedy algorithm. So the forward-stepwise method is not the answer to our best subset problem.

Another very popular method is called the Lasso method, which in short starts with the full model and then shrinks the coefficients to zero, with the added benefit that some coefficients will become exactly zero and so removes the bodypart from the formula. The big problem here is how to know when to stop. Also the Lasso has some other shortcomings too, so we still can improve the methods for finding the best subset.

In an recently published article, a new method for solving the best subset problem is proposed and in this thesis, this article is examined and reviewed. The method proposed in the article has computational and statistical improvements over the other methods, but it still isn’t the final solution to the best subset problem.