Quadratic Programming:

Theory and Algorithms

Sirong Luo

Faculty of Statistics and Data Science
Shanghai University of Finance and Economics

Quadratic Programming

A quadratic program is an optimization problem whose objective is to minimize or maximize a quadratic function subject to a finite set of linear equality and inequality constraints. By flipping signs if necessary, a quadratic program can be written in the generic form:

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{D}\boldsymbol{x} \geq \boldsymbol{d} \end{aligned} \tag{5.1}$$

for some vectors and matrices $\boldsymbol{c} \in \mathbb{R}^n, \boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{d} \in \mathbb{R}^p, \boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{D} \in \mathbb{R}^{p\times n}$, $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$. $\boldsymbol{Q}$ is a symmetric matrix. The term quadratic programming model is also used to refer to a quadratic program.

Observe that the constraint set in (5.1) is convex since it is a system of linear inequalities. Furthermore, the objective function of (5.1) is convex when $\boldsymbol{Q}$ is a positive semidefinite matrix. Throughout this chapter we assume that $\boldsymbol{Q}$ is symmetric and positive semidefinite. Therefore problem (5.1) is a convex program. A quadratic programming model is in **standard form** if it is written as follows:

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0}. \end{aligned} \tag{5.2}$$

Example 5.1 (Asset allocation)

Assume the one-year returns of the asset classes large stocks, small stocks, and bonds have the following correlations and standard deviations:

	Large	Small	Bonds	Standard deviation
Large	1	0.6	0.2	0.12
Small	0.6	1	0.5	0.20
Bonds	0.2	0.5	1	0.05

Determine the asset allocation of minimum risk, that is, find a portfolio comprised of these three asset classes whose return has the lowest standard deviation. Assume the portfolio can only hold long positions in each of the asset classes.

Example 5.1 (Asset allocation)

This problem can be formulated as a quadratic programming model. To that end, first construct the covariance matrix $\boldsymbol{V}$ of asset returns: this is the matrix whose (i, j) entry is the covariance of asset i and asset j; that is, $\rho_{ij} \cdot \sigma_i \cdot \sigma_j$. Using matrix notation and ' $\circ$ ' to denote the componentwise product of matrices, the covariance matrix can be computed as

$$
\begin{aligned}
\boldsymbol{V} &= \begin{bmatrix} 1 & 0.6 & 0.2 \\ 0.6 & 1 & 0.5 \\ 0.2 & 0.5 & 1 \end{bmatrix} \circ \begin{bmatrix} 0.12 \\ 0.20 \\ 0.05 \end{bmatrix} \begin{bmatrix} 0.12 & 0.20 & 0.05 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0.6 & 0.2 \\ 0.6 & 1 & 0.5 \\ 0.2 & 0.5 & 1 \end{bmatrix} \circ \begin{bmatrix} 0.0144 & 0.024 & 0.006 \\ 0.024 & 0.04 & 0.01 \\ 0.006 & 0.01 & 0.0025 \end{bmatrix} \\
&= \begin{bmatrix} 0.0144 & 0.0144 & 0.0012 \\ 0.0144 & 0.04 & 0.005 \\ 0.0012 & 0.005 & 0.0025 \end{bmatrix}
\end{aligned}
$$

Example 5.1 (Asset allocation)

Quadratic programming model for asset allocation Variables: x_i : percentage of the portfolio invested in asset i for $i = 1, 2, 3$. Objective (minimize the variance of the portfolio return):

$$\min_{\boldsymbol{x}} \boldsymbol{x}^\top \boldsymbol{V}\boldsymbol{x} = \min_{x_1,x_2,x_3} \big(0.0144x_1^2 + 0.04x_2^2 + 0.0025x_3^2 + 0.0288x_1x_2 + 0.0024x_1x_3 + 0.01x_2x_3\big)$$

Constraints:

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \quad \text{(percentages add up to one)} \\ x_1, x_2, x_3 &\geq 0 \quad \text{(long-only positions).} \end{aligned}$$

Observe that even in this small example the quadratic objective is much more concise and easier to write using matrix notation.

Consider a quadratic program without constraints:

$$\min_{\boldsymbol{x}} \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \tag{5.3}$$

The optimality conditions in this case are as follows.

Theorem (5.2)

Let $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$ and assume $\boldsymbol{Q}$ is symmetric and positive semidefinite. If (5.3) is bounded, then it attains its minimum. Furthermore, a point $\boldsymbol{x} \in \mathbb{R}^n$ is an optimal solution to (5.3) if and only if

$$\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c} = \boldsymbol{0}.$$

When $\boldsymbol{Q}$ is positive definite, the problem (5.3) has the unique minimizer $\boldsymbol{x} = -\boldsymbol{Q}^{-1}\boldsymbol{c}$. When $\boldsymbol{Q}$ is positive semidefinite but not positive definite, the matrix $\boldsymbol{Q}$ is singular and the problem (5.3) is either unbounded or has multiple solutions.

Example 5.3 (Ordinary least squares)

Assume $(\boldsymbol{x}_i, y_i)$, for $i = 1, \ldots, N$, is a random sample drawn from the joint distribution of X, Y where X, Y are respectively $\mathbb{R}^p$-valued and $\mathbb{R}$-valued random variables. Using the training data $(\boldsymbol{x}_i, y_i)$, with $i = 1, \ldots, N$, estimate a vector of coefficients $\boldsymbol{\beta}$ for the linear model

$$Y = \boldsymbol{\beta}^\top X + \epsilon.$$

The most popular approach to this problem is to find the estimate of $\boldsymbol{\beta}$ that solves the following least-squares problem:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{N} \left(\boldsymbol{\beta}^\top \boldsymbol{x}_i - y_i\right)^2$$

Observe that

$$\sum_{i=1}^{N} \left(\boldsymbol{\beta}^\top \boldsymbol{x}_i - y_i\right)^2 = (\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y})^\top (\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}) = \boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} - 2\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{y}^\top \boldsymbol{y}$$

Example 5.3 (Ordinary least squares)

for

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_N^\top \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}.$$

Hence the least-squares problem can be formulated as follows.
Quadratic programming formulation for least-squares estimation. Variables: $\boldsymbol{\beta}$: vector of coefficients in the linear model $Y = \boldsymbol{\beta}^\top X + \epsilon$. Objective:

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \boldsymbol{\beta}^\top \boldsymbol{Q} \boldsymbol{\beta} - \boldsymbol{b}^\top \boldsymbol{\beta}$$

where $\boldsymbol{Q} := \boldsymbol{X}^\top \boldsymbol{X}, \boldsymbol{b} = \boldsymbol{X}^\top \boldsymbol{y}$.
Constraints: None.

Example 5.3 (Ordinary least squares)

By applying Theorem 5.2 we obtain the widely known solution to the least-squares problem:

$$\hat{\boldsymbol{\beta}} := \boldsymbol{Q}^{-1}\boldsymbol{b} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

provided the $N \times p$ matrix $\boldsymbol{X}$ has full column rank. This latter condition usually holds in the typical practical situation when there are more observations than predictor variables; that is, when $N > p$. However, the case $N < p$ occurs as well. In this kind of situation the matrix $\boldsymbol{X}$ is never full column rank so the ordinary least-squares approach is not appropriate. Section 5.6.2 describes two popular variants for this kind of situation, namely ridge regression and lasso regression, both of which can be seen as modifications of the ordinary least-squares procedure.

Numerical Quadratic Programming Solvers

Figure 1 displays a printout of an Excel spreadsheet implementation of the quadratic programming model for Example 5.1 as well as the dialog box obtained when we run the Excel add-in Solver. The spreadsheet model contains the three components of the quadratic program. The decision variables are in the range B20:D20. The objective function is in cell E22. The Excel formula in this cell, using matrix operations, is as follows: MMULT(B20 : D20, MMULT(B16 : D18, TRANSPOSE(B20 : D20))).
The left-hand and right-hand sides of the equality constraint are in cells E20 and G20 respectively.

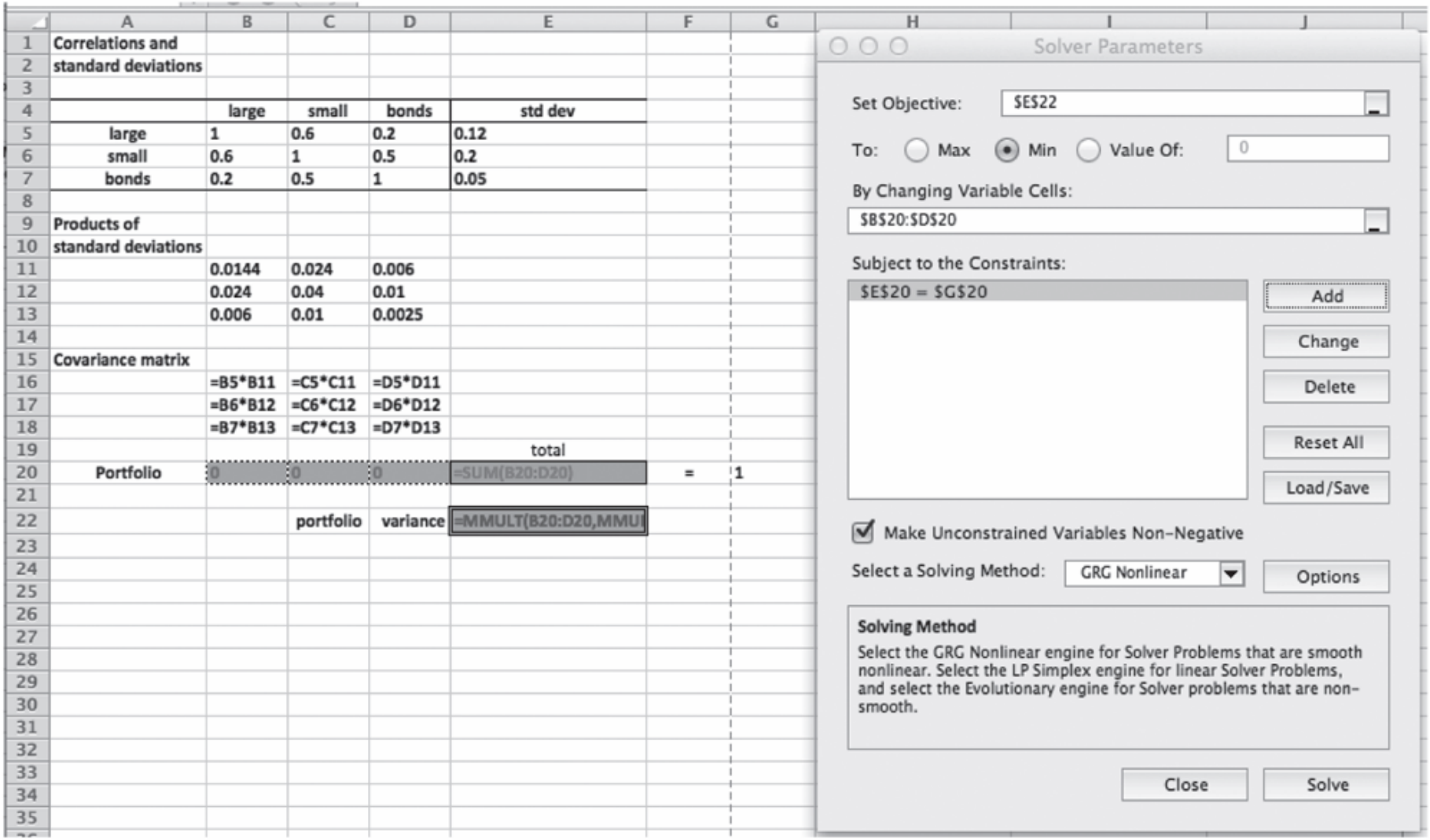

Figure 1: Spreadsheet implementation and the Solver dialog box for the asset allocation model

Sensitivity Analysis

As is the case for linear programming, the process of solving a quadratic program also generates some interesting sensitivity information via the so-called Lagrange multipliers associated with the constraints. Assume the constraints of a quadratic program, and hence the Lagrange multipliers, are indexed by $i = 1, \ldots, m$. The Lagrange multiplier y_i^* of the i th constraint has the following sensitivity interpretation:
If the right-hand side of the i th constraint changes by Δ, then the optimal value of the quadratic program changes by approximately $\Delta \cdot y_i^*$ for small Δ.

Unlike the shadow prices of a linear program, the Lagrange multipliers only give an approximation of the change in the optimal objective value. The situation is akin to how the derivative of a quadratic (or more general nonlinear) function at a particular point gives an approximation of the change in the function value when that point changes. To make this information explicit in Excel Solver, we request a sensitivity report after running Solver as shown in Figure 2.

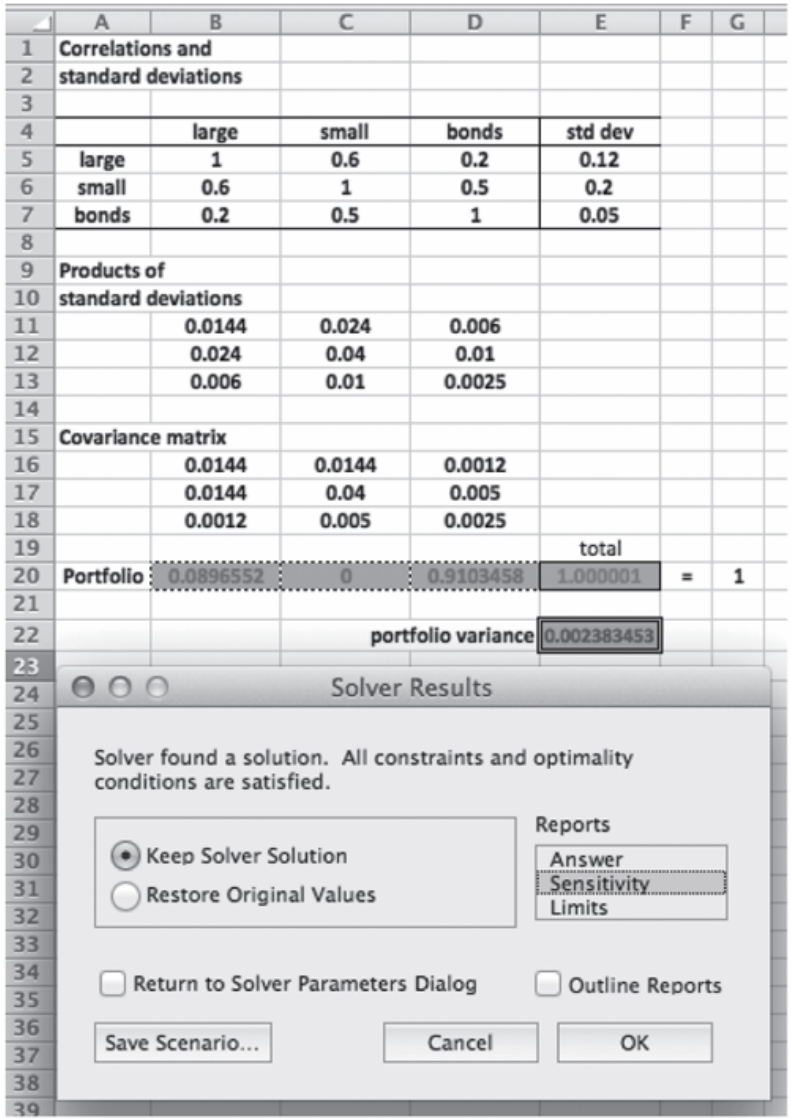

Figure 2: Requesting sensitivity report in Solver

Duality and Optimality Conditions

As in linear programming, there is a dual quadratic program associated with every primal quadratic programming problem, and this dual can be obtained via the Lagrangian function. Throughout this section consider the primal quadratic program

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{D}\boldsymbol{x} \geq \boldsymbol{d} \end{aligned} \tag{5.5}$$

where $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, $\boldsymbol{D} \in \mathbb{R}^{p\times n}$, $\boldsymbol{d} \in \mathbb{R}^p$, and $\boldsymbol{Q}$ is symmetric and positive semidefinite.
The Lagrangian function associated with Eq.(5.5) is

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) := \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{y}^\top(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) + \boldsymbol{s}^\top(\boldsymbol{d} - \boldsymbol{D}\boldsymbol{x}).$$

The constraints of Eq.(5.5) can be encoded via the Lagrangian function through the following observation: For a given vector $\boldsymbol{x}$

$$\max_{\substack{\boldsymbol{y}\geq\boldsymbol{s}\\ \boldsymbol{s}\geq\boldsymbol{0}}} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}) = \begin{cases} \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \text{ if } \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \text{ and } \boldsymbol{D}\boldsymbol{x} \geq \boldsymbol{d} \\ +\infty \text{ otherwise.} \end{cases}$$

Therefore the primal problem Eq.(5.5) can be written as

$$\min_{\boldsymbol{x}} \max_{\substack{\boldsymbol{y},\boldsymbol{s}\\ \boldsymbol{s}\geq 0}} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}).$$

The dual problem is obtained by flipping the order of the min and max operations:

$$\max_{\substack{\boldsymbol{y},\boldsymbol{s}\\ \boldsymbol{s}\geq 0}} \min_{\boldsymbol{x}} L(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}).$$

It is easy to see that the dual problem can be written as follows:

$$\begin{aligned} \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}} \; & \boldsymbol{b}^\top \boldsymbol{y} + \boldsymbol{d}^\top \boldsymbol{s} - \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \\ \text{s.t.} \; & \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{D}^\top \boldsymbol{s} - \boldsymbol{Q}\boldsymbol{x} = \boldsymbol{c} \\ & \boldsymbol{s} \geq \boldsymbol{0} \end{aligned} \tag{5.6}$$

In particular, when the primal problem is in standard form Eq.(5.2), the dual problem is

$$\begin{aligned} \max_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{s}} \; & \boldsymbol{b}^\top \boldsymbol{y} - \tfrac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \\ \text{s.t.} \; & \boldsymbol{A}^\top \boldsymbol{y} - \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{s} = \boldsymbol{c} \\ & \boldsymbol{s} \geq \boldsymbol{0} \end{aligned}$$

Weak duality

Observe that the dual problem of a quadratic program is again a quadratic program. Note that, unlike the case of linear programming, some primal-like variables $\boldsymbol{x}$ also appear in the dual problem. As in linear programming, there is a deep connection between the primal problem Eq.(5.5) and its dual Eq.(5.6). The next result follows by construction.

Theorem (5.4 Weak duality)

Assume $\boldsymbol{x}$ is a feasible point for Eq.(5.5) and $(\tilde{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{s})$ is a feasible point for Eq.(5.6). Then

$$\boldsymbol{b}^\top \boldsymbol{y} + \boldsymbol{d}^\top \boldsymbol{s} - \frac{1}{2}\tilde{\boldsymbol{x}}^\top \boldsymbol{Q}\tilde{\boldsymbol{x}} \leq \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x}.$$

Proof.

If $\boldsymbol{x}$ and $(\tilde{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{s})$ satisfy the above assumptions then

$$\begin{aligned}
\boldsymbol{b}^\top \boldsymbol{y} + \boldsymbol{d}^\top \boldsymbol{s} - \frac{1}{2}\tilde{\boldsymbol{x}}^\top \boldsymbol{Q}\tilde{\boldsymbol{x}} &\leq (\boldsymbol{A}\boldsymbol{x})^\top \boldsymbol{y} + (\boldsymbol{D}\boldsymbol{x})^\top \boldsymbol{s} - \frac{1}{2}\tilde{\boldsymbol{x}}^\top \boldsymbol{Q}\tilde{\boldsymbol{x}} \\
&= \left(\boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{D}^\top \boldsymbol{s}\right)^\top \boldsymbol{x} - \frac{1}{2}\tilde{\boldsymbol{x}}^\top \boldsymbol{Q}\tilde{\boldsymbol{x}} \\
&= (\boldsymbol{c} + \boldsymbol{Q}\tilde{\boldsymbol{x}})^\top \boldsymbol{x} - \frac{1}{2}\tilde{\boldsymbol{x}}^\top \boldsymbol{Q}\tilde{\boldsymbol{x}} \\
&= \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} - \frac{1}{2}(\boldsymbol{x} - \tilde{\boldsymbol{x}})^\top \boldsymbol{Q}(\boldsymbol{x} - \tilde{\boldsymbol{x}}) \\
&\leq \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x}.
\end{aligned}$$

The following much deeper result also holds. □

Strong duality

Theorem (5.5 Strong duality)

Assume one of the problems (5.5) or (5.6) is feasible. Then this problem is bounded if and only if the other one is feasible. In that case both problems have optimal solutions and their optimal values are the same.

Optimality conditions

Theorem (5.6 Optimality conditions)

The vectors $\boldsymbol{x} \in \mathbb{R}^n$ and $(\tilde{\boldsymbol{x}}, \boldsymbol{y}, \boldsymbol{s}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ are optimal solutions to (5.5) and (5.6) respectively if and only if $\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{Q}\tilde{\boldsymbol{x}}$ and

$$\begin{aligned} \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c} - \boldsymbol{A}^\top \boldsymbol{y} - \boldsymbol{D}^\top \boldsymbol{s} &= \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} &= \boldsymbol{0} \\ \boldsymbol{D}\boldsymbol{x} - \boldsymbol{d} &\geq \boldsymbol{0} \\ \boldsymbol{s} &\geq \boldsymbol{0} \\ (\boldsymbol{D}\boldsymbol{x} - \boldsymbol{d})_i s_i = 0, \quad & i = 1, \ldots, p \end{aligned} \tag{5.7}$$

For a quadratic program in standard form (5.2), the optimality conditions (5.7) can be written as follows:

$$\begin{aligned} -\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} &= \boldsymbol{c} \\ \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0} \\ \boldsymbol{s} &\geq \boldsymbol{0} \\ \boldsymbol{x}_i \boldsymbol{s}_i &= 0, \quad i = 1, \ldots, n \end{aligned} \tag{5.8}$$

The optimality conditions (5.7) can be seen as "saddle-point" conditions for the Lagrangian function

$$L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{y}^\top (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) + \boldsymbol{s}^\top (\boldsymbol{d} - \boldsymbol{D}\boldsymbol{x}).$$

We next discuss the special case of a quadratic program with equality constraints only. Consider the problem

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned} \tag{5.9}$$

where $\boldsymbol{c} \in \mathbb{R}^n$, $\boldsymbol{Q} \in \mathbb{R}^{n\times n}$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$, and $\boldsymbol{Q}$ is symmetric and positive semidefinite. In this case the optimality conditions (5.7) simplify to

$$\begin{aligned} \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c} - \boldsymbol{A}^\top \boldsymbol{y} &= \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} &= \boldsymbol{0} \end{aligned} \tag{5.10}$$

The optimality conditions (5.10) in turn can be stated in terms of the Lagrangian function of (5.9):

$$L(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{y}^\top (\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}).$$

Indeed observe that (5.10) can be succinctly written as

$$\nabla L(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{0}.$$

When $\boldsymbol{Q}$ is positive definite and $\boldsymbol{A}$ has full row rank, problem (5.9) has a unique minimizer $\boldsymbol{x}$ and a unique Lagrange multiplier $\boldsymbol{y}$ given by

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Q} & -\boldsymbol{A}^\top \\ \boldsymbol{A} & \boldsymbol{0} \end{bmatrix}^{-1} \begin{bmatrix} -\boldsymbol{c} \\ \boldsymbol{b} \end{bmatrix}.$$

In particular, if $\boldsymbol{Q}$ is positive definite and $\boldsymbol{A}$ has full row rank, then the minimizer and vector of Lagrange multipliers for the problem

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} \\ \text{s.t.} \ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \end{aligned} \tag{5.11}$$

are respectively

$$\begin{aligned} \boldsymbol{x}^* &= \boldsymbol{Q}^{-1}\boldsymbol{A}^\top \left(\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top\right)^{-1} \boldsymbol{b} \\ \boldsymbol{y}^* &= \left(\boldsymbol{A}\boldsymbol{Q}^{-1}\boldsymbol{A}^\top\right)^{-1} \boldsymbol{b} \end{aligned}$$

Example 5.7 (Asset allocation)

Consider the same problem as in Example 5.1 but assume this time that the portfolio is allowed to hold short positions.
The formulation for this modification of Example 5.1 is straightforward: just drop the non-negativity constraint on the variables. Thus we obtain the quadratic programming model

$$\begin{array}{ll} \min_{\boldsymbol{x}} & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{V}\boldsymbol{x} \\ \text{s.t.} & \mathbf{1}^\top \boldsymbol{x} = 1 \end{array}$$

From the above discussion it readily follows that the optimal solution and Lagrange multiplier are

$$\boldsymbol{x}^* = \frac{1}{\mathbf{1}^\top \boldsymbol{V}^{-1}\mathbf{1}} \boldsymbol{V}^{-1}\mathbf{1}$$

$$y^* = \frac{1}{\mathbf{1}^\top \boldsymbol{V}^{-1}\mathbf{1}}$$

Example 5.7 (Asset allocation)

For the particular value of $\boldsymbol{V}$ in Example 5.1 we get the following optimal solution and Lagrange multiplier

$$\boldsymbol{x}^* = \begin{bmatrix} 0.1934 \\ -0.1406 \\ 0.9472 \end{bmatrix}, \quad y^* = 0.001897074$$

Active-Set Methods

Active-set methods are based on the following key observation. Assume $\bar{\boldsymbol{x}}$ is an optimal solution to (5.5) and

$$I := \{i = 1, \ldots, p : (\boldsymbol{D}\bar{\boldsymbol{x}} - \boldsymbol{d})_i = 0\}.$$

Then the optimality conditions (5.7) can be rewritten as

$$\begin{aligned} \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c} - \boldsymbol{A}^\top \boldsymbol{y} - \boldsymbol{D}_I^\top \boldsymbol{s}_I &= \boldsymbol{0} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} &= \boldsymbol{0} \\ \boldsymbol{D}_I\boldsymbol{x} - \boldsymbol{d}_I &= \boldsymbol{0} \\ \boldsymbol{s}_I &\geq \boldsymbol{0} \end{aligned} \tag{5.12}$$

If we ignore the last constraint $\boldsymbol{s}_I \geq \mathbf{0}$, the remaining conditions in (5.12) are precisely the optimality conditions of the problem

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{D}_I \boldsymbol{x} = \boldsymbol{d}_I \end{aligned} \tag{5.13}$$

This suggests an algorithmic strategy to solve (5.5): guess the active set I and solve the subproblem (5.13). If the solution $\bar{\boldsymbol{x}}$ to this subproblem satisfies the other conditions in (5.7) then stop. Otherwise, make a new guess for I. Algorithm 5.1 gives a possible version of this strategy.
Each main iteration of Algorithm 5.1 requires solving the following subproblem for some current trial solution $\bar{\boldsymbol{x}}$ and trial active set I :

Each main iteration of Algorithm 5.1 requires solving the following subproblem for some current trial solution $\bar{\boldsymbol{x}}$ and trial active set I :

$$\begin{aligned} \min_{\Delta \boldsymbol{x}} & \frac{1}{2}(\Delta \boldsymbol{x})^{\top} \boldsymbol{Q} \Delta \boldsymbol{x}+(\boldsymbol{Q} \bar{\boldsymbol{x}}+\boldsymbol{c})^{\top} \Delta \boldsymbol{x} \\ \text{s.t. } & \boldsymbol{A} \Delta \boldsymbol{x}=\mathbf{0} \\ & \boldsymbol{D}_I \Delta \boldsymbol{x}=\mathbf{0} \end{aligned} \tag{5.14}$$

To update the trial solution we also need to compute the step length

$$\alpha:=\min \left\{1, \min_{\substack{i \notin I \\ \boldsymbol{D}_i \Delta \boldsymbol{x}<0}} \frac{\boldsymbol{d}_i-\boldsymbol{D}_i \Delta \boldsymbol{x}}{\boldsymbol{D}_i \Delta \boldsymbol{x}}\right\} \tag{5.15}$$

Algorithm 5.1 Active-set method

1: Choose $\boldsymbol{x}_0$ feasible for (5.5) and $I_0 \subseteq \{i : \boldsymbol{D}_i \boldsymbol{x}_0 = d_i, i = 1, \ldots, p\}$
2: **for** $k = 0, 1, \ldots$ **do**
3: Solve (5.14) for $I = I_k$ and $\bar{\boldsymbol{x}} = \boldsymbol{x}_k$
4: **if** $\Delta \boldsymbol{x} = 0$ **then**
5: Compute the Lagrange multipliers $\bar{\boldsymbol{s}}_I$ of (5.14) for $I = I_k$ and $\bar{\boldsymbol{x}} = \boldsymbol{x}_k$
6: **if** $\bar{\boldsymbol{s}}_I \geq 0$ **then HALT** $\bar{\boldsymbol{x}}$ is an optimal solution to (5.5)
7: **else**
8: Let $j := \arg\min_{i \in I} \bar{\boldsymbol{s}}_i$, $I_{k+1} := I_k \setminus \{j\}$, and $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k$
9: **end if**
10: **else**
11: Compute α via (5.15) for $I = I_k$ and let $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha \Delta \boldsymbol{x}$
12: **if** α_k has a blocking constraint j **then** $I_{k+1} := I_k \cup \{j\}$
13: **else** $I_{k+1} := I_k$
14: **end if**
15: **end if**
16: **end for**

Table 1: Active-set method algorithm

We say that the step length α computed in (5.15) has a blocking constraint, $j \notin I$, if

$$\alpha = \min_{\substack{i \neq I \\ \boldsymbol{D}_i \Delta x < 0}} \frac{\boldsymbol{d}_i - \boldsymbol{D}_i \Delta \boldsymbol{x}}{\boldsymbol{D}_i \Delta \boldsymbol{x}} = \frac{\boldsymbol{d}_j - \boldsymbol{D}_j \Delta \boldsymbol{x}}{\boldsymbol{D}_j \Delta \boldsymbol{x}} < 1.$$

And we say that α has no blocking constraints when

$$\alpha = 1 < \min_{\substack{i \neq I \\ \boldsymbol{D}_i \Delta x < 0}} \frac{\boldsymbol{d}_i - \boldsymbol{D}_i \Delta \boldsymbol{x}}{\boldsymbol{D}_i \Delta \boldsymbol{x}}$$

Interior-Point Methods

For notational convenience and without loss of generality we assume that the problem of interest is in standard form (5.2).
As in the linear programming case, interior-point methods generate a sequence of iterates that satisfy $\boldsymbol{x}, \boldsymbol{s} > \mathbf{0}$. Each iteration of the algorithm aims to make progress towards satisfying $-\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} = \boldsymbol{c}$, $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, and $x_i s_i = 0$, with $i = 1, \ldots, n$.
As before we use the following notational convention: Given a vector $\boldsymbol{x} \in \mathbb{R}^n$, let $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ denote the diagonal matrix defined by $X_{ii} = x_i$, with $i = 1, \ldots, n$, and let $\mathbf{1} \in \mathbb{R}^n$ denote the vector whose components are all 1 s. The optimality conditions (5.8) can be restated as

$$\begin{bmatrix} -\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} - \boldsymbol{c} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{X}\boldsymbol{S}\mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{x}, \boldsymbol{s} \geq \mathbf{0}.$$

Given $\mu > 0$, let $(\boldsymbol{x}(\mu), \boldsymbol{y}(\mu), \boldsymbol{s}(\mu))$ be the solution to the following perturbed version of the above optimality conditions:

$$\begin{bmatrix} -\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} - \boldsymbol{c} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{X}\boldsymbol{S}1 \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{0} \\ \mu \boldsymbol{1} \end{bmatrix}, \quad \boldsymbol{x}, \boldsymbol{s} > \boldsymbol{0}$$

The first condition above can be written as $\boldsymbol{r}_\mu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = \boldsymbol{0}$ for the residual vector

$$\boldsymbol{r}_\mu(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) := \begin{bmatrix} -\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{A}^\top \boldsymbol{y} + \boldsymbol{s} - \boldsymbol{c} \\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \\ \boldsymbol{X}\boldsymbol{S}\boldsymbol{1} - \mu \boldsymbol{1} \end{bmatrix}$$

The central path is the set $\{(\boldsymbol{x}(\mu), \boldsymbol{y}(\mu), \boldsymbol{s}(\mu)) : \mu > 0\}$. It is intuitively clear that $(\boldsymbol{x}(\mu), \boldsymbol{y}(\mu), \boldsymbol{s}(\mu))$ converges to an optimal solution to both (5.2) and its dual. This suggests the following algorithmic strategy. Suppose $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ is "near" $(\boldsymbol{x}(\mu), \boldsymbol{y}(\mu), \boldsymbol{s}(\mu))$ for some $\mu > 0$. Use $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ to move to a better point $(\boldsymbol{x}^+, \boldsymbol{y}^+, \boldsymbol{s}^+)$"near" $(\boldsymbol{x}(\mu^+), \boldsymbol{y}(\mu^+), \boldsymbol{s}(\mu^+))$for some $\mu^+ < \mu$.

It can be shown that if a point $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ is on the central path, then the corresponding value of μ satisfies $\boldsymbol{x}^\top \boldsymbol{s} = n\mu$. Likewise, given $\boldsymbol{x}, \boldsymbol{s} > \boldsymbol{0}$, define

$$\mu(\boldsymbol{x}, \boldsymbol{s}) := \frac{\boldsymbol{x}^\top \boldsymbol{s}}{n}$$

To move from a current point $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s})$ to a new point, we use the so-called Newton step; that is, the solution to the system of equations

$$\begin{bmatrix} -\boldsymbol{Q} & \boldsymbol{A}^\top & \boldsymbol{I} \\ \boldsymbol{A} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{S} & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x} \\ \Delta \boldsymbol{y} \\ \Delta \boldsymbol{s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{c} + \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{y} - \boldsymbol{s} \\ \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} \\ \mu \boldsymbol{1} - \boldsymbol{X}\boldsymbol{S}\boldsymbol{1} \end{bmatrix} \tag{5.16}$$

Algorithm 5.2 presents a template for an interior-point method.

Algorithm 5.2 Interior-point method for quadratic programming

1: Choose $\boldsymbol{x}^0$, $\boldsymbol{s}^0 > 0$
2: **for** $k = 0, 1, \ldots$ **do**
3: Solve the Newton system (5.16) for $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) = (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{s}^k)$ and $\mu := 0.1\mu(\boldsymbol{x}^k, \boldsymbol{s}^k)$
4: Choose a step length $\alpha \in (0, 1]$ and set $(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{s}^{k+1}) = (\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{s}^k) + \alpha(\Delta\boldsymbol{x}, \Delta\boldsymbol{y}, \Delta\boldsymbol{s})$
5: **end for**

Table 2: Interior-point method for quadratic programming

The step length α in step 4 should be chosen so that $\boldsymbol{x}^{k+1}, \boldsymbol{s}^{k+1} > 0$ and the size of $\boldsymbol{r}_\mu\left(\boldsymbol{x}^{k+1}, \boldsymbol{y}^{k+1}, \boldsymbol{s}^{k+1}\right)$ is sufficiently smaller than $\boldsymbol{r}_\mu\left(\boldsymbol{x}^k, \boldsymbol{y}^k, \boldsymbol{s}^k\right)$. A linesearch procedure such as the one described in Algorithm 2.4 in Chapter 2 can be used for choosing the step length α.

Applications to Machine Learning

- Binary Classification and Support Vector Machines.
- Ridge and Lasso Regression.

Binary Classification and Support Vector Machines

Consider feature vectors $\boldsymbol{a}_i \in \mathbb{R}^n$ for $i = 1, \ldots, k_1$ corresponding to class 1 , and vectors $\boldsymbol{b}_i \in \mathbb{R}^n$ for $i = 1, \ldots, k_2$ corresponding to class 2 . If these two vector sets can be linearly separated, a hyperplane $\boldsymbol{w}^\top \boldsymbol{x} = \gamma$ exists with $\boldsymbol{w} \in \mathbb{R}^n, \gamma \in \mathbb{R}$ such that

$$\boldsymbol{w}^\top \boldsymbol{a}_i \geq \gamma, \text{ for } i = 1, \ldots, k_1$$
$$\boldsymbol{w}^\top \boldsymbol{b}_i \leq \gamma, \text{ for } i = 1, \ldots, k_2.$$

To have a "strict" separation, we often prefer to obtain w and γ such that

$$\boldsymbol{w}^\top \boldsymbol{a}_i \geq \gamma + 1, \text{ for } i = 1, \ldots, k_1$$
$$\boldsymbol{w}^\top \boldsymbol{b}_i \leq \gamma - 1, \text{ for } i = 1, \ldots, k_2.$$

In this manner, we find two parallel lines $\left(\boldsymbol{w}^\top \boldsymbol{x} = \gamma + 1 \text{ and } \boldsymbol{w}^\top \boldsymbol{x} = \gamma - 1\right)$ that form the boundaries of the class 1 and class 2 portions of the vector space; see Figure 3.

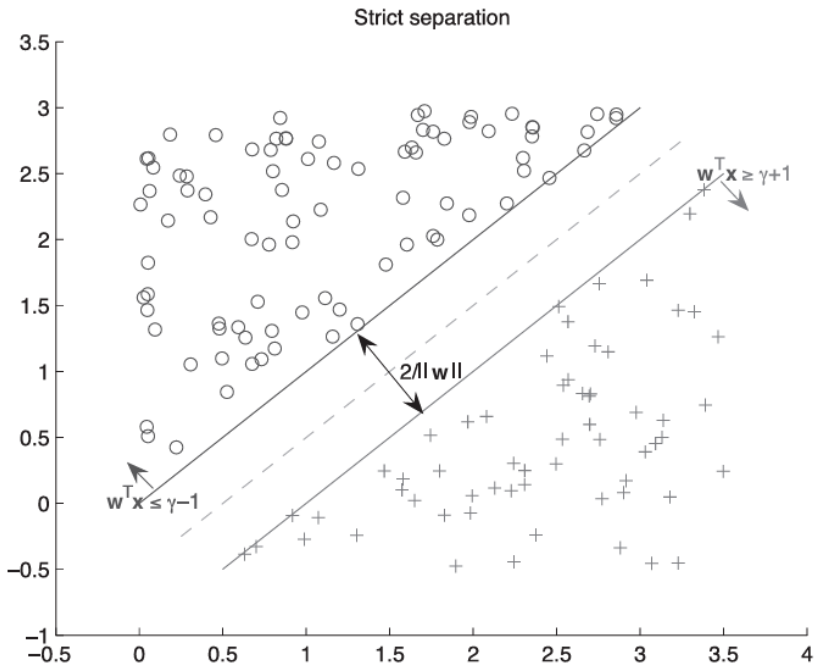

Figure 3: Linear separation of two classes of data points

Support Vector Machine

(i) Consider the following quadratic problem:

$$\begin{aligned} &\min_{\boldsymbol{w},\gamma} \|\boldsymbol{w}\|_2^2 \\ &\boldsymbol{a}_i^\top \boldsymbol{w} \geq \gamma + 1, \text{ for } i = 1, \ldots, k_1 \\ &\boldsymbol{b}_i^\top \boldsymbol{w} \leq \gamma - 1, \text{ for } i = 1, \ldots, k_2. \end{aligned} \tag{5.17}$$

The objective function of this problem is equivalent to maximizing the margin between the lines $\boldsymbol{w}^\top \boldsymbol{x} = \gamma + 1$ and $\boldsymbol{w}^\top \boldsymbol{x} = \gamma - 1$ (see Exercise 5.6).

(ii) The linear separation idea we presented above can be used even when the two vector sets $\{\boldsymbol{a}_i\}$ and $\{\boldsymbol{b}_i\}$ are not linearly separable. (Note that linearly inseparable sets will result in an infeasible problem in formulation (5.17).) This is achieved by introducing a non-negative violation variable for each constraint of (5.17). Then, one has two objectives: to minimize the total of the constraint violations and to maximize the margin. One can formulate a quadratic programming model that combines these two objectives using an adjustable parameter that can be chosen in a way to put more weight on violations or margin, depending on one's preference (see Exercise 5.7).

Ridge and Lasso Regression

Recall the regression problem described in Example 5.3, namely to estimate the linear model

$$Y = \boldsymbol{\beta}^\top X + \epsilon,$$

where X and Y are $\mathbb{R}^p$-valued and $\mathbb{R}$-valued random variables, by using some training data $(\boldsymbol{x}_i, y_i)$, with $i = 1, \ldots, N$.
When $p < N$ the $p \times p$ matrix $\boldsymbol{X}^\top \boldsymbol{X}$ has rank at most $N < p$ and thus the least-squares approach

$$\min_{\boldsymbol{\beta}} \left\| \boldsymbol{X}^\top \boldsymbol{\beta} - \boldsymbol{y} \right\|_2^2$$

is inadequate because the optimality conditions lead to an underdetermined system of equations 0

$$\left(\boldsymbol{X}^\top \boldsymbol{X} \right) \boldsymbol{\beta} = \boldsymbol{X}^\top \boldsymbol{y}$$

We next describe two popular modifications to the ordinary least-squares approach that aim to rectify this difficulty, namely ridge regression and lasso regression. Ridge regression adds a quadratic penalty term to the objective function in the least-squares model

$$\min_{\boldsymbol{\beta}} \left\| \boldsymbol{X}^\top \boldsymbol{\beta} - \boldsymbol{y} \right\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \tag{5.18}$$

where $\lambda > 0$ is a tuning parameter. The effect of the penalty term is to shrink the regression coefficients towards zero. The magnitude of λ determines the shrinking effect. In the limit when $\lambda \to \infty$ the solution to the ridge regression model is $\boldsymbol{\beta} = \mathbf{0}$. On the other hand, when $\lambda = 0$ ridge regression and ordinary least squares coincide.

The optimality conditions for (5.18) yield the following system of equations:

$$\left(\boldsymbol{X}^{\top}\boldsymbol{X}+\lambda\boldsymbol{I}\right)\boldsymbol{\beta}-\boldsymbol{X}^{\top}\boldsymbol{y}=\mathbf{0}$$

Thus the solution to (5.18) is

$$\boldsymbol{\beta}=\left(\boldsymbol{X}^{\top}\boldsymbol{X}+\lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\top}\boldsymbol{y}.$$

On the other hand, the lasso regression model, proposed in a seminal paper by Tibshirani (1996), adds a 1-norm penalty term to the objective function in the least-squares model

$$\min_{\boldsymbol{\beta}} \left\| \boldsymbol{X}^\top \boldsymbol{\beta} - \boldsymbol{y} \right\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \tag{5.19}$$

where $\lambda > 0$ is a tuning parameter. The effect of the penalty term is again to shrink the regression coefficients towards zero. However, the properties of the 1 -norm have a far more interesting effect. The penalty term $\lambda\|\boldsymbol{\beta}\|_1$ makes some of the regression coefficients be equal to zero. In particular, the solutions to the lasso regression model (5.19) are typically sparse and the level of sparsity is controlled by the tuning parameter λ. Lasso regression can be formulated as a quadratic program (see Exercise 5.9). Unlike ridge regression, there is no closed form formula for the solution to lasso regression.