CHAPTER 24

Quadratic Programming

In Chapter 23, we studied a generalization of the linear programming problem in which variables were constrained to take on integer values. In this chapter, we consider a generalization of a different kind. Namely, we shall study the class of problems that would be linear programs except that the objective function is permitted to include terms involving products of pairs of variables. Such terms are called *quadratic terms*, and the problems we shall study are called *quadratic programming* problems.

We have two reasons for being interested in quadratic programming problems. First, on the practical side, there are many real-world optimization problems that fall into this category. This is so because most real-world applications have an element of uncertainty to them and that uncertainty is modeled by including a sum of squares deviation, i.e. variance, as a measure of the robustness of the solution. It is often possible to arrange it so that these quadratic robustness terms appear only in the objective function. The quadratic version of the portfolio selection problem studied in Chapter 13 is one such example—there are many others. The second reason for our interest in quadratic programming problems is that they form a bridge to the much broader subject of convex programming that we shall take up in Chapter 25.

We begin this chapter with a quadratic variant of the portfolio selection problem.

1. The Markowitz Model

Harry Markowitz received the 1990 Nobel Prize in Economics for his portfolio optimization model in which the tradeoff between risk and reward is explicitly treated. We shall briefly describe this model in its simplest form. We start by reintroducing the basic framework of the problem. Those who have read Chapter 13 will note that the first few paragraphs here are a repeat of what was written there.

Given a collection of potential investments (indexed, say, from 1 to n), let R_j denote the return in the next time period on investment j, $j = 1, \ldots, n$. In general, R_j is a random variable, although some investments may be essentially deterministic.

A *portfolio* is determined by specifying what fraction of one's assets to put into each investment. That is, a portfolio is a collection of nonnegative numbers x_j,

R.J. Vanderbei, *Linear Programming*, International Series in Operations Research & Management Science 196, DOI 10.1007/978-1-4614-7630-6_24,

$j = 1, \ldots, n$, that sum to one. The return (on each dollar) one would obtain using a given portfolio is given by

$$R = \sum_j x_j R_j.$$

The *reward* associated with such a portfolio is defined as the expected return:

$$\mathbb{E}R = \sum_j x_j \mathbb{E}R_j.$$

If reward alone were the issue, the problem would be trivial: simply put everything in the investment with the highest expected return. But unfortunately, investments with high reward typically also carry a high level of risk. That is, even though they are expected to do very well in the long run, they also tend to be erratic in the short term. Markowitz defined the *risk* associated with an investment to be the variance of the return:

$$\begin{aligned} \text{Var}(R) &= \mathbb{E}(R - \mathbb{E}R)^2 \\ &= \mathbb{E}\left(\sum_j x_j (R_j - \mathbb{E}R_j)\right)^2 \\ &= \mathbb{E}\left(\sum_j x_j \tilde{R}_j\right)^2, \end{aligned}$$

where $\tilde{R}_j = R_j - \mathbb{E}R_j$. One would like to maximize the reward while at the same time not incur excessive risk. In the Markowitz model, one forms a linear combination of the mean and the variance (parametrized here by μ) and minimizes that:

$$\begin{aligned} \text{minimize} \quad & -\sum_j x_j \mathbb{E}R_j + \mu \mathbb{E}\left(\sum_j x_j \tilde{R}_j\right)^2 \\ \text{subject to} \quad & \sum_j x_j = 1 \\ & x_j \geq 0 \qquad j = 1, 2, \ldots, n. \end{aligned} \tag{24.1}$$

Here, as in Chapter 13, μ is a positive parameter that represents the importance of risk relative to reward: high values of μ tend to minimize risk at the expense of reward, whereas low values put more weight on reward.

Again, as in Chapter 13, whenever there are individual investments that are negatively correlated, i.e. one is likely to go up exactly on those days where the other is likely to go down, it is wise to buy some of each. This is called hedging. In statistics, the so-called covariance matrix is the key to identifying negative correlations. And, the covariance matrix is what appears in the Markowitz model. To see it, let us expand the square in our expression for the variance of the portfolio:

$$\begin{aligned}\mathbb{E}\left(\sum_j x_j\tilde{R}_j\right)^2 &= \mathbb{E}\left(\sum_i x_i\tilde{R}_i\right)\left(\sum_j x_j\tilde{R}_j\right)\\ &= \mathbb{E}\left(\sum_i\sum_j x_i x_j\tilde{R}_i\tilde{R}_j\right)\\ &= \sum_i\sum_j x_i x_j\mathbb{E}(\tilde{R}_i\tilde{R}_j)\\ &= \sum_i\sum_j x_i x_j C_{i,j},\end{aligned}$$

where

$$C_{i,j} = \mathbb{E}(\tilde{R}_i\tilde{R}_j)$$

is the covariance matrix. Hence, problem (24.1) can be rewritten as

$$\begin{aligned}&\text{minimize} \quad -\sum_j r_j x_j + \mu\sum_i\sum_j x_i x_j C_{i,j}\\ &\text{subject to} \quad \sum_j x_j = 1\\ &\qquad\qquad x_j \geq 0 \qquad j = 1, 2, \ldots, n,\end{aligned} \tag{24.2}$$

where we have introduced $r_j = \mathbb{E}R_j$ for the mean return on investment j.

Solving problem (24.2) requires an estimate of the mean return for each of the investments as well as an estimate of the covariance matrix. However, these quantities are not known theoretically but instead must be estimated by looking at historical data. For example, Table 24.1 shows annual returns from 1973 to 1994 for eight different possible investments: U.S. 3-Month T-Bills, U.S. Government Long Bonds, S&P 500, Wilshire 5000 (a collection of small company stocks), NASDAQ Composite, Lehman Brothers Corporate Bonds Index, EAFE (a securities index for Europe, Asia, and the Far East), and Gold. Let $R_j(t)$ denote the return on investment j in year $1972 + t$. One way to estimate the mean $\mathbb{E}R_j$ is simply to take the average of the historical returns:

$$r_j = \mathbb{E}R_j = \frac{1}{T}\sum_{t=1}^{T} R_j(t).$$

There are two drawbacks to this simple formula. First, whatever happened in 1973 certainly has less bearing on the future than what happened in 1994. Hence, giving all the past returns equal weight puts too much emphasis on the distant past at the expense of the recent past. A better estimate is obtained by using a discounted sum:

$$\mathbb{E}R_j = \frac{\sum_{t=1}^{T} p^{T-t}R_j(t)}{\sum_{t=1}^{T} p^{T-t}}.$$

Here, p is a discount factor. Putting $p = 0.9$ gives a weighted average that puts more weight on the most recent years. To see the effect of discounting the past, consider the Gold investment. The unweighted average return is 1.129, whereas the weighted

Year	US 3-Month T-Bills	US Gov. Long Bonds	S&P 500	Wilshire 5000	NASDAQ Composite	Lehman Bros. Corp. Bonds	EAFE	Gold
1973	1.075	0.942	0.852	0.815	0.698	1.023	0.851	1.677
1974	1.084	1.020	0.735	0.716	0.662	1.002	0.768	1.722
1975	1.061	1.056	1.371	1.385	1.318	1.123	1.354	0.760
1976	1.052	1.175	1.236	1.266	1.280	1.156	1.025	0.960
1977	1.055	1.002	0.926	0.974	1.093	1.030	1.181	1.200
1978	1.077	0.982	1.064	1.093	1.146	1.012	1.326	1.295
1979	1.109	0.978	1.184	1.256	1.307	1.023	1.048	2.212
1980	1.127	0.947	1.323	1.337	1.367	1.031	1.226	1.296
1981	1.156	1.003	0.949	0.963	0.990	1.073	0.977	0.688
1982	1.117	1.465	1.215	1.187	1.213	1.311	0.981	1.084
1983	1.092	0.985	1.224	1.235	1.217	1.080	1.237	0.872
1984	1.103	1.159	1.061	1.030	0.903	1.150	1.074	0.825
1985	1.080	1.366	1.316	1.326	1.333	1.213	1.562	1.006
1986	1.063	1.309	1.186	1.161	1.086	1.156	1.694	1.216
1987	1.061	0.925	1.052	1.023	0.959	1.023	1.246	1.244
1988	1.071	1.086	1.165	1.179	1.165	1.076	1.283	0.861
1989	1.087	1.212	1.316	1.292	1.204	1.142	1.105	0.977
1990	1.080	1.054	0.968	0.938	0.830	1.083	0.766	0.922
1991	1.057	1.193	1.304	1.342	1.594	1.161	1.121	0.958
1992	1.036	1.079	1.076	1.090	1.174	1.076	0.878	0.926
1993	1.031	1.217	1.100	1.113	1.162	1.110	1.326	1.146
1994	1.045	0.889	1.012	0.999	0.968	0.965	1.078	0.990

TABLE 24.1. Returns per dollar for each of eight investments over several years. That is, \$1 invested in US 3-Month T-Bills on January 1, 1973, was worth \$1.075 on December 31, 1973.

average is 1.053. Most experts in 1995 felt that a $5.3\,\%$ return represented a more realistic expectation than a $12.9\,\%$ return. In the results that follow, all expectations are estimated by computing weighted averages using $p = 0.9$.

The second issue concerns the estimation of means (not variances). An investment that returns 1.1 one year and 0.9 the next has an (unweighted) average return of 1, that is, no gain or loss. However, one dollar invested will actually be worth $(1.1)(0.9) = 0.99$ at the end of the second year. While a $1\,\%$ error is fairly small, consider what happens if the return is 2.0 one year and then 0.5 the next. Clearly, the value of one dollar at the end of the 2 years is $(2.0)(0.5) = 1$, but the average of the two returns is $(2.0 + 0.5)/2 = 1.25$. There is a very significant difference between an investment that is flat and one that yields a $25\,\%$ return in 2 years. This is obviously an effect for which a correction is required. We need to average 2.0 and 0.5 in such a way that they cancel out—and this cancellation must work not only for 2.0 and 0.5 but for every positive number and its reciprocal. The trick is to average the logarithm of the returns (and then exponentiate the average). The logarithm has the correct effect of cancelling a return r and its reciprocal:

$$\log r + \log \frac{1}{r} = 0.$$

Hence, we estimate means from Table 24.1 using

$$\mathbb{E}R_j = \exp\left(\frac{\sum_{t=1}^{T} p^{T-t} \log R_j(t)}{\sum_{t=1}^{T} p^{T-t}}\right).$$

μ	Gold	US 3-Month T-Bills	Lehman Bros. Corp. Bonds	NASDAQ Composite	S&P 500	EAFE	Mean	Std. dev.
0.0						1.000	1.122	0.227
0.1					0.603	0.397	1.121	0.147
1.0					0.876	0.124	1.120	0.133
2.0		0.036	0.322		0.549	0.092	1.108	0.102
4.0		0.487	0.189		0.261	0.062	1.089	0.057
8.0		0.713	0.123		0.117	0.047	1.079	0.037
1024.0	0.008	0.933	0.022	0.016		0.022	1.070	0.028

TABLE 24.2. Optimal portfolios for several choices of μ.

This estimate for Gold gives an estimate of its return at 2.9 %, which is much more in line with the beliefs of experts (at least in 1995).

Table 24.2 shows the optimal portfolios for several choices of μ. The corresponding optimal values for the mean and standard deviation (which is defined as the square root of the variance) are plotted in Figure 24.1. Letting μ vary continuously generates a curve of optimal solutions. This curve is called the *efficient frontier*. Any portfolio that produces a mean–variance combination that does not lie on the efficient frontier can be improved either by increasing its mean without changing the variance or by decreasing the variance without changing the mean. Hence, one should only invest in portfolios that lie on the efficient frontier.

Of course, the optimal portfolios shown in Table 24.2 were obtained by solving (24.1). The rest of this chapter is devoted to describing an algorithm for solving quadratic programs such as this one.

2. The Dual

We have seen that duality plays a fundamental role in our understanding and derivation of algorithms for linear programming problems. The same is true for quadratic programming. Hence, our first goal is to figure out what the dual of a quadratic programming problem should be.

Quadratic programming problems are usually formulated as minimizations. Therefore, we shall consider problems given in the following form:

$$\begin{aligned} &\text{minimize} && c^T x + \tfrac{1}{2} x^T Q x \\ &\text{subject to} && Ax \geq b \\ &&& x \geq 0. \end{aligned} \tag{24.3}$$

Of course, we may (and do) assume that the matrix Q is symmetric (see Exercise 24.2). Note that we have also changed the sense of the inequality constraints from our usual less-than to greater-than. This change is not particularly important—its only purpose is to maintain a certain level of parallelism with past formulations (that is, minimizations have always gone hand-in-hand with greater-than constraints, while maximizations have been associated with less-than constraints).

In Chapter 5, we derived the dual problem by looking for tight bounds on the optimal solution to the primal problem. This approach could be followed here, but

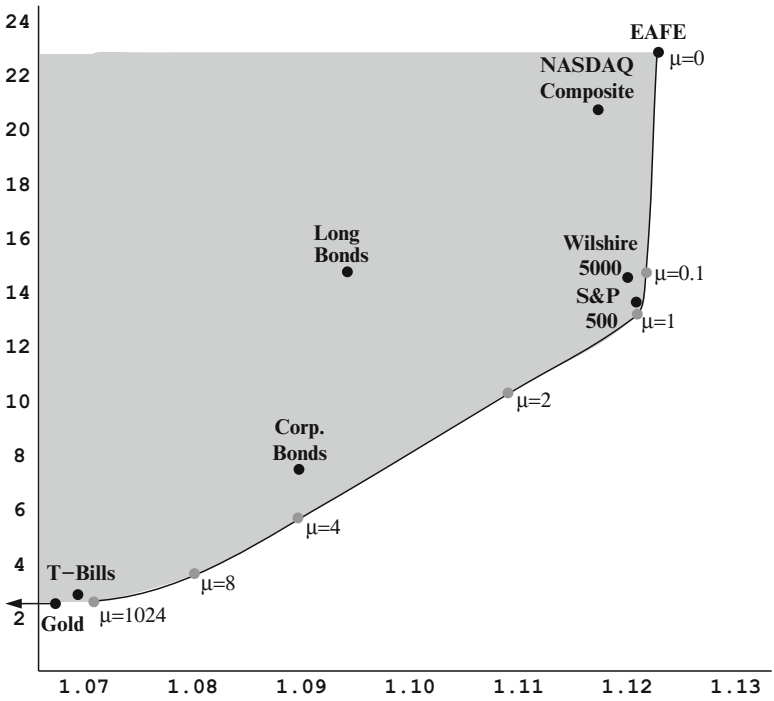

FIGURE 24.1. The efficient frontier.

it seems less compelling in the context of quadratic programming. A more direct approach stems from the connection between duality and the first-order optimality conditions for the barrier problem that we examined in Chapter 17. Indeed, let us start by writing down the barrier problem associated with (24.3). To this end, we introduce a nonnegative vector w of surplus variables and then subtract a barrier term for each nonnegative variable to get the following barrier problem:

$$\begin{array}{ll} \text{minimize} & c^T x + \frac{1}{2} x^T Q x - \mu \sum_j \log x_j - \mu \sum_i \log w_i \\ \text{subject to} & Ax - w = b. \end{array}$$

Next, we introduce the Lagrangian:

$$\begin{aligned} f(x, w, y) = c^T x + \frac{1}{2} x^T Q x - \mu \sum_j \log x_j - \mu \sum_i \log w_i \\ + y^T (b - Ax + w). \end{aligned}$$

The first-order optimality conditions for the barrier problem are obtained by differentiating the Lagrangian with respect to each of its variables and setting these derivatives to zero. In vector notation, setting to zero the derivative with respect to the x variables gives

$$c + Qx - \mu X^{-1} e - A^T y = 0.$$

Similarly, setting to zero the derivatives with respect to the w and y variables gives

$$\begin{aligned} -\mu W^{-1}e + y &= 0 \\ b - Ax + w &= 0, \end{aligned}$$

respectively. As we did in our study of linear programming problems, we now introduce a new vector z given by

$$z = \mu X^{-1}e.$$

With this definition, the first-order optimality conditions can be summarized as

$$\begin{aligned} A^T y + z - Qx &= c \\ Ax - w &= b \\ XZe &= \mu e \\ YWe &= \mu e. \end{aligned}$$

From the last two conditions, we see that the dual problem involves an n-vector of variables z that are complementary to the primal variables x and an m-vector of variables y that are complementary to the primal slack variables w. Because of these complementarity conditions, we expect that the variables y and z are constrained to be nonnegative in the dual problem. Also, to establish the proper connection between the first-order optimality conditions and the dual problem, we must recognize the first condition as a dual constraint. Hence, the constraints for the dual problem are

$$\begin{aligned} A^T y + z - Qx &= c \\ y, z &\geq 0. \end{aligned}$$

It is interesting to note that the dual constraints involve an n-vector x that seems as if it should belong to the primal problem. This may seem odd, but when understood properly it turns out to be entirely harmless. The correct interpretation is that the variable x appearing in the dual has, in principle, no connection to the variable x appearing in the primal (except that, as we shall soon see, at optimality they will be equal).

The barrier problem has helped us write down the dual constraints, but it does not shed any light on the dual objective function. To see what the dual objective function should be, we look at what it needs to be for the weak duality theorem to hold true. In the weak duality theorem, we assume that we have a primal feasible solution (x, w) and a dual feasible solution (x, y, z). We then follow the obvious chains of equalities:

$$y^T(Ax) = y^T(b + w) = b^T y + y^T w$$

and

$$(A^T y)^T x = (c - z + Qx)^T x = c^T x - z^T x + x^T Qx.$$

Now, since $y^T(Ax) = (A^T y)^T x$, we see that

$$\begin{aligned} 0 \leq y^T w + z^T x &= c^T x + x^T Q x - b^T y \\ &= (c^T x + \frac{1}{2} x^T Q x) - (b^T y - \frac{1}{2} x^T Q x). \end{aligned}$$

From this inequality, we see that the dual objective function is $b^T y - \frac{1}{2} x^T Q x$. Hence, the dual problem can be stated now as

$$\begin{array}{ll} \text{maximize} & b^T y - \frac{1}{2} x^T Q x \\ \text{subject to} & A^T y + z - Q x = c \\ & y, z \geq 0. \end{array}$$

For linear programming, the fundamental connection between the primal and dual problems is summarized in the Complementary Slackness Theorem. In the next section, we shall derive a version of this theorem for quadratic programming.

3. Convexity and Complexity

In linear programming, the dual problem is important because it provides a certificate of optimality as manifest in the Complementary Slackness Theorem. Under certain conditions, the same is true here. Let us start by deriving the analogue of the Complementary Slackness Theorem. The derivation begins with a reiteration of the derivation of the Weak Duality Theorem. Indeed, let (x, w) denote a feasible solution to the primal problem and let $(\bar{x}, y, z)$ denote a feasible solution to the dual problem (we have put a bar on the dual x to distinguish it from the one appearing in the primal). The chain of equalities that form the backbone of the proof of the Weak Duality Theorem are, as always, obtained by writing $y^T A x$ two ways, namely,

$$y^T(Ax) = (A^T y)^T x,$$

and then producing the obvious substitutions

$$y^T(Ax) = y^T(b + w) = b^T y + y^T w$$

and

$$(A^T y)^T x = (c - z + Q\bar{x})^T x = c^T x - z^T x + \bar{x}^T Q x.$$

Comparing the ends of these two chains and using the fact that both $y^T w$ and $z^T x$ are nonnegative, we see that

$$0 \leq y^T w + z^T x = c^T x + \bar{x}^T Q x - b^T y. \tag{24.4}$$

So far, so good.

Now, what about the Complementary Slackness Theorem? In the present context, we expect this theorem to say roughly the following: given a solution (x^*, w^*) that is feasible for the primal and a solution (x^*, y^*, z^*) that is feasible for the dual, if these solutions make inequality (24.4) into an equality, then the primal solution is optimal for the primal problem and the dual solution is optimal for the dual problem.

Let's try to prove this. Let (x, w) be an arbitrary primal feasible solution. Weak duality applied to (x, w) on the primal side and (x^*, y^*, z^*) on the dual side says that

$$c^T x + x^{*T} Q x - b^T y^* \geq 0.$$

But for the specific primal feasible solution (x^*, w^*), this inequality is an equality:

$$c^T x^* + {x^*}^T Q x^* - b^T y^* = 0.$$

Combining these, we get

$$c^T x^* + {x^*}^T Q x^* \leq c^T x + {x^*}^T Q x.$$

This is close to what we want, but not quite it. Recall that our aim is to show that the primal objective function evaluated at x^* is no larger than its value at x. That is,

$$c^T x^* + \frac{1}{2}{x^*}^T Q x^* \leq c^T x + \frac{1}{2}x^T Q x.$$

It is easy to get from the one to the other. Starting from the desired left-hand side, we compute as follows:

$$\begin{aligned} c^T x^* + \frac{1}{2}{x^*}^T Q x^* &= c^T x^* + {x^*}^T Q x^* - \frac{1}{2}{x^*}^T Q x^* \\ &\leq c^T x + {x^*}^T Q x - \frac{1}{2}{x^*}^T Q x^* \\ &= c^T x + \frac{1}{2}x^T Q x - \frac{1}{2}x^T Q x + {x^*}^T Q x - \frac{1}{2}{x^*}^T Q x^* \\ &= c^T x + \frac{1}{2}x^T Q x - \frac{1}{2}(x - x^*)^T Q (x - x^*). \end{aligned}$$

The last step in the derivation is to drop the subtracted term on the right-hand side of the last expression. We can do this if the quantity being subtracted is nonnegative. But is it? In general, the answer is no. For example, if Q were the negative of the identity matrix, then the expression $(x - x^*)^T Q (x - x^*)$ would be negative rather than nonnegative.

So it is here that we must impose a restriction on the class of quadratic programming problems that we study. The correct assumption is that Q is positive semidefinite. Recall from Chapter 19 that a matrix Q is *positive semidefinite* if

$$\xi^T Q \xi \geq 0 \qquad \text{for all } \xi \in \mathbb{R}^n.$$

With this assumption, we can finish the chain of inequalities and conclude that

$$c^T x^* + \frac{1}{2}{x^*}^T Q x^* \leq c^T x + \frac{1}{2}x^T Q x.$$

Since x was an arbitrary primal feasible point, it follows that x^* (together with w^*) is optimal for the primal problem. A similar analysis shows that y^* (together with x^* and z^*) is optimal for the dual problem (see Exercise 24.4).

A quadratic programming problem of the form (24.3) in which the matrix Q is positive semidefinite is called a *convex quadratic programming problem*. The discussion given above can be summarized in the following theorem:

THEOREM 24.1. *For convex quadratic programming problems, given a solution (x^*, w^*) that is feasible for the primal and a solution (x^*, y^*, z^*) that is feasible for the dual, if these solutions make inequality (24.4) into an equality, then the primal solution is optimal for the primal problem and the dual solution is optimal for the dual problem.*

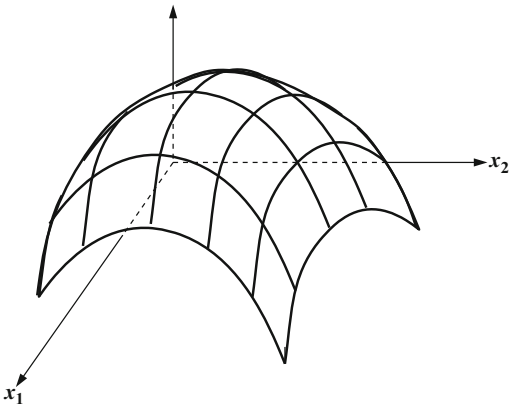

FIGURE 24.2. The objective function for (24.5) in the case where $n = 2$.

To see how bad things are when Q is not positive semidefinite, consider the following example:

$$\begin{array}{ll} \text{minimize} & \sum_j x_j(1-x_j) + \sum_j c_j x_j \\ \text{subject to} & 0 \le x_j \le 1, \qquad j = 1, 2, \ldots, n. \end{array} \tag{24.5}$$

We assume that the coefficients, c_j, $j = 1, 2, \ldots, n$, are small. To be precise, we assume that

$$|c_j| < 1, \qquad j = 1, 2, \ldots, n.$$

Let $f(x)$ denote the value of the objective function at point x. Setting the gradient to zero,

$$\nabla f(x) = e - 2x + c = 0,$$

we see that there is one interior critical point. It is given by

$$x = (e + c)/2$$

(the assumption that c is small guarantees that this x lies in the interior of the feasible set: $0 < x < 1$). However, this critical point is a local maximum, since the matrix of second derivatives is $-2I$. The algebraic details are tedious, but if we look at Figure 24.2, it is easy to be convinced that every vertex of the feasible set is a local minimum. While this particular problem is easy to solve explicitly, it does indicate the essential difficulty associated with nonconvex quadratic programming problems—namely, for such problems one may need to check every vertex individually, and there may be an exponential number of such vertices.

The situation for convex quadratic programming problems is much better, since they inherit most of the properties that made linear programs efficiently solvable. Indeed, in the next section, we derive an interior-point method for quadratic programming problems.

4. Solution via Interior-Point Methods

In this section, we derive an interior-point method for quadratic programming problems. We start from the first-order optimality conditions, which we saw in the last section are given by

$$\begin{aligned} A^T y + z - Qx &= c \\ Ax - w &= b \\ XZe &= \mu e \\ YWe &= \mu e. \end{aligned}$$

Following the derivation given in Chapter 18, we replace (x, w, y, z) with $(x + \Delta x, w + \Delta w, y + \Delta y, z + \Delta z)$ to get the following nonlinear system in $(\Delta x, \Delta w, \Delta y, \Delta z)$:

$$\begin{aligned} A^T \Delta y + \Delta z - Q\Delta x &= c - A^T y - z + Qx =: \sigma \\ A\Delta x - \Delta w &= b - Ax + w \qquad =: \rho \\ Z\Delta x + X\Delta z + \Delta X \Delta Z e &= \mu e - XZe \\ W\Delta y + Y\Delta w + \Delta Y \Delta W e &= \mu e - YWe. \end{aligned}$$

Next, we drop the nonlinear terms to get the following linear system for the step directions $(\Delta x, \Delta w, \Delta y, \Delta z)$:

$$\begin{aligned} A^T \Delta y + \Delta z - Q\Delta x &= \sigma \\ A\Delta x - \Delta w &= \rho \\ Z\Delta x + X\Delta z &= \mu e - XZe \\ W\Delta y + Y\Delta w &= \mu e - YWe. \end{aligned}$$

Following the reductions of Chapter 19, we use the last two equations to solve for Δz and Δw to get

$$\begin{aligned} \Delta z &= X^{-1}(\mu e - XZe - Z\Delta x) \\ \Delta w &= Y^{-1}(\mu e - YWe - W\Delta y). \end{aligned}$$

We then use these expressions to eliminate Δz and Δw from the remaining two equations in the system. After elimination, we arrive at the following *reduced KKT system*:

$$A^T \Delta y - (X^{-1}Z + Q)\Delta x = \sigma - \mu X^{-1} e + z \tag{24.6}$$

$$A\Delta x + Y^{-1}W\Delta y = \rho + \mu Y^{-1} e - w. \tag{24.7}$$

Substituting in the definitions of ρ and σ and writing the system in matrix notation, we get

$$\begin{bmatrix} -(X^{-1}Z + Q) & A^T \\ A & Y^{-1}W \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c - A^T y - \mu X^{-1} e + Qx \\ b - Ax + \mu Y^{-1} e \end{bmatrix}.$$

A summary of the algorithm is shown in Figure 24.3. It should be clear that the quadratic term in the objective function plays a fairly small role. In fact, the

initialize $(x, w, y, z) > 0$

while (not optimal) {

$\rho = b - Ax + w$

$\sigma = c - A^T y - z + Qx$

$\gamma = z^T x + y^T w$

$\mu = \delta \dfrac{\gamma}{n+m}$

solve:

$$\begin{bmatrix} -(X^{-1}Z + Q) & A^T \\ A & Y^{-1}W \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c - A^T y - \mu X^{-1} e + Qx \\ b - Ax + \mu Y^{-1} e \end{bmatrix}.$$

$\Delta z = X^{-1}(\mu e - XZe - Z\Delta x)$

$\Delta w = Y^{-1}(\mu e - YWe - W\Delta y)$

$\theta = r \left(\max_{ij} \left\{ -\dfrac{\Delta x_j}{x_j}, -\dfrac{\Delta w_i}{w_i}, -\dfrac{\Delta y_i}{y_i}, -\dfrac{\Delta z_j}{z_j} \right\} \right)^{-1} \wedge 1$

$x \leftarrow x + \theta \Delta x, \qquad w \leftarrow w + \theta \Delta w$

$y \leftarrow y + \theta \Delta y, \qquad z \leftarrow z + \theta \Delta z$

}

FIGURE 24.3. The path-following method for quadratic programming problems.

convergence analysis given in Chapter 18 can be easily adapted to yield analogous results for quadratic programming problems (see Exercise 24.6).

5. Practical Considerations

For practical implementations of interior-point algorithms, we saw in Chapter 19 that the difficulties created by dense rows/columns suggest that we solve the reduced KKT system using an equation solver that can handle symmetric indefinite systems (such as those described in Chapter 20). Quadratic programming problems give us even more reason to prefer the reduced KKT system. To see why, let us reduce the system further to get a feel for the normal equations for quadratic programming.

If we use (24.6) to solve for Δx and then eliminate it from (24.7), we get

$$\Delta x = -(X^{-1}Z + Q)^{-1} \left(c - A^T y + Qx - \mu X^{-1} e - A^T \Delta y \right)$$

and the associated system of normal equations (in primal form):

$$\begin{aligned} \left(A(X^{-1}Z + Q)^{-1} A^T + Y^{-1}W \right) \Delta y = {} & b - Ax + \mu Y^{-1} e \\ & + A(X^{-1}Z + Q)^{-1} \left(c - A^T y + Qx - \mu X^{-1} e \right). \end{aligned}$$

As we saw in Chapter 19, the most significant disadvantage of the normal equations is that they could involve a dense matrix even when the original constraint matrix is sparse. For quadratic programming, this disadvantage is even more pronounced. Now the matrix of normal equations has the nonzero pattern of $A(D + Q)^{-1}A^T$,

where D is a diagonal matrix. If Q is a diagonal matrix, then this matrix appearing between A and A^T is diagonal, and the system has the same structure as we saw for linear programming. But if Q is not a diagonal matrix, then all hope for any sparsity in $A(D+Q)^{-1}A^T$ is lost.

Fortunately, however, the dual form of the normal equations is likely to retain some sparsity. Indeed, to derive the dual form, we use (24.7) to solve for Δy and then eliminate it from (24.6). The result is

$$\Delta y = YW^{-1}\left(b - Ax + \mu Y^{-1}e - A\Delta x\right)$$

and

$$\begin{aligned} -\left(X^{-1}Z + Q + A^TYW^{-1}A\right)\Delta x = c - A^Ty + Qx - \mu X^{-1}e \\ - A^TYW^{-1}\left(b - Ax + \mu Y^{-1}e\right). \end{aligned}$$

Now the matrix has a nonzero pattern of A^TA+Q. This pattern is much more likely to be sparse than the pattern we had above.

As mentioned earlier, there is significantly less risk of fill-in if Q is diagonal. A quadratic programming problem for which Q is diagonal is called a *separable quadratic programming problem*. It turns out that every nonseparable quadratic programming problem can be replaced by an equivalent separable version, and sometimes this replacement results in a problem that can be solved dramatically faster than the original nonseparable problem. The trick reveals itself when we remind ourselves that the problems we are studying are convex quadratic programs, and so we ask the question: how do we know that the matrix Q is positive semidefinite? Or, more to the point, how does the creator of the model know that Q is positive semidefinite? There are many equivalent characterizations of positive semidefiniteness, but the one that is easiest to check is the one that says that Q is positive semidefinite if and only if it can be factored as follows:

$$Q = F^TDF.$$

Here F is a $k \times n$ matrix and D is a $k \times k$ diagonal matrix having all nonnegative diagonal entries. In fact, the model creator often started with F and D and then formed Q by multiplying. In these cases, the matrix F will generally be less dense than Q. And if k is substantially less than n, then the following substitution is almost guaranteed to dramatically improve the solution time. Introduce new variables y by setting

$$y = Fx.$$

With this definition, the nonseparable quadratic programming problem (24.3) can be replaced by the following equivalent separable one:

$$\begin{aligned} \text{minimize} \quad & c^Tx + \tfrac{1}{2}y^TDy \\ \text{subject to} \quad & Ax \geq b \\ & Fx - y = 0 \\ & x \geq 0. \end{aligned}$$

The cost of separation is the addition of k new constraints. As we said before, if k is small and/or F is sparse, then we can expect this formulation to be solved more efficiently.

To illustrate this trick, let us return to the Markowitz model. Recall that the quadratic terms in this model come from the variance of the portfolio's return, which is given by

$$\begin{aligned}\text{Var}(R) &= \mathbb{E}(\sum_j x_j \tilde{R}_j)^2 \\ &= \sum_{t=1}^{T} p(t) \left(\sum_j x_j \tilde{R}_j(t) \right)^2 .\end{aligned}$$

Here,

$$p(t) = \frac{p^{T-t}}{\sum_{s=1}^{T} p^{T-s}}$$

for $t = 1, 2, \ldots, T$, and

$$\tilde{R}_j(t) = R_j(t) - \sum_{t=1}^{T} p(t) R_j(t).$$

If we introduce the variables,

$$y(t) = \sum_j x_j \tilde{R}_j(t), \qquad t = 1, 2, \ldots, T,$$

then we get the following separable version of the Markowitz model:

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_j x_j \mathbb{E}R_j - \mu \sum_{t=1}^{T} p(t) y(t)^2 \\ \text{subject to} & \displaystyle\sum_j x_j = 1 \\ & y(t) = \sum_j x_j \tilde{R}_j(t), \qquad t = 1, 2, \ldots, T, \\ & x_j \geq 0 \qquad\qquad\qquad\quad j = 1, 2, \ldots, n. \end{array}$$

Using specific data involving 500 possible investments and 20 historical time periods, the separable version solves 60 times faster than the nonseparable version using a QP-solver called LOQO.

Exercises

24.1 Show that the gradient of the function

$$f(x) = \frac{1}{2} x^T Q x$$

is given by

$$\nabla f(x) = Qx.$$

24.2 Suppose that Q is an $n \times n$ matrix that is not necessarily symmetric. Let $\tilde{Q} = \frac{1}{2}(Q + Q^T)$. Show that

(a) $x^T Q x = x^T \tilde{Q} x$, for every $x \in \mathbb{R}^n$, and
(b) $\tilde{Q}$ is symmetric.

24.3 *Penalty Methods*.
(a) Consider the following problem:

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2} x^T Q x \\ &\text{subject to} && Ax = b, \end{aligned}$$

where Q is symmetric, positive semidefinite, and invertible (these last two conditions are equivalent to saying that Q is positive definite). By solving the first-order optimality conditions, give an explicit formula for the solution to this problem.

(b) Each equality constraint in the above problem can be replaced by a *penalty term* added to the objective function. Penalty terms should be small when the associated constraint is satisfied and become rapidly larger as it becomes more and more violated. One choice of penalty function is the quadratic function. The *quadratic penalty problem* is defined as follows:

$$\text{minimize } \frac{1}{2} x^T Q x + \frac{\lambda}{2} (b - Ax)^T (b - Ax),$$

where λ is a large real-valued parameter. Derive an explicit formula for the solution to this problem.

(c) Show that, in the limit as λ tends to infinity, the solution to the quadratic penalty problem converges to the solution to the original problem.

24.4 Consider a convex quadratic programming problem. Suppose that (x^*, w^*) is a feasible solution for the primal and that (x^*, y^*, z^*) is a feasible solution for the dual. Suppose further that these solutions make inequality (24.4) into an equality. Show that the dual solution is optimal for the dual problem.

24.5 A real-valued function f defined on $\mathbb{R}^n$ is called *convex* if, for every $x, y \in \mathbb{R}^n$, and for every $0 < t < 1$,

$$f(tx + (1-t)y) \leq t f(x) + (1-t) f(y).$$

Show that the function

$$f(x) = c^T x + \frac{1}{2} x^T Q x, \qquad x \in \mathbb{R}^n,$$

is convex if Q is positive semidefinite.

24.6 Extend the convergence analysis given in Chapter 18 so that it applies to convex quadratic programming problems, and identify in particular any steps that depend on Q being positive semidefinite.

24.7 Consider the quadratic programming problem given in the following form:

$$\begin{array}{ll} \text{minimize} & c^T x + \frac{1}{2} x^T Q x \\ \text{subject to} & Ax \geq b, \end{array}$$

(i.e., without assuming nonnegativity of the x vector). Show that the formulas for the step directions Δx and Δy are given by the following reduced KKT system:

$$\begin{bmatrix} -Q & A^T \\ A & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c - A^T y + Qx \\ b - Ax + \mu Y^{-1} e \end{bmatrix}. \tag{24.8}$$

Notes

The portfolio optimization model presented in Section 24.1 was first introduced by Markowitz (1959). He received the 1990 Nobel Prize in Economics for this work.

Quadratic programming is the simplest class of problems from the subject called nonlinear programming. Two excellent recent texts that cover nonlinear programming are those by Bertsekas (1995) and Nash and Sofer (1996). The first paper that extended the path-following method to quadratic programming was Monteiro and Adler (1989). The presentation given here follows Vanderbei (1999).