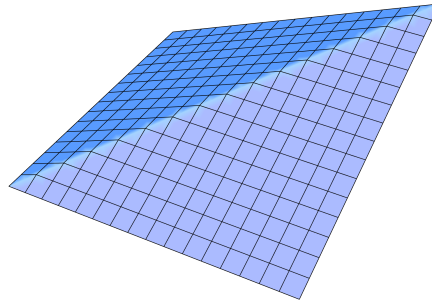




CO 372: Portfolio Optimization Models

Prof. Wei Xu • Fall 2014 • University of Waterloo
TRF 8:30–9:50AM in MC 1085

Transcribed, L^AT_EXed, and edited by Rob Zimmerman

Note: These are personal lecture notes, not official course materials. They may contain errors (which by default you should attribute to me, Rob Zimmerman, rather than the course instructor). The permanent home of these — and all of my other lecture notes — is <https://rob-zimmerman.github.io/coursenotes>.

Contents

1	Applied Linear Algebra	2
	Course Overview and Motivation	3
	Gaussian Elimination and LU Factorizations	5
	Overdetermined Systems and Linear Regression	10
	QR Factorizations	12
	Underdetermined Systems and Basic Solutions	13
	Low-Rank Updates and Block Solvers	17
	Spectral Decompositions	22

2	Computing Derivatives	23
	First Derivatives and Taylor Approximation	23
	Nonlinear Least Squares	25
	Finite Difference Approximations	25
	Automatic Differentiation	27
	Sparse Jacobians	29
3	Unconstrained Optimization and Nonlinear Systems	30
	Quadratic Models and First-Order Equations	30
	Newton's Method	33
	Convergence Rates	34
	Descent Direction Methods	36
	Trust Region Methods	38
4	Constrained Optimization	41
	Equality-Constrained Optimization	41
	Null Space Approach and Optimality Conditions	42
	Full Space and Range-Space Approaches	45
	Nonlinear Objective Function with Linear Equalities	46
	Inequality Constraints and Interior Methods	48
	The Quadratic Problem	51
	Box Constraints	54
	Penalty Functions	57
5	Portfolio Optimization Models	58
	The Efficient Frontier	58
	Practical Portfolio Constraints	61
	Capital Asset Pricing Model (CAPM)	62
	The Sharpe Ratio	65
	Appendix: Lecture and Tutorial Index	67

1. Applied Linear Algebra

Lecture 1 — Tuesday, September 09, 2014

Course Overview and Motivation

This course studies the computational optimization methodology underlying portfolio problems. The material is organized into three connected parts:

1. applied numerical linear algebra,
2. computational optimization, and
3. portfolio optimization models.

The first part develops **numerical linear algebra** tools that we will need later. The core topics include:

1. solving square linear systems,
2. handling underdetermined and overdetermined systems,
3. LU and QR factorizations, and
4. eigenvalue decomposition and singular value decomposition.

These methods provide the computational foundation for solving real optimization models efficiently and accurately.

The second part turns to **nonlinear optimization**. Since derivatives are central in nonlinear problems, we study several ways to compute them:

1. finite difference methods,
2. symbolic computation, and
3. automatic differentiation.

We then study Newton's method for nonlinear equations and optimization, together with direction search ideas and the trust region method for improving stability. Finally, we introduce constrained

optimization algorithms, with special attention to constrained quadratic problems because of their strong connection to portfolio models.

After developing linear and nonlinear solvers, we apply them to **portfolio optimization** problems in finance. A central example is the Markowitz portfolio model, which balances expected return against risk and leads naturally to quadratic programming formulations. We also study the capital asset pricing model (CAPM), which gives a theoretical expected return for a risky asset, and the Sharpe ratio, which measures excess return per unit of risk.

The portfolio portion of the course is organized around three questions:

1. How do we choose an optimal portfolio with maximum expected return or minimum risk?
2. Given the market risk, what is the appropriate expected return for an individual risky asset?
3. If two risky portfolios are available, which one is the better investment?

MATLAB is the main computational environment for the course. It is especially convenient for matrix manipulation and also provides optimization and finance functionality that we will use throughout the term.

Lecture 2 — Thursday, September 11, 2014

Suppose that we want to invest in risky assets $S_1, \dots, S_n$. The **efficient frontier** records the portfolios whose weights balance risk and return optimally.

The **capital asset pricing model** (CAPM) gives the theoretical relation for the expected return on the i th asset as

$$\mathbb{E}[R_i] = r_f + \beta_i [\mathbb{E}[R_m] - r_f],$$

where r_f is the **risk-free return**, β_i is the i th asset's **beta**, R_m is the **market return**, and $\mathbb{E}[R_m] - r_f$ is the **market risk premium**.

When the risk-free return r_f is deterministic, the **Sharpe ratio** measures excess expected return per unit of standard deviation:

$$S_a = \frac{\mathbb{E}[R_a] - r_f}{\sqrt{\text{Var}(R_a)}}.$$

For a random benchmark return R_b , the analogous quantity

$$\frac{\mathbb{E}[R_a - R_b]}{\sqrt{\text{Var}(R_a - R_b)}}$$

is usually called the **information ratio**, rather than the Sharpe ratio.

Gaussian Elimination and LU Factorizations

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be nonsingular, and let $\mathbf{b} \in \mathbb{R}^n$. We want to solve the system

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

Although the solution is formally $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, explicitly computing $\mathbf{A}^{-1}$ is computationally expensive and numerically unattractive. What else can we do?

Gaussian elimination has two parts:

1. Use elementary row operations to transform $\mathbf{A}$ into an upper triangular matrix. In the first column, for example, set $m_{i1} = a_{i1}/a_{11}$ for $i \geq 2$ and replace row i by row i minus m_{i1} times row 1.
2. Solve the resulting upper triangular system by back substitution.

Example 2.1 Consider

$$\begin{cases} x_1 + 2x_2 - 3x_3 = 3, \\ 2x_1 - 3x_2 + x_3 = 1, \\ -x_1 + 8x_2 + 2x_3 = 0. \end{cases}$$

The first column multipliers are $m_{21} = 2$ and $m_{31} = -1$. After eliminating the first column and then the second column, we obtain

$$\begin{cases} x_1 + 2x_2 - 3x_3 = 3, \\ -7x_2 + 7x_3 = -5, \\ 9x_3 = -29/7. \end{cases}$$

Back substitution gives

$$x_3 = -\frac{29}{63}, \quad x_2 = \frac{16}{63}, \quad \text{and} \quad x_1 = \frac{10}{9}.$$

The pivot a_{11} cannot be zero. If it is zero, or even very small, we interchange rows before eliminating. More generally, **partial pivoting** chooses a pivot with large absolute value, usually the largest available entry in the current column, and moves it into position by interchanging rows.

Example 2.2 If we wish to solve

$$\begin{cases} 0x_1 + 3x_2 + 2x_3 = 1 \\ 3x_1 + 3x_2 + x_3 = 0 \\ x_1 + 2x_2 + 3x_3 = 0 \end{cases},$$

we can start by interchanging the first two rows to obtain the equivalent system

$$\begin{cases} 3x_1 + 3x_2 + x_3 = 0 \\ 0x_1 + 3x_2 + 2x_3 = 1 \\ x_1 + 2x_2 + 3x_3 = 0 \end{cases}$$

and solve as usual. This row interchange creates a nonzero first pivot, after which Gaussian elimination proceeds as usual.

In MATLAB, the preferred command is $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$, which performs Gaussian elimination. The command $\mathbf{x} = \text{inv}(\mathbf{A}) * \mathbf{b}$ is mathematically meaningful, but it should not be used for numerical linear solves because it explicitly forms the inverse.

Gaussian elimination can be written in matrix form as an LU factorization. In its simplest form,

$$\mathbf{A} = \mathbf{L}\mathbf{U},$$

where $\mathbf{L}$ is unit lower triangular and $\mathbf{U}$ is upper triangular. To solve $\mathbf{A}\mathbf{x} = \mathbf{b}$, factor $\mathbf{A} = \mathbf{L}\mathbf{U}$, solve $\mathbf{L}\mathbf{y} = \mathbf{b}$, and then solve $\mathbf{U}\mathbf{x} = \mathbf{y}$.

The computational sequence is shown in [Figure 1](#). The factorization is the expensive step, but once it is available, each new right-hand side requires only two triangular solves.

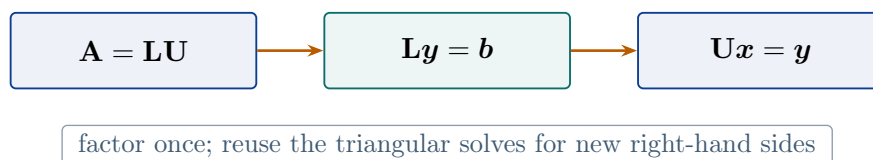


FIGURE 1

Lecture 3 — Friday, September 12, 2014

Example 3.1 For another pivoting example, let $\mathbf{A} = \begin{pmatrix} 1 & 1 & 3 \\ 2 & 1 & 2 \\ 1 & -1 & -1 \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix}$, and solve $\mathbf{A}\mathbf{x} = \mathbf{b}$.

$$\begin{cases} x_1 + x_2 + 3x_3 = 1 \\ 2x_1 + x_2 + 2x_3 = 0 \\ x_1 - x_2 - x_3 = 5 \end{cases} \xrightarrow{\text{pivot}} \begin{cases} 2x_1 + x_2 + 2x_3 = 0 \\ x_1 + x_2 + 3x_3 = 1 \\ x_1 - x_2 - x_3 = 5 \end{cases} \xrightarrow[m_{31}=1/2]{m_{21}=1/2} \begin{cases} 2x_1 + x_2 + 2x_3 = 0 \\ 0 + \frac{1}{2}x_2 + 2x_3 = 1 \\ 0 - \frac{3}{2}x_2 - 2x_3 = 5 \end{cases}$$

Continuing the elimination gives $x_3 = 2$, $x_2 = -6$, and $x_1 = 1$.

When solving a system using Gaussian elimination, the cost of elimination is $(2/3)n^3 + O(n^2)$ flops, while backward substitution costs $O(n^2)$ flops. The total cost is therefore $(2/3)n^3 + O(n^2)$ flops, which is $O(n^3)$.

In MATLAB, once $\mathbf{A}$ and $\mathbf{b}$ have been entered, the command $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$ solves the system. MATLAB warns when $\mathbf{A}$ is singular, nearly singular, or badly scaled, but it ordinarily still returns a result. Near-singularity is assessed through conditioning, not by comparing $\det(\mathbf{A})$ with a fixed threshold; the determinant is scale-dependent and therefore unsuitable for that test.

Another method is **LU factorization**, which is mathematically equivalent to Gaussian elimination:

1. Factor $\mathbf{PA} = \mathbf{LU}$, where $\mathbf{L}$ is unit lower triangular, $\mathbf{U}$ is upper triangular, and $\mathbf{P}$ is a permutation matrix.

2. Solve $\mathbf{L}\mathbf{y} = \mathbf{P}\mathbf{b}$.
3. Solve $\mathbf{U}\mathbf{x} = \mathbf{y}$.

In MATLAB, this is written as $[\mathbf{L}, \mathbf{U}, \mathbf{P}] = \text{lu}(\mathbf{A})$, $\mathbf{y} = \mathbf{L} \backslash (\mathbf{P} * \mathbf{b})$, and $\mathbf{x} = \mathbf{U} \backslash \mathbf{y}$.

This approach stores $\mathbf{L}$, $\mathbf{U}$, and $\mathbf{P}$ explicitly, whereas $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$ does not. The factorization step costs $O(n^3)$, but once the factors are available, each new right-hand side can be solved using only the two triangular solves, each costing $O(n^2)$.

We still want to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$. A matrix $\mathbf{A}$ is symmetric if $\mathbf{A} = \mathbf{A}^\top$, and it is positive definite if $\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0$ for all nonzero $\mathbf{x}$. Equivalently, a real symmetric matrix is positive definite if all of its eigenvalues are positive. If $\mathbf{A}$ is symmetric positive definite, then there is a nonsingular lower triangular matrix $\mathbf{L}$ such that

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\top.$$

This is the **Cholesky factorization**.

Lecture 4 — Tuesday, September 16, 2014

For $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$, we have now seen several ways to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$:

1. Gaussian elimination with partial pivoting.
2. MATLAB's linear solve command $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$.
3. LU factorization: factor $\mathbf{P}\mathbf{A} = \mathbf{L}\mathbf{U}$, solve $\mathbf{L}\mathbf{y} = \mathbf{P}\mathbf{b}$, and solve $\mathbf{U}\mathbf{x} = \mathbf{y}$.

If $\mathbf{A}$ is known ahead of time to be symmetric positive definite, then the Cholesky factorization gives a more efficient method:

1. Factor $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$.
2. Solve $\mathbf{L}\mathbf{y} = \mathbf{b}$.
3. Solve $\mathbf{L}^\top \mathbf{x} = \mathbf{y}$.

The Cholesky factorization costs $(1/3)n^3 + O(n^2)$ flops, about half the leading-order cost of LU factorization.

In MATLAB, $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$ detects symmetric positive definite structure when possible and uses Cholesky factorization. Alternatively, we may write $\mathbf{R} = \text{chol}(\mathbf{A})$, $\mathbf{y} = \mathbf{R}' \backslash \mathbf{b}$, and $\mathbf{x} = \mathbf{R} \backslash \mathbf{y}$; MATLAB's default factor is upper triangular, so $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$. If $\mathbf{A}$ is not positive definite, $\text{chol}(\mathbf{A})$ fails, although numerical roundoff means that its behaviour on positive semidefinite matrices can vary. The two-output form $[\mathbf{R}, \text{flag}] = \text{chol}(\mathbf{A})$ reports failure through **flag** rather than raising an error, making it a fast test for positive definiteness.

Example 4.1 Let

$$\mathbf{A} = \begin{pmatrix} 4 & 2 & -2 \\ 2 & 2 & -3 \\ -2 & -3 & 14 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

To solve $\mathbf{A}\mathbf{x} = \mathbf{b}$ by Cholesky, first find $\mathbf{L}$ such that $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$. Here

$$\mathbf{L} = \begin{pmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{pmatrix} \quad \text{and} \quad \mathbf{L}\mathbf{L}^\top = \begin{pmatrix} \ell_{11}^2 & \ell_{11}\ell_{21} & \ell_{11}\ell_{31} \\ \ell_{11}\ell_{21} & \ell_{21}^2 + \ell_{22}^2 & \ell_{21}\ell_{31} + \ell_{22}\ell_{32} \\ \ell_{11}\ell_{31} & \ell_{21}\ell_{31} + \ell_{22}\ell_{32} & \ell_{31}^2 + \ell_{32}^2 + \ell_{33}^2 \end{pmatrix}.$$

Matching entries with $\mathbf{A}$ and using positive diagonal entries for Cholesky factors,

$$\ell_{11}^2 = 4 \Rightarrow \ell_{11} = 2, \quad \ell_{11}\ell_{21} = 2 \Rightarrow \ell_{21} = 1, \quad \text{and} \quad \ell_{11}\ell_{31} = -2 \Rightarrow \ell_{31} = -1,$$

$$\ell_{21}^2 + \ell_{22}^2 = 2 \Rightarrow 1 + \ell_{22}^2 = 2 \Rightarrow \ell_{22} = 1,$$

$$\text{and} \quad \ell_{21}\ell_{31} + \ell_{22}\ell_{32} = -3 \Rightarrow -1 + \ell_{32} = -3 \Rightarrow \ell_{32} = -2,$$

$$\ell_{31}^2 + \ell_{32}^2 + \ell_{33}^2 = 14 \Rightarrow 1 + 4 + \ell_{33}^2 = 14 \Rightarrow \ell_{33} = 3.$$

Therefore,

$$\mathbf{L} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -2 & 3 \end{pmatrix}.$$

Solving $\mathbf{L}\mathbf{y} = \mathbf{b}$ gives $\mathbf{y} = \begin{pmatrix} 1/2 \\ -1/2 \\ 1/6 \end{pmatrix}$, and solving $\mathbf{L}^\top \mathbf{x} = \mathbf{y}$ gives

$$\mathbf{x} = \begin{pmatrix} 17/36 \\ -7/18 \\ 1/18 \end{pmatrix}.$$

Overdetermined Systems and Linear Regression

Given data $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, suppose that we want to fit a line $y = a_1x + a_0$. The equations $a_1x_i + a_0 = y_i$ can be written as

$$\begin{pmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix},$$

or simply as $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{n \times 2}$ and $\mathbf{b} \in \mathbb{R}^n$. Usually this system is overdetermined, so we seek $\mathbf{x}$ such that $\mathbf{Ax}$ is as close to $\mathbf{b}$ as possible. Suppose that $\mathbf{x}_1$ and $\mathbf{x}_2$ are two solution “candidates.” To compare residuals such as $\mathbf{Ax}_1 - \mathbf{b}$ and $\mathbf{Ax}_2 - \mathbf{b}$, we need a norm. Define the **2-norm**, the **1-norm**, and the **∞ -norm** as

$$\|\mathbf{y}\|_2 := \sqrt{y_1^2 + \dots + y_n^2}, \quad \|\mathbf{y}\|_1 := |y_1| + \dots + |y_n|, \quad \text{and} \quad \|\mathbf{y}\|_\infty := \max\{|y_1|, \dots, |y_n|\},$$

respectively. We minimize the squared 2-norm of the residual vector (or rather half of it, purely for mathematical convenience). Assuming $\mathbf{A}$ has full column rank, we obtain

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \iff \min_{\mathbf{x}} \frac{1}{2} (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b})$$

and hence, after dropping the constant term, the equivalent quadratic objective is

$$q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}.$$

The first-order condition is then

$$\nabla q(\mathbf{x}) = \mathbf{A}^\top \mathbf{Ax} - \mathbf{A}^\top \mathbf{b} = \mathbf{0}_n.$$

Lecture 5 — Thursday, September 18, 2014

Recall that for a symmetric positive definite system $\mathbf{Ax} = \mathbf{b}$, the Cholesky method consists of factoring $\mathbf{A} = \mathbf{LL}^\top$, solving $\mathbf{Ly} = \mathbf{b}$, and solving $\mathbf{L}^\top \mathbf{x} = \mathbf{y}$.

Now let $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m > n$. The least-squares problem is

$$\mathbf{Ax} \approx \mathbf{b} \iff \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2.$$

With $q(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} - (\mathbf{A}^\top \mathbf{b})^\top \mathbf{x}$, the optimality condition is $\nabla q(\mathbf{x}) = \mathbf{0}_n$.

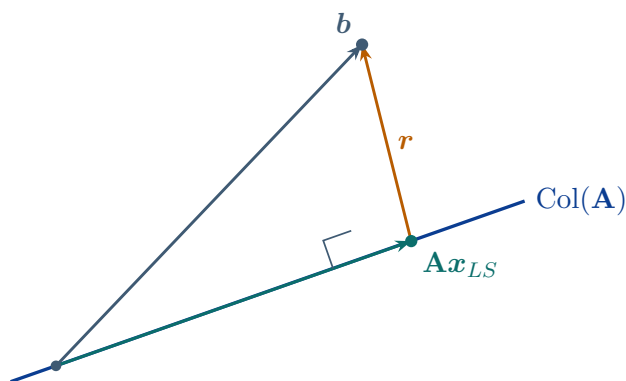
Thus

$$\mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{b}.$$

If $\text{rank}(\mathbf{A}) = n$, then $\mathbf{A}^\top \mathbf{A}$ is symmetric positive definite and the **least-squares solution** is

$$\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}.$$

Geometrically, $\mathbf{Ax}_{LS}$ is the orthogonal projection of $\mathbf{b}$ onto the column space of $\mathbf{A}$. Thus the residual $\mathbf{r} = \mathbf{b} - \mathbf{Ax}_{LS}$ is orthogonal to that space, exactly as Figure 2 illustrates; the normal equations are simply $\mathbf{A}^\top \mathbf{r} = \mathbf{0}$.



least squares makes the residual orthogonal to every column of $\mathbf{A}$

FIGURE 2

The **method of normal equations** is therefore:

1. Compute $\mathbf{M} = \mathbf{A}^\top \mathbf{A}$ and $\tilde{\mathbf{b}} = \mathbf{A}^\top \mathbf{b}$.
2. Compute the Cholesky factorization $\mathbf{M} = \mathbf{LL}^\top$.
3. Solve $\mathbf{Ly} = \tilde{\mathbf{b}}$.
4. Solve $\mathbf{L}^\top \mathbf{x} = \mathbf{y}$.

QR Factorizations

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$. A **QR factorization** has the form

$$\mathbf{A} = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{0}_{(m-n) \times n} \end{pmatrix},$$

where $\mathbf{Q}$ is an $m \times m$ orthogonal matrix, $\mathbf{R}$ is an $n \times n$ upper triangular matrix, and $\mathbf{0}_{(m-n) \times n}$ is an $(m-n) \times n$ zero matrix. If $\text{rank}(\mathbf{A}) = n$, then the diagonal entries of $\mathbf{R}$ are nonzero.

If we split $\mathbf{Q}$ into $[\mathbf{Y}, \mathbf{Z}]$ where $\mathbf{Y}$ consists of the first n columns of $\mathbf{Q}$, we have

$$\mathbf{A} = [\mathbf{Y}, \mathbf{Z}] \begin{pmatrix} \mathbf{R} \\ \mathbf{0}_{(m-n) \times n} \end{pmatrix} = \mathbf{Y}\mathbf{R}.$$

This is the **compact QR factorization**, which is much cheaper to compute than the full form. Here $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}_n$, $\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}_{m-n}$, and $\mathbf{Y}^\top \mathbf{Z} = \mathbf{0}_{n \times (m-n)}$. Thus the columns of $\mathbf{Y}$ are orthonormal.

In MATLAB, `[Q,R] = qr(A)` computes the full form, while `[Y,R] = qr(A,0)` computes the compact form.

Since $\mathbf{A} = \mathbf{Y}\mathbf{R}$, the normal equations become

$$\mathbf{R}^\top \mathbf{Y}^\top \mathbf{Y} \mathbf{R} \mathbf{x} = \mathbf{R}^\top \mathbf{Y}^\top \mathbf{b}.$$

Using $\mathbf{Y}^\top \mathbf{Y} = \mathbf{I}$ and the nonsingularity of $\mathbf{R}$, this reduces to

$$\mathbf{R} \mathbf{x} = \mathbf{Y}^\top \mathbf{b}.$$

Thus the QR method for solving $\mathbf{A} \mathbf{x} \approx \mathbf{b}$ is:

1. Compute $\mathbf{A} = \mathbf{Y}\mathbf{R}$.
2. Compute $\tilde{\mathbf{b}} = \mathbf{Y}^\top \mathbf{b}$.
3. Solve $\mathbf{R} \mathbf{x} = \tilde{\mathbf{b}}$.

In MATLAB, we may write `[Y,R] = qr(A,0)`, `bbar = Y'*b`, and `x = R\bbar`. The command `x = A\b` calls an appropriate solver automatically.

The QR method is generally more numerically stable than the method of normal equations because forming $\mathbf{A}^\top \mathbf{A}$ squares the condition number. How many reliable digits this saves depends on the problem; in a poorly conditioned example, normal equations might give about 7 reliable digits while QR gives 12 or 13.

Lecture 6 — Friday, September 19, 2014

For overdetermined systems, the normal equations solve $\mathbf{A}^\top \mathbf{A} \mathbf{x} = \mathbf{A}^\top \mathbf{b}$, while the QR approach writes $\mathbf{A} = \mathbf{Y} \mathbf{R}$ and solves $\mathbf{R} \mathbf{x} = \mathbf{Y}^\top \mathbf{b}$.

Underdetermined Systems and Basic Solutions

Now consider an underdetermined system $\mathbf{A} \mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $m < n$. Such systems often arise in portfolio optimization. For example, suppose that stocks $S_1, \dots, S_n$ have returns $\mu_1, \dots, \mu_n$, and suppose that we want a portfolio with return 5%. We may then minimize an objective $F(\mathbf{x})$ subject to

$$\sum_{i=1}^n x_i \mu_i = 5\%,$$

and

$$\mathbf{x}^\top \mathbf{1} = 1.$$

There may be infinitely many feasible solutions. We can describe all of them once we know one particular solution and the null space of $\mathbf{A}$. If $\mathbf{x}_0$ is a solution, then $\mathbf{A} \mathbf{x}_0 = \mathbf{b}$. For any $\mathbf{y} \in \text{Null}(\mathbf{A})$,

$$\mathbf{A}(\mathbf{x}_0 + \mathbf{y}) = \mathbf{A} \mathbf{x}_0 + \mathbf{A} \mathbf{y} = \mathbf{A} \mathbf{x}_0 + \mathbf{0}_m = \mathbf{b}.$$

Thus every vector of the form $\mathbf{x}_0 + \mathbf{y}$, with $\mathbf{y} \in \text{Null}(\mathbf{A})$, is also a solution. The problem is therefore to find a particular solution $\mathbf{x}_0$ and a basis for the null space.

One way to find such a particular solution is to use a QR factorization of $\mathbf{A}^\top$:

$$\mathbf{A}^\top = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{0}_{(n-m) \times m} \end{pmatrix} = (\mathbf{Y} \quad \mathbf{Z}) \begin{pmatrix} \mathbf{R} \\ \mathbf{0}_{(n-m) \times m} \end{pmatrix} = \mathbf{Y} \mathbf{R} \implies \mathbf{A} = \mathbf{R}^\top \mathbf{Y}^\top.$$

Consider the following construction:

1. Solve $\mathbf{R}^\top \mathbf{v} = \mathbf{b}$.
2. Set $\mathbf{x}_0 = \mathbf{Y}\mathbf{v}$.

Then

$$\mathbf{A}\mathbf{x}_0 = \mathbf{A}\mathbf{Y}(\mathbf{R}^\top)^{-1}\mathbf{b} = \mathbf{R}^\top \mathbf{Y}^\top \mathbf{Y}(\mathbf{R}^\top)^{-1}\mathbf{b} = \mathbf{b}.$$

It follows that $\mathbf{x}_0$ is the minimum-2-norm solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$. Indeed, $\mathbf{x}_0 \in \text{Range}(\mathbf{A}^\top) = \text{Null}(\mathbf{A})^\perp$. Every solution has the form $\mathbf{x}_0 + \mathbf{Z}\mathbf{w}$, and therefore

$$\|\mathbf{x}_0 + \mathbf{Z}\mathbf{w}\|_2^2 = \|\mathbf{x}_0\|_2^2 + \|\mathbf{Z}\mathbf{w}\|_2^2 \geq \|\mathbf{x}_0\|_2^2.$$

In Figure 3, the feasible set is the translate $\mathbf{x}_0 + \text{Null}(\mathbf{A})$. The QR construction chooses the point where the line from the origin meets that affine space orthogonally, which explains why $\mathbf{x}_0$ has minimum norm.

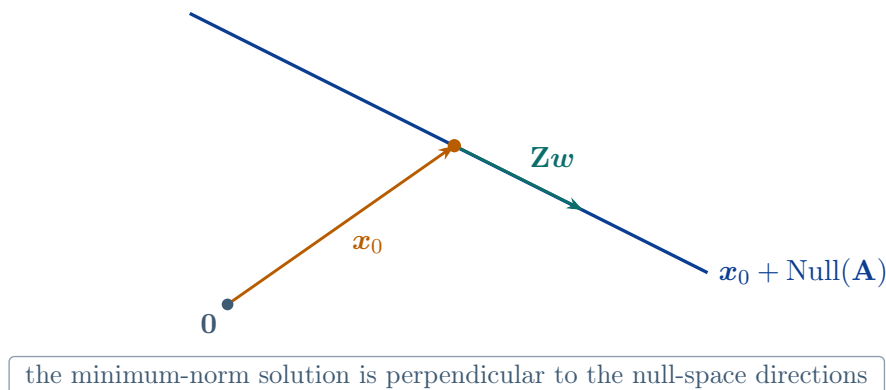


FIGURE 3

For the null space, we have

$$\mathbf{A}\mathbf{Z} = \mathbf{R}^\top \mathbf{Y}^\top \mathbf{Z} = \mathbf{R}^\top \mathbf{0}_{m \times (n-m)} = \mathbf{0}_{m \times (n-m)},$$

so the columns of $\mathbf{Z}$ form a basis for $\text{Null}(\mathbf{A})$. Since $\text{rank}(\mathbf{Z}) = n - m$, every vector in $\text{Null}(\mathbf{A})$ can be written as $\mathbf{Z}\mathbf{w}$ for some $\mathbf{w} \in \mathbb{R}^{n-m}$.

In summary:

1. Compute $\mathbf{A}^\top = \mathbf{Q} \begin{pmatrix} \mathbf{R} \\ \mathbf{0}_{(n-m) \times m} \end{pmatrix} = (\mathbf{Y} \quad \mathbf{Z}) \begin{pmatrix} \mathbf{R} \\ \mathbf{0}_{(n-m) \times m} \end{pmatrix}$.
2. Solve $\mathbf{R}^\top \mathbf{v} = \mathbf{b}$.

3. Set $\mathbf{x}_0 = \mathbf{Y}\mathbf{v}$.
4. Write the general solution as $\mathbf{x} = \mathbf{x}_0 + \mathbf{Z}\mathbf{w}$.

Another way to obtain a particular solution is to partition $\mathbf{A} = [\mathbf{A}_1, \mathbf{A}_2]$ and assume first that $\mathbf{A}_1$ is nonsingular. Then a particular solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is obtained as follows:

1. Solve $\mathbf{A}_1\mathbf{x}_1 = \mathbf{b}$, for instance by LU factorization.
2. Set $\mathbf{x}_0 = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0}_{n-m} \end{pmatrix}$.

Construct a basis of $\text{Null}(\mathbf{A})$ by setting

$$\mathbf{Z} = \begin{pmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I}_{n-m} \end{pmatrix}.$$

Indeed,

$$\mathbf{AZ} = (\mathbf{A}_1, \mathbf{A}_2) \begin{pmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I}_{n-m} \end{pmatrix} = -\mathbf{A}_2 + \mathbf{A}_2 = \mathbf{0}_{m \times (n-m)}.$$

Also, $\text{rank}(\mathbf{Z}) = n - m$.

Lecture 7 — Tuesday, September 23, 2014

We are trying to solve the underdetermined system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $m < n$. Our current approach writes $\mathbf{A} = (\mathbf{A}_1, \mathbf{A}_2)$ and does the following:

1. Solve $\mathbf{A}_1\mathbf{x}_1 = \mathbf{b}$.
2. Set $\mathbf{x}_0 = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0}_{n-m} \end{pmatrix}$, $\mathbf{Z} = \begin{pmatrix} -\mathbf{A}_1^{-1}\mathbf{A}_2 \\ \mathbf{I}_{n-m} \end{pmatrix}$.

If $\mathbf{A}_1$ is singular, we can introduce a permutation matrix $\mathbf{P}$ such that

$$\mathbf{AP}^\top = (\tilde{\mathbf{A}}_1, \tilde{\mathbf{A}}_2),$$

where $\tilde{\mathbf{A}}_1$ is nonsingular. One way to find $\mathbf{P}$ is to factor $\mathbf{A}^\top$:

$$\mathbf{P}\mathbf{A}^\top = \mathbf{L}\mathbf{U} = \begin{pmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{pmatrix} \mathbf{U} \implies \mathbf{A}\mathbf{P}^\top = \mathbf{U}^\top(\mathbf{L}_1^\top, \mathbf{L}_2^\top) = (\mathbf{U}^\top \mathbf{L}_1^\top, \mathbf{U}^\top \mathbf{L}_2^\top).$$

1. Solve $\mathbf{A}\mathbf{P}^\top \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0}_{n-m} \end{pmatrix} = \mathbf{b}$. Since $\mathbf{U}^\top \mathbf{L}_1^\top \mathbf{x}_1 = \mathbf{b}$, solve $\mathbf{U}^\top \mathbf{y} = \mathbf{b}$ and $\mathbf{L}_1^\top \mathbf{x}_1 = \mathbf{y}$.
2. $\mathbf{x}_0 = \mathbf{P}^\top \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0}_{n-m} \end{pmatrix}$.
3. Set $\mathbf{Z} = \mathbf{P}^\top \begin{pmatrix} -\mathbf{L}_1^{-\top} \mathbf{L}_2^\top \\ \mathbf{I}_{n-m} \end{pmatrix}$, so $\text{rank}(\mathbf{Z}) = n - m$.

This choice of $\mathbf{Z}$ satisfies

$$\mathbf{A}\mathbf{Z} = \mathbf{A}\mathbf{P}^\top \begin{pmatrix} -\mathbf{L}_1^{-\top} \mathbf{L}_2^\top \\ \mathbf{I}_{n-m} \end{pmatrix} = \mathbf{U}^\top(\mathbf{L}_1^\top, \mathbf{L}_2^\top) \begin{pmatrix} -\mathbf{L}_1^{-\top} \mathbf{L}_2^\top \\ \mathbf{I}_{n-m} \end{pmatrix} = \mathbf{U}^\top(-\mathbf{L}_2^\top + \mathbf{L}_2^\top) = \mathbf{0}_{m \times (n-m)}.$$

In summary, the **basic solution method** for $\mathbf{A}\mathbf{x} = \mathbf{b}$ produces a solution as follows:

1. LU factor $\mathbf{A}^\top$ as $\mathbf{P}\mathbf{A}^\top = \mathbf{L}\mathbf{U} = \begin{pmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{pmatrix} \mathbf{U}$.
2. Solve $\mathbf{U}^\top \mathbf{y} = \mathbf{b}$ and $\mathbf{L}_1^\top \mathbf{x}_1 = \mathbf{y}$.
3. $\mathbf{x}_0 = \mathbf{P}^\top \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0}_{n-m} \end{pmatrix}$.
4. Construct $\mathbf{Z} = \mathbf{P}^\top \begin{pmatrix} -\mathbf{L}_1^{-\top} \mathbf{L}_2^\top \\ \mathbf{I}_{n-m} \end{pmatrix}$.
5. Construct any feasible point $\mathbf{x}_0 + \mathbf{Z}\mathbf{w}$, for any $\mathbf{w} \in \mathbb{R}^{n-m}$.

Lecture 8 — Thursday, September 25, 2014

Low-Rank Updates and Block Solvers

For a constrained problem

$$\min f(\mathbf{x}) \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b},$$

we can substitute the general feasible form $\mathbf{x} = \mathbf{x}_0 + \mathbf{Z}\mathbf{w}$ and solve the unconstrained problem

$$\min_{\mathbf{w}} f(\mathbf{x}_0 + \mathbf{Z}\mathbf{w}).$$

Low-rank updates arise when a factorization is already known and the matrix changes by a low-rank matrix. For example, given a QR or Cholesky factorization of $\mathbf{A}$, we may seek a factorization of $\mathbf{A} + \mathbf{B}$ efficiently when $\text{rank}(\mathbf{B})$ is small. The official notes emphasize that such updates can reduce the dominant work from about $O(n^3)$ to about $O(n^2)$ in favourable cases.

Suppose that $\mathbf{H} = \mathbf{W} + \mathbf{V}\mathbf{V}^\top$, where $\mathbf{V} \in \mathbb{R}^{n \times t}$ with $t \ll n$, and suppose that $\mathbf{W}$ is symmetric positive definite and systems involving it can be solved quickly. Then $\mathbf{H}$ is also positive definite, and the system $\mathbf{H}\mathbf{x} = \mathbf{r}$ is equivalent to

$$\begin{pmatrix} \mathbf{W} & \mathbf{V} \\ \mathbf{V}^\top & -\mathbf{I}_t \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{0}_t \end{pmatrix}$$

because this block system gives

$$\begin{aligned} \mathbf{W}\mathbf{x} + \mathbf{V}\mathbf{z} &= \mathbf{r}, \\ \mathbf{V}^\top \mathbf{x} - \mathbf{z} &= \mathbf{0}_t. \end{aligned}$$

Multiplying the first equation by $-\mathbf{V}^\top \mathbf{W}^{-1}$ gives

$$-\mathbf{V}^\top \mathbf{W}^{-1}(\mathbf{W}\mathbf{x} + \mathbf{V}\mathbf{z}) = -\mathbf{V}^\top \mathbf{W}^{-1}\mathbf{r}$$

and hence

$$-\mathbf{V}^\top \mathbf{x} - \mathbf{V}^\top \mathbf{W}^{-1}\mathbf{V}\mathbf{z} = -\mathbf{V}^\top \mathbf{W}^{-1}\mathbf{r}$$

Lecture 9 — Friday, September 26, 2014

For an underdetermined system $\mathbf{A}\mathbf{x} = \mathbf{b}$, with $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $m < n$, and $\text{rank}(\mathbf{A}) = m$, the basic solution construction can be summarized as follows:

1. LU factorize $\mathbf{A}^\top$: $\mathbf{P}\mathbf{A}^\top = \mathbf{L}\mathbf{U} = \begin{pmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{pmatrix} \mathbf{U}$.
2. Solve $\mathbf{U}^\top \mathbf{y} = \mathbf{b}$ and $\mathbf{L}_1^\top \mathbf{x}_1 = \mathbf{y}$.
3. $\mathbf{x}_0 = \mathbf{P}^\top \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{0}_{n-m} \end{pmatrix}$ and $\mathbf{Z} = \mathbf{P}^\top \begin{pmatrix} -\mathbf{L}_1^{-\top} \mathbf{L}_2^\top \\ \mathbf{I}_{n-m} \end{pmatrix}$.

For the block solver, suppose $\mathbf{H} = \mathbf{W} + \mathbf{V}\mathbf{V}^\top$, where $\mathbf{V} \in \mathbb{R}^{n \times t}$ with $t \ll n$, and $\mathbf{W} \in \mathbb{R}^{n \times n}$ is symmetric positive definite. Then

$$\mathbf{H}\mathbf{x} = \mathbf{r} \iff \begin{pmatrix} \mathbf{W} & \mathbf{V} \\ \mathbf{V}^\top & -\mathbf{I}_t \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{0}_t \end{pmatrix}.$$

Eliminating $\mathbf{x}$ gives

$$\begin{pmatrix} \mathbf{W} & \mathbf{V} \\ \mathbf{0}_{t \times n} & -\mathbf{I}_t - \mathbf{V}^\top \mathbf{W}^{-1} \mathbf{V} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ -\mathbf{V}^\top \mathbf{W}^{-1} \mathbf{r} \end{pmatrix}.$$

1. Solve $(\mathbf{I}_t + \mathbf{V}^\top \mathbf{W}^{-1} \mathbf{V})\mathbf{z} = \mathbf{V}^\top \mathbf{W}^{-1} \mathbf{r}$, which is a $t \times t$ system.
2. Solve $\mathbf{W}\mathbf{x} = \mathbf{r} - \mathbf{V}\mathbf{z}$.

When solves with $\mathbf{W}$ are fast, the reduced solve costs $O(t^3)$, while forming and applying the dense low-rank terms typically costs $O(nt^2)$ in addition to the required solves with $\mathbf{W}$. For fixed t , the work can therefore scale linearly with n rather than cubically.

A special case is $\mathbf{V} = \mathbf{v} \in \mathbb{R}^n$ and $\mathbf{W} = \mathbf{T}$, where $\mathbf{T}$ is symmetric positive definite and tridiagonal. Then $\mathbf{H}\mathbf{x} = (\mathbf{v}\mathbf{v}^\top + \mathbf{T})\mathbf{x} = \mathbf{r}$, and the computation is as follows:

1. Solve $\mathbf{T}\mathbf{w} = \mathbf{v}$, and compute $s = \mathbf{v}^\top \mathbf{w}$ and $p = \mathbf{r}^\top \mathbf{w}$.
2. Compute $z = p/(1 + s)$.

3. Solve $\mathbf{T}\mathbf{x} = \mathbf{r} - \mathbf{v}z$.

Example 9.1 Let

$$\mathbf{T} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{r} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Then $\mathbf{H} = \mathbf{T} + \mathbf{v}\mathbf{v}^\top$ is

$$\mathbf{H} = \mathbf{T} + \mathbf{v}\mathbf{v}^\top = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}.$$

Solving $\mathbf{T}\mathbf{w} = \mathbf{v}$ gives $\mathbf{w} = \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix}$, so $s = \mathbf{v}^\top \mathbf{w} = 2/3$ and $p = \mathbf{r}^\top \mathbf{w} = 1/3$. Hence $z = p/(1 + s) = 1/5$. Finally, solving

$$\mathbf{T}\mathbf{x} = \mathbf{r} - \mathbf{v}z = \begin{pmatrix} 4/5 \\ -1/5 \end{pmatrix}$$

gives

$$\mathbf{x} = \begin{pmatrix} 3/5 \\ -2/5 \end{pmatrix}.$$

Example 9.2 Let

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}, \quad \mathbf{v} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{r} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

We want to solve

$$(\mathbf{T} + \mathbf{v}\mathbf{v}^\top)\mathbf{x} = \mathbf{r}$$

using the block solver steps above.

First solve $\mathbf{T}\mathbf{w} = \mathbf{v}$:

$$\mathbf{w} = \begin{pmatrix} 1 \\ 1/2 \\ 1/3 \\ 1/4 \end{pmatrix}.$$

Then

$$s = \mathbf{v}^\top \mathbf{w} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = \frac{25}{12} \quad \text{and} \quad p = \mathbf{r}^\top \mathbf{w} = 1.$$

Hence

$$z = \frac{p}{1+s} = \frac{1}{1+25/12} = \frac{12}{37}.$$

Now solve

$$\mathbf{T}\mathbf{x} = \mathbf{r} - \mathbf{v}z = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \frac{12}{37} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 25/37 \\ -12/37 \\ -12/37 \\ -12/37 \end{pmatrix}.$$

Since $\mathbf{T}$ is diagonal,

$$\mathbf{x} = \begin{pmatrix} 25/37 \\ -6/37 \\ -4/37 \\ -3/37 \end{pmatrix}$$

is our solution.

Now consider the constrained block system

$$\begin{pmatrix} \mathbf{H} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0}_{m \times m} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{b} \end{pmatrix},$$

where $\mathbf{H} = \mathbf{W} + \mathbf{V}\mathbf{V}^\top$ and $\mathbf{A} \in \mathbb{R}^{m \times n}$. Define $\mathbf{z} = \mathbf{V}^\top \mathbf{x}$.

$$\begin{pmatrix} \mathbf{W} & \mathbf{V} & \mathbf{A}^\top \\ \mathbf{V}^\top & -\mathbf{I}_t & \mathbf{0}_{t \times m} \\ \mathbf{A} & \mathbf{0}_{m \times t} & \mathbf{0}_{m \times m} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{0}_t \\ \mathbf{b} \end{pmatrix}$$

is equivalent, after eliminating through $\mathbf{W}$, to

$$\begin{pmatrix} \mathbf{W} & \mathbf{V} & \mathbf{A}^\top \\ \mathbf{0}_{t \times n} & -\mathbf{I}_t - \mathbf{V}^\top \mathbf{W}^{-1} \mathbf{V} & -\mathbf{V}^\top \mathbf{W}^{-1} \mathbf{A}^\top \\ \mathbf{0}_{m \times n} & -\mathbf{A} \mathbf{W}^{-1} \mathbf{V} & -\mathbf{A} \mathbf{W}^{-1} \mathbf{A}^\top \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{z} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ -\mathbf{V}^\top \mathbf{W}^{-1} \mathbf{r} \\ \mathbf{b} - \mathbf{A} \mathbf{W}^{-1} \mathbf{r} \end{pmatrix}.$$

So for this setup, we perform the following steps:

1. Solve $\bar{\mathbf{W}} \begin{pmatrix} \mathbf{z} \\ \mathbf{y} \end{pmatrix} = \mathbf{d}$.
2. Solve $\mathbf{W}\mathbf{x} = \mathbf{r} - \mathbf{V}\mathbf{z} - \mathbf{A}^\top \mathbf{y}$.

If applying $\mathbf{W}^{-1}$ to $m + t$ right-hand sides costs $O(n(m + t))$, then forming the dense reduced blocks typically costs $O(n(m + t)^2)$ and factoring them costs $O((m + t)^3)$. The resulting total is therefore of order

$$O(n(m + t)^2 + (m + t)^3),$$

apart from structure-dependent constants, compared with $O((m + n)^3)$ for a direct dense solve. For example, if

$$m = 6, \quad t = 4, \quad \text{and} \quad n = 100.$$

then $m + t = 10$, while a direct method would involve size $m + n = 106$. Comparing only the cubic terms, the block method requires roughly three orders of magnitude less work than a dense Gaussian-elimination solve; the actual speedup also depends on the matrix structure and implementation.

Lecture 10 — Tuesday, September 30, 2014

Suppose $\mathbf{W}$ is symmetric positive definite and $\mathbf{W}\mathbf{s} = \mathbf{p}$ has a fast linear solver. To solve $\mathbf{H}\mathbf{x} = \mathbf{r}$, where $\mathbf{H} = \mathbf{W} + \mathbf{v}\mathbf{v}^\top$, use the block form

$$\begin{pmatrix} \mathbf{W} & \mathbf{v} \\ \mathbf{v}^\top & -\mathbf{I}_1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ z \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{0}_1 \end{pmatrix}.$$

After elimination this becomes

$$\begin{pmatrix} \mathbf{W} & \mathbf{v} \\ \mathbf{0}_{1 \times n} & \mathbf{I}_1 + \mathbf{v}^\top \mathbf{W}^{-1} \mathbf{v} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ z \end{pmatrix} = \begin{pmatrix} \mathbf{r} \\ \mathbf{v}^\top \mathbf{W}^{-1} \mathbf{r} \end{pmatrix}.$$

The computational steps are:

1. Solve $\mathbf{W}\bar{\mathbf{v}} = \mathbf{v}$.
2. Solve $\mathbf{W}\bar{\mathbf{r}} = \mathbf{r}$.
3. Compute $z = (1 + \mathbf{v}^\top \bar{\mathbf{v}})^{-1} \mathbf{v}^\top \bar{\mathbf{r}}$.
4. Set $\mathbf{x} = \bar{\mathbf{r}} - \bar{\mathbf{v}}z$.

Spectral Decompositions

If $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{A} = \mathbf{A}^\top$, then $\mathbf{A}$ is diagonalizable. This means that we can write

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top,$$

where $\mathbf{U}$ is orthogonal and $\mathbf{\Sigma} = \text{diag}(\lambda_1, \dots, \lambda_n)$ with $\lambda_i \in \mathbb{R}$. Also, $\mathbf{A}\mathbf{u}_i = \lambda_i\mathbf{u}_i$, where $\mathbf{u}_i$ is the i th column of $\mathbf{U}$ and hence the i th eigenvector.

In MATLAB, `eig(A)` returns the eigenvalues, while `[U, D] = eig(A)` returns the decomposition.

The full decomposition is expensive, so it is used only when necessary. In many applications, it is enough to compute only the extreme eigenvalues, such as the largest and smallest eigenvalues.

If $\mathbf{A}$ is symmetric positive definite, then $\lambda_i > 0$ for all i . If $\mathbf{A}$ is positive semidefinite, then $\lambda_i \geq 0$ for all i . For any $\mathbf{A} \in \mathbb{R}^{m \times n}$ (symmetric or otherwise), there is a **singular value decomposition**

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top,$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthogonal, and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is diagonal except for possible zero padding. Its nonzero diagonal entries are the singular values

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0.$$

The singular value decomposition exists for every matrix, but naturally it is expensive to compute, especially compared with LU, QR, or Cholesky factorizations. Its advantage is that it gives reliable information about rank, range, and null spaces. In particular, $\text{rank}(\mathbf{A})$ is the number of nonzero singular values, the first r columns of $\mathbf{U}$ span $\text{Range}(\mathbf{A})$, and the last $n - r$ columns of $\mathbf{V}$ span $\text{Null}(\mathbf{A})$.

For any $k < r$, let $\mathbf{U}_k$, $\mathbf{\Sigma}_k$, and $\mathbf{V}_k$ denote the leading rank- k pieces. Then

$$\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\top$$

is the best rank- k approximation to $\mathbf{A}$ in Frobenius norm. In MATLAB, `d = svd(A)` returns the singular values and `[U,D,V] = svd(A)` returns the full decomposition.

The singular value decomposition is enormously useful, but also expensive. Use it only if you are loaded (computationally speaking).

2. Computing Derivatives

First Derivatives and Taylor Approximation

Let

$$F(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{pmatrix} \quad \text{and} \quad f_i : \mathbb{R}^n \rightarrow \mathbb{R}.$$

Thus $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$. The matrix of first derivatives is the **Jacobian**

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} \in \mathbb{R}^{m \times n}.$$

When $m = 1$, the Jacobian is the transpose of the gradient:

$$\mathbf{J}(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right) = \nabla f(\mathbf{x})^\top.$$

Lecture 11 — Thursday, October 02, 2014

First derivatives represent rates of change. For a scalar objective function $f : \mathbb{R}^n \rightarrow \mathbb{R}$, the gradient $\nabla f(\bar{\mathbf{x}})$ points in the local direction of steepest ascent at $\bar{\mathbf{x}}$, while $-\nabla f(\bar{\mathbf{x}})$ points in the local direction of steepest descent.

Taylor's theorem says that for any direction $\mathbf{d} \in \mathbb{R}^n$ with $\|\mathbf{d}\|_2 = 1$ and any continuously differentiable $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$,

$$F(\mathbf{x} + \alpha \mathbf{d}) = F(\mathbf{x}) + \alpha \mathbf{J}(\mathbf{x}) \mathbf{d} + o(\alpha).$$

Here $g(\alpha) = o(\alpha)$ means that $g(\alpha)/\alpha \rightarrow 0$ as $\alpha \rightarrow 0$. For example, $\alpha^2 = o(\alpha)$, while $\sin(\alpha)$ is not $o(\alpha)$.

Proposition 11.1 Suppose that $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $\|\mathbf{d}\|_2 = 1$, and $\mathbf{d}^\top \nabla f(\mathbf{x}) > 0$. Then $\mathbf{d}$ is a unit ascent direction for f at $\mathbf{x}$, meaning that $f(\mathbf{x} + \alpha \mathbf{d}) > f(\mathbf{x})$ for all sufficiently small positive α .

Proof. Taylor's theorem gives

$$f(\mathbf{x} + \alpha \mathbf{d}) = f(\mathbf{x}) + \alpha \nabla f(\mathbf{x})^\top \mathbf{d} + o(\alpha).$$

Thus

$$\frac{f(\mathbf{x} + \alpha \mathbf{d}) - f(\mathbf{x})}{\alpha} = \nabla f(\mathbf{x})^\top \mathbf{d} + \frac{o(\alpha)}{\alpha} \rightarrow \nabla f(\mathbf{x})^\top \mathbf{d} > 0$$

as $\alpha \rightarrow 0$. Hence $f(\mathbf{x} + \alpha \mathbf{d}) - f(\mathbf{x}) > 0$ for all sufficiently small positive α . □

Similarly, if $\mathbf{d}^\top \nabla f(\mathbf{x}) < 0$, then $\mathbf{d}$ is a unit descent direction for f at $\mathbf{x}$.

If $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is C^1 , then

$$F(\mathbf{x} + \alpha \mathbf{d}) = F(\mathbf{x}) + \alpha \mathbf{J}(\mathbf{x}) \mathbf{d} + o(\alpha).$$

This is the **general linear approximation**.

If $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice continuously differentiable, then

$$f(\mathbf{x} + \alpha \mathbf{d}) = f(\mathbf{x}) + \alpha \nabla f(\mathbf{x})^\top \mathbf{d} + \frac{1}{2} \alpha^2 \mathbf{d}^\top \nabla^2 f(\mathbf{x}) \mathbf{d} + o(\alpha^2),$$

which is the **general quadratic approximation**. Here

$$\nabla^2 f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

is the **Hessian matrix**, which is symmetric under the usual smoothness assumptions.

Nonlinear Least Squares

Let $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, and consider the **nonlinear least squares** problem

$$\min_{\mathbf{x}} \frac{1}{2} \|F(\mathbf{x})\|_2^2.$$

Equivalently, let

$$f(\mathbf{x}) = \frac{1}{2} F(\mathbf{x})^\top F(\mathbf{x}).$$

Then

$$\nabla f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top F(\mathbf{x}),$$

and the Hessian has the structure

$$\mathbf{H}(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \sum_{i=1}^m F_i(\mathbf{x}) \nabla^2 F_i(\mathbf{x}).$$

When the residuals $F_i(\mathbf{x})$ are small, it is natural to use the approximation

$$\mathbf{H}(\mathbf{x}) \approx \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}).$$

This gives the **Gauss–Newton method**. A related approximation is

$$\mathbf{H}(\mathbf{x}) \approx \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \lambda \mathbf{I},$$

which leads to the **Levenberg–Marquardt method** (or **damped least squares**).

Lecture 12 — Friday, October 03, 2014

Finite Difference Approximations

For $f : \mathbb{R}^n \rightarrow \mathbb{R}$, let $\mathbf{e}_i$ be the i th standard basis vector. [Taylor’s theorem](#) gives us

$$f(\mathbf{x} + \alpha_i \mathbf{e}_i) = f(\mathbf{x}) + \alpha_i \nabla f(\mathbf{x})^\top \mathbf{e}_i + o(\alpha_i).$$

Rearranging gives the i th entry of the gradient:

$$\frac{f(\mathbf{x} + \alpha_i \mathbf{e}_i) - f(\mathbf{x})}{\alpha_i} = \nabla f(\mathbf{x})^\top \mathbf{e}_i + \frac{o(\alpha_i)}{\alpha_i} \rightarrow \nabla f(\mathbf{x})^\top \mathbf{e}_i$$

as $\alpha_i \rightarrow 0$.

The resulting **finite difference method** to approximate the gradient is:

1. For $i = 1, \dots, n$, choose a step size α_i .
2. Set

$$g_i(\mathbf{x}) = \frac{f(\mathbf{x} + \alpha_i \mathbf{e}_i) - f(\mathbf{x})}{\alpha_i}.$$

Then $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_n(\mathbf{x}))^\top$ approximates the gradient of f at $\mathbf{x}$. The cost of this approximation is $(n + 1)$ evaluations of f .

The step size α_i balances two competing errors. If α_i is too large, truncation error dominates because the Taylor approximation is too crude. If α_i is too small, roundoff error dominates because the subtraction $f(\mathbf{x} + \alpha_i \mathbf{e}_i) - f(\mathbf{x})$ loses precision. This is why choosing α_i near machine precision, such as 10^{-16} in double precision, is usually too small.

Similarly, we can approximate the Jacobian matrix for $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$. If $\|\mathbf{d}\|_2 = 1$, then

$$F(\mathbf{x} + \alpha \mathbf{d}) = F(\mathbf{x}) + \alpha \mathbf{J}(\mathbf{x}) \cdot \mathbf{d} + o(\alpha)$$

as $\alpha \rightarrow 0$. Thus

$$\mathbf{J}(\mathbf{x})\mathbf{d} \approx \frac{F(\mathbf{x} + \alpha \mathbf{d}) - F(\mathbf{x})}{\alpha}.$$

Taking $\mathbf{d} = \mathbf{e}_i$ gives the i th column of $\mathbf{J}(\mathbf{x})$. Computing all columns this way costs $(n + 1)$ evaluations of F .

If $\nabla f(\mathbf{x})$ is available, then Hessian–vector products can be approximated by differencing gradients. With $\mathbf{d} = \mathbf{e}_i$,

$$\mathbf{H}(\mathbf{x}) \cdot \mathbf{d} \approx \frac{\nabla f(\mathbf{x} + \alpha \mathbf{d}) - \nabla f(\mathbf{x})}{\alpha}.$$

Example 12.1 Let $F(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ where

$$F(\mathbf{x}) = \begin{pmatrix} (3 - 2x_1)x_1 - 2x_2 + 1 \\ (3 - 2x_2)x_2 - x_1 - 2x_3 + 1 \\ (3 - 2x_3)x_3 - x_2 + 1 \end{pmatrix}.$$

An easy calculation gives

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} 3 - 4x_1 & -2 & 0 \\ -1 & 3 - 4x_2 & -2 \\ 0 & -1 & 3 - 4x_3 \end{pmatrix}.$$

If $\mathbf{x} = (1, 1, 1)^\top$, then $F(\mathbf{x}) = (0, -1, 1)^\top$. Using $\alpha = 0.005$ and $\bar{\mathbf{x}} = \mathbf{x} + \alpha \mathbf{e}_1 = (1.005, 1, 1)^\top$ gives

$$\mathbf{J}(\mathbf{x}) \cdot \mathbf{e}_1 \approx \frac{F(\bar{\mathbf{x}}) - F(\mathbf{x})}{\alpha} = \begin{pmatrix} -1.0100 \\ -1.000 \\ 0 \end{pmatrix},$$

which is not far from the true value of

$$\mathbf{J}(\mathbf{x}) \cdot \mathbf{e}_1 = \begin{pmatrix} -1 \\ -1 \\ 0 \end{pmatrix}.$$

Lecture 13 — Tuesday, October 07, 2014

The finite-difference method approximates Jacobian–vector products by

$$\mathbf{J}(\mathbf{x})\mathbf{d} \approx \frac{F(\mathbf{x} + \alpha\mathbf{d}) - F(\mathbf{x})}{\alpha}.$$

If $\mathbf{d} = \mathbf{e}_i$, this recovers the i th Jacobian column. Similarly, if $\nabla f(\mathbf{x})$ is available, then

$$\mathbf{H}(\mathbf{x})\mathbf{d} \approx \frac{\nabla f(\mathbf{x} + \alpha\mathbf{d}) - \nabla f(\mathbf{x})}{\alpha}.$$

The same tradeoff in choosing the step size appears here: too small a step leads to roundoff error, while too large a step leads to truncation error.

Automatic Differentiation

Automatic differentiation (AD) computes derivatives by applying the chain rule to the elementary operations used to evaluate a function. It is not the same as symbolic differentiation: AD differentiates the executed computation rather than producing a closed-form derivative formula. See www.autodiff.org for background.

For example, let $f(x) = \sin(e^{x^2})$ and evaluate at $x = 0.5$. In forward-mode AD, we carry each intermediate quantity together with its derivative with respect to the input. We therefore initialize

$$x.\text{val} = 0.5 \quad \text{and} \quad x.\text{deriv} = 1.$$

Now compute each operation in order.

$$\begin{aligned} y_1 &= x^2, \\ y_1.\text{val} &= (x.\text{val})^2 = (0.5)^2 = 0.25, \\ y_1.\text{deriv} &= 2x.\text{val}x.\text{deriv} = 2(0.5)(1) = 1. \end{aligned}$$

$$\begin{aligned} y_2 &= e^{y_1}, \\ y_2.\text{val} &= e^{y_1.\text{val}} = e^{0.25} \approx 1.2840, \\ y_2.\text{deriv} &= e^{y_1.\text{val}} y_1.\text{deriv} \approx 1.2840 \cdot 1 = 1.2840. \end{aligned}$$

$$\begin{aligned} y_3 &= \sin(y_2), \\ y_3.\text{val} &= \sin(y_2.\text{val}) = \sin(1.2840) \approx 0.9594, \\ y_3.\text{deriv} &= \cos(y_2.\text{val}) y_2.\text{deriv} \approx \cos(1.2840) \cdot 1.2840 \approx 0.3639. \end{aligned}$$

Hence

$$f(0.5) = y_3.\text{val} \approx 0.9594 \quad \text{and} \quad f'(0.5) = y_3.\text{deriv} \approx 0.3639.$$

Forward-mode AD follows the computation graph in [Figure 4](#), carrying a value and its directional derivative through each elementary operation. The chain rule is therefore local: every node needs information only from its immediate predecessor.

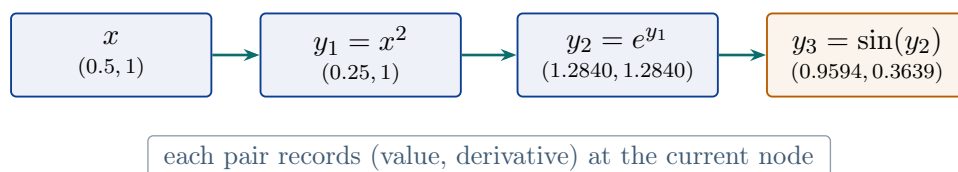


FIGURE 4

Remark 13.1 Suppose $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$.

1. If both forward and reverse modes are available, a dense Jacobian can be assembled with n forward sweeps or m reverse sweeps. Its cost is therefore typically

$O(\min\{m, n\} \omega(F))$, where $\omega(F)$ is the cost of evaluating F , up to implementation-dependent constants.

2. For a scalar output, reverse mode computes the gradient for a small constant multiple of the evaluation cost; the precise multiple depends on the computation and implementation.
3. A dense Hessian can be assembled from n Hessian–vector products, so its cost is typically $O(n \omega(f))$ up to comparable constants.

Sparse Jacobians

For a sparse Jacobian with a structure such as

$$\mathbf{J}(\mathbf{x}) = \begin{pmatrix} \times & \times & 0 \\ \times & 0 & \times \\ 0 & \times & \times \end{pmatrix}$$

we can sometimes choose directions that recover multiple entries at once, especially when forward products $\mathbf{J}\mathbf{d}$ and reverse products $\mathbf{J}^\top \mathbf{d}$ are both available. Sparsity alone does not guarantee a saving, however. In this particular pattern, every pair of columns shares a row containing two possible nonzero entries. Combining any two columns in one forward direction would therefore add two unknown entries in that row, so three forward directions are still required. The larger pattern below has genuinely structurally orthogonal column groups and does permit compression.

Lecture 14 — Thursday, October 09, 2014

Finite differences approximate

$$\mathbf{J}(\mathbf{x})\mathbf{d} \approx \frac{F(\mathbf{x} + \alpha\mathbf{d}) - F(\mathbf{x})}{\alpha},$$

whereas automatic differentiation computes $\mathbf{J}(\mathbf{x})\mathbf{d}$ and $\mathbf{J}(\mathbf{x})^\top \mathbf{d}$ to machine precision.

For a larger example of a sparse Jacobian, suppose

$$\mathbf{J} = \begin{pmatrix} \times & \times & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 \\ 0 & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times \end{pmatrix}.$$

A naive finite-difference method uses all five standard basis directions. The sparsity pattern allows fewer grouped directions, such as

$$\mathbf{d}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{d}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad \text{and} \quad \mathbf{d}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

The main message is that sparsity and structure can substantially reduce the cost of derivative computations.

3. Unconstrained Optimization and Nonlinear Systems

Quadratic Models and First-Order Equations

We study nonlinear systems

$$F(\mathbf{x}) = \mathbf{0}, \quad F : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

and unconstrained minimization problems

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad f : \mathbb{R}^n \rightarrow \mathbb{R}.$$

A linear objective (or “trivial” function) $f(\mathbf{x}) = \mathbf{c}^\top \mathbf{x}$ with $\mathbf{c} \neq \mathbf{0}_n$ is unbounded below. Indeed, if $\bar{\mathbf{x}} = -\alpha \mathbf{c}$ with $\alpha > 0$, then

$$f(\bar{\mathbf{x}}) = \mathbf{c}^\top (-\alpha \mathbf{c}) = -\alpha \|\mathbf{c}\|^2 \rightarrow -\infty$$

as $\alpha \rightarrow \infty$.

The next simplest model is a quadratic function (or “easy” function)

$$f(\mathbf{x}) = \mathbf{g}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{H} \mathbf{x},$$

where we may assume $\mathbf{H} = \mathbf{H}^\top$. If $\mathbf{H}$ is not symmetric, then

$$\mathbf{x}^\top \mathbf{H} \mathbf{x} = \frac{1}{2} (\mathbf{x}^\top \mathbf{H} \mathbf{x} + \mathbf{x}^\top \mathbf{H}^\top \mathbf{x}) = \frac{1}{2} \mathbf{x}^\top (\mathbf{H} + \mathbf{H}^\top) \mathbf{x},$$

and $(\mathbf{H} + \mathbf{H}^\top)/2$ is symmetric. If $\mathbf{x}^*$ is a minimizer, then

$$\nabla f(\mathbf{x}^*) = \mathbf{g} + \mathbf{H} \mathbf{x}^* = \mathbf{0}_n \implies \mathbf{x}^* = -\mathbf{H}^{-1} \mathbf{g}$$

is the natural candidate when $\mathbf{H}$ is nonsingular.

Proposition 14.1 For the easy function

$$f(\mathbf{x}) = \mathbf{g}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{H} \mathbf{x},$$

the following hold:

1. If $\mathbf{H}$ is symmetric positive definite, then there is a unique global minimizer $\mathbf{x}^* = -\mathbf{H}^{-1} \mathbf{g}$.
2. If $\mathbf{H}$ is symmetric but not positive semidefinite, then f is unbounded below.
3. If $\mathbf{H}$ is symmetric positive semidefinite and singular, then either f is unbounded below or f has infinitely many minimizers.

Proof. For (1), assume $\mathbf{H}$ is symmetric positive definite. Then $\mathbf{H}$ is nonsingular, so the first-order condition

$$\nabla f(\mathbf{x}) = \mathbf{g} + \mathbf{H} \mathbf{x} = \mathbf{0}_n$$

has the unique solution $\mathbf{x}^* = -\mathbf{H}^{-1} \mathbf{g}$. For any $\mathbf{x}$,

$$f(\mathbf{x}) - f(\mathbf{x}^*) = \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^\top \mathbf{H} (\mathbf{x} - \mathbf{x}^*).$$

Because $\mathbf{H}$ is positive definite, the right-hand side is strictly positive whenever $\mathbf{x} \neq \mathbf{x}^*$. Hence $\mathbf{x}^*$ is the unique global minimizer.

For (2), assume $\mathbf{H}$ is symmetric but not positive semidefinite. Then there exists $\mathbf{z} \neq \mathbf{0}_n$ with

$\mathbf{z}^\top \mathbf{H} \mathbf{z} < 0$. Replacing $\mathbf{z}$ by $-\mathbf{z}$ if needed, we may assume $\mathbf{g}^\top \mathbf{z} \leq 0$. Along the ray $\mathbf{x} = \alpha \mathbf{z}$,

$$f(\alpha \mathbf{z}) = \alpha \mathbf{g}^\top \mathbf{z} + \frac{1}{2} \alpha^2 \mathbf{z}^\top \mathbf{H} \mathbf{z}.$$

The quadratic coefficient is negative, so $f(\alpha \mathbf{z}) \rightarrow -\infty$ as $\alpha \rightarrow \infty$. Therefore f is unbounded below.

For (3), assume $\mathbf{H}$ is symmetric positive semidefinite and singular.

If $\mathbf{g} \notin \text{Range}(\mathbf{H})$, then, since $\mathbf{H}$ is symmetric,

$$\text{Range}(\mathbf{H}) = (\ker(\mathbf{H}))^\perp.$$

So there exists $\mathbf{z} \in \ker(\mathbf{H})$ with $\mathbf{g}^\top \mathbf{z} \neq 0$. For any $\mathbf{x}$,

$$f(\mathbf{x} + \alpha \mathbf{z}) = f(\mathbf{x}) + \alpha \mathbf{g}^\top \mathbf{z}$$

because $\mathbf{H} \mathbf{z} = \mathbf{0}_n$ eliminates the mixed and quadratic terms in α . Choosing the sign of α so that $\alpha \mathbf{g}^\top \mathbf{z} \rightarrow -\infty$ shows that f is unbounded below.

If $\mathbf{g} \in \text{Range}(\mathbf{H})$, choose $\mathbf{x}^*$ such that $\mathbf{H} \mathbf{x}^* = -\mathbf{g}$. Then, for any $\mathbf{x}$,

$$f(\mathbf{x}) - f(\mathbf{x}^*) = \frac{1}{2} (\mathbf{x} - \mathbf{x}^*)^\top \mathbf{H} (\mathbf{x} - \mathbf{x}^*) \geq 0,$$

so $\mathbf{x}^*$ is a global minimizer. Since $\mathbf{H}$ is singular, there exists nonzero $\mathbf{z} \in \ker(\mathbf{H})$. Also,

$$\mathbf{g}^\top \mathbf{z} = (-\mathbf{H} \mathbf{x}^*)^\top \mathbf{z} = -(\mathbf{x}^*)^\top \mathbf{H} \mathbf{z} = 0.$$

Hence

$$f(\mathbf{x}^* + \mathbf{z}) = f(\mathbf{x}^*) + \mathbf{g}^\top \mathbf{z} + \frac{1}{2} \mathbf{z}^\top \mathbf{H} \mathbf{z} = f(\mathbf{x}^*),$$

and similarly $f(\mathbf{x}^* + \alpha \mathbf{z}) = f(\mathbf{x}^*)$ for every $\alpha \in \mathbb{R}$. Therefore there are infinitely many minimizers. $\square$

Lecture 15 — Friday, October 10, 2014

Newton's Method

Most methods for nonlinear minimization are iterative. Starting from $\mathbf{x}_0$, they construct $\mathbf{x}_1, \mathbf{x}_2, \dots$, with the hope that $\mathbf{x}_k \rightarrow \mathbf{x}_*$ as $k \rightarrow \infty$. A method is **globally convergent** if it converges from every starting point, and **locally convergent** if it converges whenever $\mathbf{x}_0$ is sufficiently close to the desired solution.

To minimize f , we must solve $\nabla f(\mathbf{x}) = \mathbf{0}_n$. This motivates studying nonlinear equations first. For $F(\mathbf{x}) = \mathbf{0}_n$, the basic idea is to approximate F near $\mathbf{x}_k$ by

$$F(\mathbf{x}_k + \mathbf{s}) \approx F(\mathbf{x}_k) + \mathbf{B}_k \mathbf{s},$$

where $\mathbf{B}_k$ is $\mathbf{J}(\mathbf{x}_k)$ or an approximation to it. Setting this approximation equal to zero gives

$$\mathbf{B}_k \mathbf{s} = -F(\mathbf{x}_k),$$

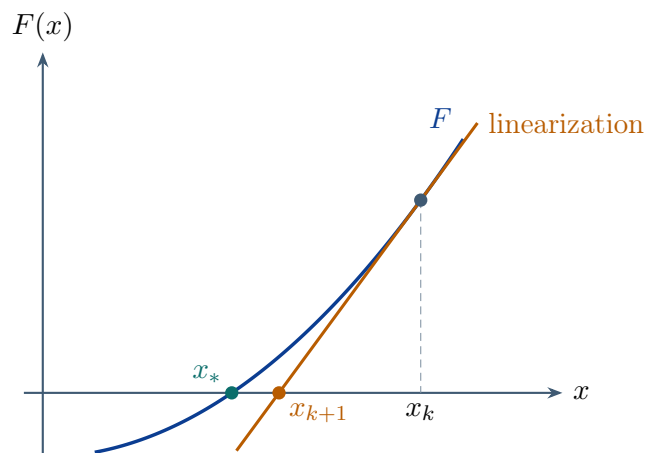
and then $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}$.

The general **Newton process** starts with $\mathbf{x}_0$, an initial matrix $\mathbf{B}_0$, and $k = 0$:

1. Evaluate $F(\mathbf{x}_k)$ and choose $\mathbf{B}_k$.
2. Solve $\mathbf{B}_k \mathbf{s}_k = -F(\mathbf{x}_k)$.
3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$ and increment k .

If $\mathbf{B}_k = \mathbf{J}(\mathbf{x}_k)$ exactly, this is **Newton's method**. If $\mathbf{B}_k$ approximates $\mathbf{J}(\mathbf{x}_k)$, the method is a **quasi-Newton method**. Newton-type methods can be very fast, but the basic theory is local: convergence is guaranteed only when the starting point is sufficiently close to the solution and the Jacobian approximation remains sufficiently accurate.

For a scalar equation, the update has the familiar tangent-line geometry shown in [Figure 5](#): replace the function by its linearization at x_k , then take the zero of that line as x_{k+1} .



Newton's step solves the local linear model, not the original equation

FIGURE 5

Theorem 15.1 Assume that F is continuously differentiable, $F(\mathbf{x}_*) = \mathbf{0}_n$, and $\mathbf{J}(\mathbf{x}_*)$ is nonsingular. Generate $\{\mathbf{x}_k\}$ by the Newton process. Then there are positive constants δ_1 and δ_2 such that if $\|\mathbf{x}_0 - \mathbf{x}_*\| \leq \delta_1$ and $\|\mathbf{B}_k - \mathbf{J}(\mathbf{x}_k)\| \leq \delta_2$ for all k , then $\{\mathbf{x}_k\}$ converges linearly to $\mathbf{x}_*$.

Lecture 16 — Tuesday, October 14, 2014

When $\mathbf{B}_k = \mathbf{J}(\mathbf{x}_k)$, the update can be written

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}(\mathbf{x}_k)]^{-1} F(\mathbf{x}_k).$$

When $\mathbf{B}_k \approx \mathbf{J}(\mathbf{x}_k)$, the same process gives a quasi-Newton method.

Convergence Rates

1. The sequence $\{\mathbf{x}_k\}$ converges **linearly** to $\mathbf{x}_*$ if there are $\bar{k}$ and $c \in (0, 1)$ such that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq c \|\mathbf{x}_k - \mathbf{x}_*\|$$

for all $k \geq \bar{k}$. For example, we might have $\text{err}_{k+1} \approx 0.1 \text{ err}_k$.

2. The sequence converges **superlinearly** if

$$\frac{\|\mathbf{x}_{k+1} - \mathbf{x}_*\|}{\|\mathbf{x}_k - \mathbf{x}_*\|} \rightarrow 0$$

as $k \rightarrow \infty$, whenever the denominator is nonzero.

3. The sequence converges **quadratically** if there are $\bar{k}$ and $c > 0$ such that

$$\|\mathbf{x}_{k+1} - \mathbf{x}_*\| \leq c \|\mathbf{x}_k - \mathbf{x}_*\|^2$$

for all $k \geq \bar{k}$.

For example,

$$\{\|\mathbf{x}_k - \mathbf{x}_*\|\}_{k=1}^{\infty} = \{2^{-k}\}_{k=1}^{\infty}$$

converges linearly, while a typical quadratic pattern looks like $\{2^{-2^k}\}_{k=1}^{\infty}$. Under standard smoothness and nonsingularity assumptions, Newton's method converges quadratically near the solution.

Example 16.1 To compute $\sqrt{a}$ for $a > 0$, solve $F(\mathbf{x}) = x^2 - a = 0$. Newton's method gives

$$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k}.$$

Then

$$x_{k+1} - \sqrt{a} = \frac{1}{2x_k} (x_k - \sqrt{a})^2.$$

Thus the error satisfies

$$|x_{k+1} - \sqrt{a}| = \frac{1}{2|x_k|} |x_k - \sqrt{a}|^2,$$

which exhibits quadratic convergence when x_k stays near $\sqrt{a}$.

For minimization, the equation to solve is $\nabla f(\mathbf{x}) = 0$. This leads to the standard optimality conditions:

Theorem 16.2 (Standard optimality conditions)

1. **First-order necessary condition:** if f is continuously differentiable and $\mathbf{x}_*$ is a local minimizer, then $\nabla f(\mathbf{x}_*) = \mathbf{0}_n$.
2. **Second-order necessary condition:** if f is twice continuously differentiable and $\mathbf{x}_*$

is a local minimizer, then $\nabla f(\mathbf{x}_*) = \mathbf{0}_n$ and $\nabla^2 f(\mathbf{x}_*)$ is positive semidefinite.

3. **Second-order sufficient condition:** If f is twice continuously differentiable, $\nabla f(\mathbf{x}_*) = \mathbf{0}_n$, and $\nabla^2 f(\mathbf{x}_*)$ is symmetric positive definite, then $\mathbf{x}_*$ is a strict local minimizer.

Lecture 19 — Tuesday, October 21, 2014

The second-order sufficient condition says that if $\nabla f(\mathbf{x}_*) = \mathbf{0}_n$ and $\nabla^2 f(\mathbf{x}_*)$ is symmetric positive definite, then $\mathbf{x}_*$ is a strict local minimizer.

Descent Direction Methods

We say that $\mathbf{d}$ is a **descent direction** of f at $\bar{\mathbf{x}}$ if there is $\bar{\alpha} > 0$ such that

$$f(\bar{\mathbf{x}} + \alpha \mathbf{d}) < f(\bar{\mathbf{x}})$$

for all $0 < \alpha < \bar{\alpha}$.

If $\nabla f(\bar{\mathbf{x}})^\top \mathbf{d} < 0$, then $\mathbf{d}$ is a descent direction. Indeed, [Taylor's theorem](#) gives

$$f(\bar{\mathbf{x}} + \alpha \mathbf{d}) = f(\bar{\mathbf{x}}) + \alpha \nabla f(\bar{\mathbf{x}})^\top \mathbf{d} + o(\alpha) = f(\bar{\mathbf{x}}) + \alpha \left(\nabla f(\bar{\mathbf{x}})^\top \mathbf{d} + \frac{o(\alpha)}{\alpha} \right),$$

which is less than $f(\bar{\mathbf{x}})$ for all sufficiently small positive α .

The generic **descent direction method** is as follows:

1. Construct a descent direction $\mathbf{d}_k$ with $\nabla f(\mathbf{x}_k)^\top \mathbf{d}_k < 0$.
2. Choose a step length, often approximately, by minimizing along $\mathbf{d}_k$:

$$\alpha_k \approx \arg \min_{\alpha > 0} f(\mathbf{x}_k + \alpha \mathbf{d}_k).$$

3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$.

A common choice of $\mathbf{d}_k$ is the steepest descent direction $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$. Taking the raw unit step

in that direction can be too aggressive (imagine dropping off the roof of MC in order to get to the ground floor), which is why the line search controls the step length α_k . Another common choice is an approximate Newton direction obtained by solving

$$\mathbf{H}_k \mathbf{d}_k = -\nabla f(\mathbf{x}_k),$$

where $\mathbf{H}_k \approx \nabla^2 f(\mathbf{x}_k)$. In a quasi-Newton method, $\mathbf{H}_{k+1}$ is updated using $\mathbf{H}_k$, the step $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, and the gradient difference $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$. Low-rank updates let us incorporate this new information without rebuilding the approximation from scratch. A general rank-two update takes the form

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \mathbf{a}\mathbf{b}^\top + \mathbf{c}\mathbf{d}^\top.$$

A famous example is the **BFGS** (Broyden–Fletcher–Goldfarb–Shanno) method:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^\top}{\mathbf{y}_k^\top \mathbf{s}_k} - \frac{\mathbf{H}_k \mathbf{s}_k \mathbf{s}_k^\top \mathbf{H}_k}{\mathbf{s}_k^\top \mathbf{H}_k \mathbf{s}_k}.$$

If $\mathbf{H}_k$ is positive definite, $\mathbf{s}_k \neq \mathbf{0}_n$, and the curvature condition $\mathbf{y}_k^\top \mathbf{s}_k > 0$ holds, both denominators are nonzero and the BFGS update preserves positive definiteness. Under appropriate additional conditions, it also gives superlinear convergence. As a bonus, it provides an explicit formula for $\mathbf{H}_{k+1}^{-1}$.

Theorem 19.1 If $\mathbf{H}_k$ is symmetric positive definite and $\mathbf{H}_k \mathbf{d}_k = -\nabla f(\mathbf{x}_k)$, then $\mathbf{d}_k$ is a descent direction for f at $\mathbf{x}_k$.

Proof. Since $\mathbf{d}_k = -\mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k)$,

$$\nabla f(\mathbf{x}_k)^\top \mathbf{d}_k = -\nabla f(\mathbf{x}_k)^\top \mathbf{H}_k^{-1} \nabla f(\mathbf{x}_k).$$

Because $\mathbf{H}_k^{-1}$ is symmetric positive definite, the right-hand side is strictly negative unless $\nabla f(\mathbf{x}_k) = \mathbf{0}_n$. Thus $\mathbf{d}_k$ is a descent direction whenever $\mathbf{x}_k$ is not stationary. $\square$

Lecture 20 — Tuesday, October 28, 2014

The general descent framework will reappear in the trust region method, where the step is controlled by a local model rather than by a line search alone.

Trust Region Methods

Recall that [Taylor's theorem](#) gives

$$f(\mathbf{x}_k + \mathbf{s}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{H}(\mathbf{x}_k) \mathbf{s} + o(\|\mathbf{s}\|_2^2).$$

The **trust region method** minimizes this quadratic model inside a ball of radius Δ_k :

$$\min \left\{ \nabla f(\mathbf{x}_k)^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{H}(\mathbf{x}_k) \mathbf{s} \mid \|\mathbf{s}\|_2 \leq \Delta_k \right\}.$$

The vector $\mathbf{s}$ is called the **trial step** for Δ_k . A typical trust region iteration solves this subproblem approximately, accepts $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}$ if the actual decrease is sufficient, and adjusts Δ_k according to how well the model predicts the actual objective decrease.

The ball in [Figure 6](#) limits how far we trust the quadratic model. When the unconstrained Newton step lies outside it, the trial step terminates on the boundary instead.

Provided that the model predicts a positive reduction, the usual ratio compares actual reduction to predicted reduction:

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{s})}{-\nabla f(\mathbf{x}_k)^\top \mathbf{s} - \frac{1}{2} \mathbf{s}^\top \mathbf{H}(\mathbf{x}_k) \mathbf{s}}.$$

If the higher-order term is negligible, then $\rho_k \approx 1$. In practice, we may enlarge the region when $\rho_k > 3/4$, shrink it when $\rho_k < 1/4$, and otherwise leave it unchanged.

Under standard assumptions, a well-designed trust region method converges to a stationary point; it provides no blanket guarantee of finding a global minimizer (go hire a guy from IBM for that). Near a nondegenerate local minimizer, methods that eventually take the full Newton step can converge quadratically.

The trust region subproblem of finding the constrained minimizer is subtle, but efficient methods

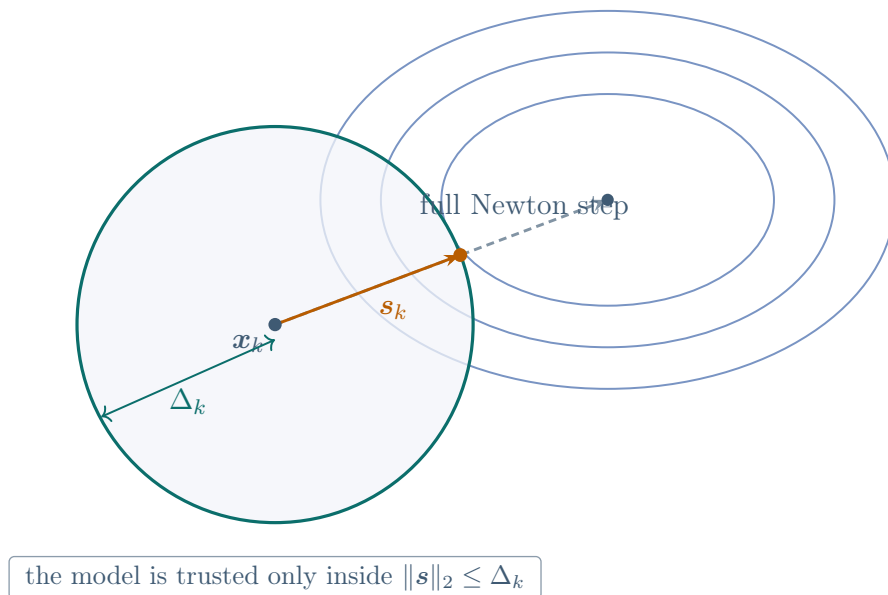


FIGURE 6

exist. If there were no trust region constraint, the model

$$\mathbf{g}^\top \mathbf{s} + \frac{1}{2} \mathbf{s}^\top \mathbf{H} \mathbf{s}$$

would have the minimizer $\mathbf{s}_* = -\mathbf{H}^{-1} \mathbf{g}$ when $\mathbf{H}$ is symmetric positive definite. With the constraint present, a vector $\mathbf{s}$ is a global solution precisely when there is a parameter $\lambda_k \geq 0$ satisfying

$$\begin{cases} (\mathbf{H}(\mathbf{x}_k) + \lambda_k \mathbf{I}) \mathbf{s} = -\nabla f(\mathbf{x}_k), \\ \mathbf{H}(\mathbf{x}_k) + \lambda_k \mathbf{I} \succeq \mathbf{0}, \\ \|\mathbf{s}\|_2 \leq \Delta_k, \\ \lambda_k (\Delta_k - \|\mathbf{s}\|_2) = 0. \end{cases}$$

If $\mathbf{H}(\mathbf{x}_k)$ is positive definite and the unconstrained step satisfies $\|\mathbf{s}\|_2 \leq \Delta_k$, then $\lambda_k = 0$. Otherwise, the solution lies on the boundary $\|\mathbf{s}\|_2 = \Delta_k$. In general, positive semidefiniteness requires $\lambda_k \geq \max\{0, -\lambda_{\min}\}$, where $\lambda_{\min}$ is the smallest eigenvalue of $\mathbf{H}(\mathbf{x}_k)$.

Lecture 21 — Thursday, October 30, 2014

The practical stopping conditions for an iterative method are usually based on small first-order residuals or small steps; for example, we might stop when

$$\|\nabla f(\mathbf{x}_k)\| \leq \text{tol}_1 \quad \text{or} \quad \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \leq \text{tol}_2.$$

For nonlinear least squares,

$$f(\mathbf{x}) = \frac{1}{2} \|F(\mathbf{x})\|_2^2,$$

and we have

$$\nabla f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top F(\mathbf{x}) \quad \text{and} \quad \nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \sum_{i=1}^m F_i(\mathbf{x}) \nabla^2 F_i(\mathbf{x}).$$

For a descent direction method, we could choose $\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$ and control its length by a line search. A curvature-informed alternative is the Newton step $\mathbf{H}(\mathbf{x}_k)\mathbf{d}_k = -\nabla f(\mathbf{x}_k)$; however, computing $\mathbf{H}(\mathbf{x}_k)$ is expensive, so we often approximate it. Neglecting the residual-weighted second-derivative term gives the Gauss–Newton system

$$\mathbf{J}(\mathbf{x}_k)^\top \mathbf{J}(\mathbf{x}_k) \mathbf{d}_k = -\mathbf{J}(\mathbf{x}_k)^\top F(\mathbf{x}_k)$$

and the Levenberg–Marquardt variant

$$(\mathbf{J}(\mathbf{x}_k)^\top \mathbf{J}(\mathbf{x}_k) + \lambda \mathbf{I}) \mathbf{d}_k = -\nabla f(\mathbf{x}_k).$$

The most expensive part of the trust region method is the trust region subproblem. If we use an approximation, the optimization problem can also be written in least-squares form as

$$\min \{ \|\mathbf{J}(\mathbf{x}_k)\mathbf{s} + F(\mathbf{x}_k)\|_2^2 \mid \|\mathbf{s}\| \leq \Delta_k \}.$$

Lecture 22 — Friday, October 31, 2014

4. Constrained Optimization

So far we have considered unconstrained problems of the form $\min f(\mathbf{x})$, where $f : \mathbb{R}^n \rightarrow \mathbb{R}$. We now introduce constraints.

Equality-Constrained Optimization

Instead of searching over all of $\mathbb{R}^n$, we search over a feasible set such as an affine subspace.

As a finance example, suppose Eva wants to invest in 12 stocks $S_1, \dots, S_{12}$ with portfolio weights $x_1, \dots, x_{12}$ satisfying the budget constraint

$$\sum_{i=1}^{12} x_i = 1.$$

Suppose 25% of the investment must go to the technology stocks $S_1, \dots, S_4$, and 20% must go to the energy stocks S_{10}, S_{11}, S_{12} . If the desired portfolio return is r_p , then

$$\sum_{i=1}^{12} x_i \bar{r}_i = r_p,$$

where $\bar{r}_i$ is the expected return of stock i .

Suppose the returns are estimated from 100 observations collected in the data matrix

$$\mathbf{R} = \begin{pmatrix} r_1^1 & \dots & r_{12}^1 \\ \vdots & \ddots & \vdots \\ r_1^{100} & \dots & r_{12}^{100} \end{pmatrix} \in \mathbb{R}^{100 \times 12}.$$

Then

$$\bar{r}_i = \frac{1}{100} \sum_{j=1}^{100} r_i^j.$$

Let $\bar{\mathbf{r}} = (\bar{r}_1, \dots, \bar{r}_{12})^\top$, and let $\bar{\mathbf{R}} \in \mathbb{R}^{100 \times 12}$ denote the matrix whose every row is $\bar{\mathbf{r}}^\top$. The goal is to minimize portfolio risk:

$$\min_{\mathbf{x}} \text{Risk}(\mathbf{x}) := \mathbb{E} \left[(\bar{\mathbf{r}}^\top \mathbf{x} - \mathbf{r}^\top \mathbf{x})^2 \right] = \frac{1}{100} \sum_{i=1}^{100} \left[\bar{\mathbf{r}}^\top \mathbf{x} - (\mathbf{r}^i)^\top \mathbf{x} \right]^2.$$

In matrix form the objective function is

$$\frac{1}{100} \|\bar{\mathbf{R}}\mathbf{x} - \mathbf{R}\mathbf{x}\|_2^2 = \frac{1}{100} \|\mathbf{A}_1\mathbf{x}\|_2^2,$$

where $\mathbf{A}_1 = \bar{\mathbf{R}} - \mathbf{R}$. The complete constrained problem is therefore to minimize $\|\mathbf{A}_1\mathbf{x}\|_2^2$ subject to

$$\sum_{i=1}^{12} x_i = 1, \quad (1, 1, 1, 1, 0, \dots, 0)\mathbf{x} = 0.25, \quad (0, \dots, 0, 1, 1, 1)\mathbf{x} = 0.20, \quad \text{and} \quad \bar{\mathbf{r}}^\top \mathbf{x} = r_p.$$

Let

$$\mathbf{A}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \bar{r}_1 & \bar{r}_2 & \bar{r}_3 & \bar{r}_4 & \bar{r}_5 & \bar{r}_6 & \bar{r}_7 & \bar{r}_8 & \bar{r}_9 & \bar{r}_{10} & \bar{r}_{11} & \bar{r}_{12} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0.25 \\ 0.20 \\ r_p \end{pmatrix}.$$

Then the problem becomes

$$\min \mathbf{x}^\top \mathbf{A}_1^\top \mathbf{A}_1 \mathbf{x} \quad \text{subject to} \quad \mathbf{A}_2 \mathbf{x} = \mathbf{b},$$

which is a quadratic problem with linear equality constraints.

Null Space Approach and Optimality Conditions

Consider the quadratic problem

$$\min \left\{ \mathbf{g}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{H} \mathbf{x} \mid \mathbf{A} \mathbf{x} = \mathbf{b} \right\}.$$

Assume $\mathbf{Z}$ is a basis for $\text{Null}(\mathbf{A})$ and $\mathbf{x}_0$ is a known feasible point. Then every feasible point can be written as $\mathbf{x} = \mathbf{x}_0 + \mathbf{Z}\mathbf{w}$. Substitution gives the unconstrained problem $\min_{\mathbf{w}} \bar{f}(\mathbf{w})$, where

$$\bar{f}(\mathbf{w}) = \mathbf{g}^\top (\mathbf{x}_0 + \mathbf{Z}\mathbf{w}) + \frac{1}{2} (\mathbf{x}_0 + \mathbf{Z}\mathbf{w})^\top \mathbf{H} (\mathbf{x}_0 + \mathbf{Z}\mathbf{w})$$

$$= \mathbf{g}^\top \mathbf{x}_0 + \mathbf{w}^\top \mathbf{Z}^\top \mathbf{g} + \frac{1}{2} \mathbf{x}_0^\top \mathbf{H} \mathbf{x}_0 + \mathbf{w}^\top \mathbf{Z}^\top \mathbf{H} \mathbf{x}_0 + \frac{1}{2} \mathbf{w}^\top \mathbf{Z}^\top \mathbf{H} \mathbf{Z} \mathbf{w}.$$

The first-order condition is

$$\nabla \bar{f}(\mathbf{w}) = \mathbf{Z}^\top \mathbf{g} + \mathbf{Z}^\top \mathbf{H} \mathbf{x}_0 + \mathbf{Z}^\top \mathbf{H} \mathbf{Z} \mathbf{w} = \mathbf{0}_{n-m},$$

so

$$(\mathbf{Z}^\top \mathbf{H} \mathbf{Z}) \mathbf{w} = \mathbf{Z}^\top (-\mathbf{g} - \mathbf{H} \mathbf{x}_0).$$

If $\mathbf{H}$ is symmetric positive definite, then $\mathbf{Z}^\top \mathbf{H} \mathbf{Z}$ is symmetric positive definite on the reduced space, so Cholesky factorization can be used.

The resulting null space algorithm is as follows:

1. Compute $\mathbf{x}_0$ and $\mathbf{Z}$, for example by the QR method.
2. Form $\bar{\mathbf{H}} = \mathbf{Z}^\top \mathbf{H} \mathbf{Z}$ and $\bar{\mathbf{r}} = \mathbf{Z}^\top (-\mathbf{g} - \mathbf{H} \mathbf{x}_0)$.
3. Solve $\bar{\mathbf{H}} \mathbf{w} = \bar{\mathbf{r}}$ by Cholesky factorization.
4. Set $\mathbf{x}_* = \mathbf{x}_0 + \mathbf{Z} \mathbf{w}$.

Lecture 23 — Tuesday, November 04, 2014

Now consider the more general equality-constrained problem

$$\min\{f(\mathbf{x}) \mid \mathbf{A} \mathbf{x} = \mathbf{b}\},$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$, $m < n$, and $\text{rank}(\mathbf{A}) = m$.

Geometrically, in the two-dimensional case with $\mathbf{A} = \mathbf{a}^\top$, the constraint $\mathbf{a}^\top \mathbf{x} = b$ is a line. At a constrained minimizer, the gradient of f must be orthogonal to the feasible directions and hence parallel to $\mathbf{a}$: there can be no gradient component along the feasible line.

This tangency condition is visible in [Figure 7](#). The objective's smallest attainable level set touches the feasible line at $\mathbf{x}_*$, where the gradient is normal to the feasible directions.

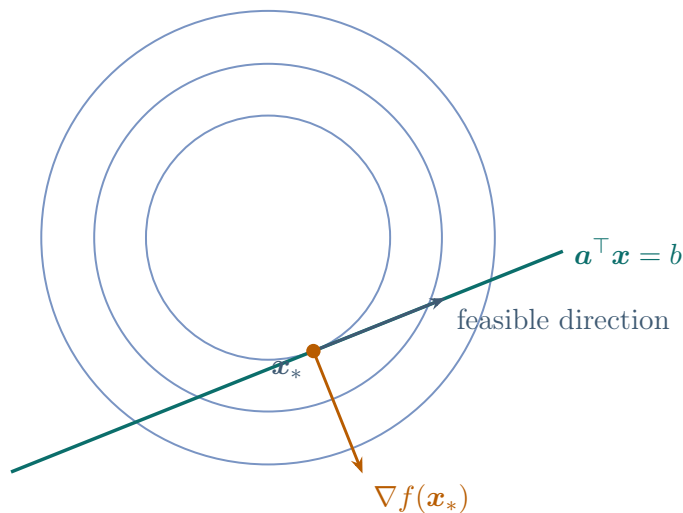


FIGURE 7

Theorem 23.1 If $\bar{x}$ is a minimizer of $\min\{f(x) \mid \mathbf{A}x = \mathbf{b}\}$, then $\nabla f(\bar{x}) = \mathbf{A}^\top \boldsymbol{\lambda}$ for some $\boldsymbol{\lambda} \in \mathbb{R}^m$.

Proof. Let the columns of $\mathbf{Z}$ form a basis for $\text{Null}(\mathbf{A})$. If $\nabla f(\bar{x})$ has a nonzero null-space component, write $\nabla f(\bar{x}) = \mathbf{A}^\top \boldsymbol{\lambda}_1 + \mathbf{Z}\boldsymbol{\lambda}_2$ with $\boldsymbol{\lambda}_2 \neq \mathbf{0}_{n-m}$. Taking $\mathbf{d} = -\mathbf{Z}\boldsymbol{\lambda}_2$, we have $\mathbf{A}\mathbf{d} = \mathbf{0}_m$, so $\bar{x} + \alpha\mathbf{d}$ remains feasible for small α . Moreover,

$$f(\bar{x} + \alpha\mathbf{d}) = f(\bar{x}) + \alpha \nabla f(\bar{x})^\top \mathbf{d} + o(\alpha) = f(\bar{x}) - \alpha \boldsymbol{\lambda}_2^\top \mathbf{Z}^\top \mathbf{Z} \boldsymbol{\lambda}_2 + o(\alpha) < f(\bar{x})$$

for sufficiently small positive α , a contradiction. $\square$

Theorem 23.2 Let $\mathbf{Z}$ be a basis for $\text{Null}(\mathbf{A})$. We have $\nabla f(\bar{x}) = \mathbf{A}^\top \boldsymbol{\lambda}$ for some $\boldsymbol{\lambda}$ if and only if $\mathbf{Z}^\top \nabla f(\bar{x}) = \mathbf{0}_{n-m}$. Equivalently, $\nabla f(\bar{x})$ is orthogonal to $\text{Null}(\mathbf{A})$.

Proof. If $\nabla f(\bar{x}) = \mathbf{A}^\top \boldsymbol{\lambda}$, then $\mathbf{Z}^\top \nabla f(\bar{x}) = \mathbf{Z}^\top \mathbf{A}^\top \boldsymbol{\lambda} = \mathbf{0}_{n-m}$. Conversely, decompose $\nabla f(\bar{x}) = \mathbf{A}^\top \boldsymbol{\lambda} + \mathbf{Z}\boldsymbol{\lambda}_2$. If $\mathbf{Z}^\top \nabla f(\bar{x}) = \mathbf{0}_{n-m}$, then $\mathbf{Z}^\top \mathbf{Z}\boldsymbol{\lambda}_2 = \mathbf{0}_{n-m}$. Since $\mathbf{Z}^\top \mathbf{Z}$ is symmetric positive definite, $\boldsymbol{\lambda}_2 = \mathbf{0}_{n-m}$, so $\nabla f(\bar{x}) = \mathbf{A}^\top \boldsymbol{\lambda}$. $\square$

The null space approach reduces the constrained problem to an unconstrained one. Assume

throughout that $\mathbf{A}$ has full row rank and that the columns of $\mathbf{Z}$ span $\text{Null}(\mathbf{A})$. The first-order statement below requires f to be continuously differentiable, while the second-order statements require f to be twice continuously differentiable near $\mathbf{x}_*$.

Theorem 23.3 (Null space approach: optimality conditions)

1. **First-order necessary condition:** if $\mathbf{x}_*$ is a local minimizer and $\mathbf{x}_* = \mathbf{x}_0 + \mathbf{Z}\mathbf{w}$, then $\mathbf{Z}^\top \nabla f(\mathbf{x}_*) = \mathbf{0}_{n-m}$.
2. **Second-order necessary condition:** if $\mathbf{x}_*$ is a local minimizer, then $\mathbf{Z}^\top \nabla f(\mathbf{x}_*) = \mathbf{0}_{n-m}$ and $\mathbf{Z}^\top \nabla^2 f(\mathbf{x}_*) \mathbf{Z}$ is positive semidefinite.
3. **Second-order sufficiency condition:** if $\mathbf{Z}^\top \nabla f(\mathbf{x}_*) = \mathbf{0}_{n-m}$ and $\mathbf{Z}^\top \nabla^2 f(\mathbf{x}_*) \mathbf{Z}$ is symmetric positive definite, then $\mathbf{x}_*$ is a strict local minimizer on the feasible affine space.

These conditions motivate the full-space approach.

Lecture 24 — Thursday, November 06, 2014

Full Space and Range-Space Approaches

Consider the equality-constrained quadratic model

$$\min \left\{ \mathbf{g}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{H} \mathbf{x} \mid \mathbf{A} \mathbf{x} = \mathbf{b} \right\},$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$.

In the **full-space approach**, we keep both primal and dual variables. First-order stationarity implies that there exists $\boldsymbol{\lambda} \in \mathbb{R}^m$ such that

$$\mathbf{H} \mathbf{x} + \mathbf{A}^\top \boldsymbol{\lambda} = -\mathbf{g}.$$

Together with feasibility, this gives the system

$$\begin{pmatrix} \mathbf{H} & \mathbf{A}^\top \\ \mathbf{A} & \mathbf{0}_{m \times m} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\mathbf{g} \\ \mathbf{b} \end{pmatrix}.$$

This formulation is direct, but the linear system has size $(n + m)$, so a dense solve costs about $O((n + m)^3)$. Also, a solution is a stationary point and may fail to be a minimizer if second-order conditions do not hold.

In the **range-space approach**, assume $\mathbf{H}$ is symmetric positive definite and $\mathbf{A}$ has full row rank, and eliminate $\mathbf{x}$. From

$$\mathbf{x} = -\mathbf{H}^{-1}(\mathbf{g} + \mathbf{A}^\top \boldsymbol{\lambda}),$$

substitution into $\mathbf{Ax} = \mathbf{b}$ yields the reduced multiplier system

$$(\mathbf{AH}^{-1}\mathbf{A}^\top)\boldsymbol{\lambda} = -\mathbf{b} - \mathbf{AH}^{-1}\mathbf{g}.$$

Then

$$\mathbf{x} = -\mathbf{H}^{-1}\mathbf{g} - \mathbf{H}^{-1}\mathbf{A}^\top \boldsymbol{\lambda}.$$

The full-row-rank assumption makes $\mathbf{AH}^{-1}\mathbf{A}^\top$ symmetric positive definite and hence invertible. Because the reduced system is only $m \times m$, this approach is attractive when $m \ll n$.

A practical implementation is:

1. Compute the Cholesky factorization $\mathbf{H} = \mathbf{LL}^\top$ and define $\tilde{\mathbf{A}} = \mathbf{AL}^{-\top}$.
2. Solve $\mathbf{Lu} = -\mathbf{g}$ and $\mathbf{L}^\top \mathbf{v} = \mathbf{u}$, so $\mathbf{v} = -\mathbf{H}^{-1}\mathbf{g}$.
3. Form $\tilde{\mathbf{b}} = \mathbf{Av} - \mathbf{b}$.
4. Solve $(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top)\mathbf{w} = \tilde{\mathbf{b}}$.
5. Solve $\mathbf{Ly} = \mathbf{A}^\top \mathbf{w}$ and $\mathbf{L}^\top \mathbf{t} = \mathbf{y}$.
6. Set $\mathbf{x} = \mathbf{v} - \mathbf{t}$.

Lecture 25 — Friday, November 07, 2014

Nonlinear Objective Function with Linear Equalities

Unconstrained minimization ideas can be adapted to linear equality-constrained problems. In particular, the subspace trust region framework extends naturally. The key object is the reduced Hessian $\mathbf{Z}^\top \mathbf{HZ}$. Near a local constrained minimizer satisfying the second-order sufficient condition, this matrix remains positive definite, and the same methods used for unconstrained problems can

be applied to compute constrained Newton steps.

At feasible points far from a local minimizer, the reduced Hessian may be indefinite. In that setting, a trust-region strategy is still effective and can be adapted directly to the constrained case. General-purpose constrained solvers, including MATLAB's `fmincon`, provide numerical implementations of related methods.

Recall Eva's portfolio model:

$$\min_{\mathbf{x} \in \mathbb{R}^{12}} \|\mathbf{A}_1 \mathbf{x}\|_2^2 \quad \text{subject to} \quad \mathbf{A}_2 \mathbf{x} = \mathbf{b},$$

where $\mathbf{A}_1 = \bar{\mathbf{R}} - \mathbf{R}$,

$$\mathbf{A}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ \bar{r}_1 & \bar{r}_2 & \bar{r}_3 & \bar{r}_4 & \bar{r}_5 & \bar{r}_6 & \bar{r}_7 & \bar{r}_8 & \bar{r}_9 & \bar{r}_{10} & \bar{r}_{11} & \bar{r}_{12} \end{pmatrix} \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 1 \\ 0.25 \\ 0.20 \\ r_p \end{pmatrix}.$$

Assume $\mathbf{A}_2$ has full row rank. To parameterize all feasible points, compute a QR factorization of $\mathbf{A}_2^\top$:

$$\mathbf{A}_2^\top = \mathbf{Q}_2 \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{0}_{8 \times 4} \end{pmatrix} = (\mathbf{Y}_2, \mathbf{Z}_2) \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{0}_{8 \times 4} \end{pmatrix} = \mathbf{Y}_2 \mathbf{R}_2,$$

where $\mathbf{Y}_2 \in \mathbb{R}^{12 \times 4}$ and $\mathbf{Z}_2 \in \mathbb{R}^{12 \times 8}$. Then all solutions of $\mathbf{A}_2 \mathbf{x} = \mathbf{b}$ are

$$\mathbf{x} = \mathbf{x}_0 + \mathbf{Z}_2 \mathbf{w}, \quad \mathbf{w} \in \mathbb{R}^8, \quad \text{and} \quad \mathbf{x}_0 = \mathbf{Y}_2 \mathbf{R}_2^{-\top} \mathbf{b}.$$

Substituting into the objective gives an unconstrained least-squares problem:

$$\min_{\mathbf{w} \in \mathbb{R}^8} \|\mathbf{A}_1(\mathbf{x}_0 + \mathbf{Z}_2 \mathbf{w})\|_2^2 = \min_{\mathbf{w} \in \mathbb{R}^8} \|\bar{\mathbf{b}} + \bar{\mathbf{A}} \mathbf{w}\|_2^2,$$

where

$$\bar{\mathbf{A}} = \mathbf{A}_1 \mathbf{Z}_2 \in \mathbb{R}^{100 \times 8} \quad \text{and} \quad \bar{\mathbf{b}} = \mathbf{A}_1 \mathbf{x}_0 \in \mathbb{R}^{100}.$$

If $\bar{\mathbf{A}}$ has full column rank, solve this overdetermined system by QR:

$$\bar{\mathbf{A}} = \mathbf{Q} \begin{pmatrix} \bar{\mathbf{R}} \\ \mathbf{0}_{92 \times 8} \end{pmatrix} = (\mathbf{Y}, \mathbf{Z}) \begin{pmatrix} \bar{\mathbf{R}} \\ \mathbf{0}_{92 \times 8} \end{pmatrix} = \mathbf{Y} \bar{\mathbf{R}},$$

so

$$\mathbf{v}_w = -\bar{\mathbf{R}}^{-1} \mathbf{Y}^\top \bar{\mathbf{b}}.$$

Hence Eva's unique optimal strategy is

$$\mathbf{x}_* = \mathbf{x}_0 + \mathbf{Z}_2 \mathbf{v}_w = \mathbf{Y}_2 \mathbf{R}_2^{-\top} \mathbf{b} - \mathbf{Z}_2 \bar{\mathbf{R}}^{-1} \mathbf{Y}^\top \bar{\mathbf{b}}.$$

If either rank condition fails, a rank-revealing QR factorization or singular-value decomposition is required; in particular, the least-squares minimizer need not be unique.

The computation can be summarized as follows:

1. Form $\mathbf{A}_1 = \bar{\mathbf{R}} - \mathbf{R} \in \mathbb{R}^{100 \times 12}$.
2. Form $\mathbf{A}_2$ and $\mathbf{b} = (1 \quad 0.25 \quad 0.20 \quad r_p)^\top$.
3. Compute $\mathbf{A}_2^\top = \mathbf{Q}_2 \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{0}_{8 \times 4} \end{pmatrix} = (\mathbf{Y}_2, \mathbf{Z}_2) \begin{pmatrix} \mathbf{R}_2 \\ \mathbf{0}_{8 \times 4} \end{pmatrix} = \mathbf{Y}_2 \mathbf{R}_2$.
4. Compute $\mathbf{x}_0 = \mathbf{Y}_2 \mathbf{R}_2^{-\top} \mathbf{b}$.
5. Form $\bar{\mathbf{A}} = \mathbf{A}_1 \mathbf{Z}_2 \in \mathbb{R}^{100 \times 8}$ and $\bar{\mathbf{b}} = \mathbf{A}_1 \mathbf{x}_0 \in \mathbb{R}^{100}$.
6. Compute $\bar{\mathbf{A}} = \mathbf{Q} \begin{pmatrix} \bar{\mathbf{R}} \\ \mathbf{0}_{92 \times 8} \end{pmatrix} = (\mathbf{Y}, \mathbf{Z}) \begin{pmatrix} \bar{\mathbf{R}} \\ \mathbf{0}_{92 \times 8} \end{pmatrix} = \mathbf{Y} \bar{\mathbf{R}}$.
7. Compute $\mathbf{v}_w = -\bar{\mathbf{R}}^{-1} \mathbf{Y}^\top \bar{\mathbf{b}}$.
8. Compute $\mathbf{x}_* = \mathbf{x}_0 + \mathbf{Z}_2 \mathbf{v}_w = \mathbf{Y}_2 \mathbf{R}_2^{-\top} \mathbf{b} - \mathbf{Z}_2 \bar{\mathbf{R}}^{-1} \mathbf{Y}^\top \bar{\mathbf{b}}$.

Lecture 26 — Tuesday, November 11, 2014

Inequality Constraints and Interior Methods

We next allow both equality and inequality constraints:

$$\min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{B}\mathbf{x} = \mathbf{b}, \mathbf{A}\mathbf{x} \leq \mathbf{d}\}. \quad (*)$$

In Eva's investment problem, suppose we add the restrictions

$$x_5 + x_6 + x_7 \leq 10\% \quad \text{and} \quad x_8 + x_9 \geq 15\%.$$

The second inequality is equivalently $-x_8 - x_9 \leq -15\%$. These can be written as

$$\mathbf{A}_3 \mathbf{x} \leq \mathbf{d}, \quad \mathbf{A}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{d} = \begin{pmatrix} 0.10 \\ -0.15 \end{pmatrix}.$$

Theorem 26.1 Assume f is continuously differentiable, $\mathbf{x}^*$ is a local minimizer of $(*)$, and the rows of $\mathbf{B}$ together with the rows $\mathbf{A}_i$ active at $\mathbf{x}^*$ are linearly independent. Then there exist multipliers $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^*$ such that

$$\begin{aligned} \nabla f(\mathbf{x}^*) + \mathbf{A}^\top \boldsymbol{\mu}^* + \mathbf{B}^\top \boldsymbol{\lambda}^* &= \mathbf{0}_n, & \mathbf{A} \mathbf{x}^* &\leq \mathbf{d}, & \text{and} & \quad \mathbf{B} \mathbf{x}^* = \mathbf{b}, \\ \mu_i^* (\mathbf{A}_i \mathbf{x}^* - d_i) &= 0 \quad (i = 1, \dots, p) & \text{and} & \quad \boldsymbol{\mu}^* \geq \mathbf{0}_p. \end{aligned}$$

The equality-constrained condition $\nabla f(\mathbf{x}^*) + \mathbf{B}^\top \boldsymbol{\lambda}^* = \mathbf{0}_n$ is the special case with no inequalities.

What does it all mean? The complementarity condition means that each inequality is either inactive, in which case $(\mathbf{A} \mathbf{x}^* - \mathbf{d})_i < 0$ and $\mu_i^* = 0$, or active, in which case $(\mathbf{A} \mathbf{x}^* - \mathbf{d})_i = 0$ and the multiplier may be positive.

Example 26.2 For a two-variable example, consider

$$\min (x_1 - 2)^2 + 2(x_2 - 1)^2 \quad \text{subject to} \quad x_1 + 4x_2 \leq 3 \quad \text{and} \quad x_1 - x_2 \geq 0.$$

Equivalently,

$$\mathbf{A} = \begin{pmatrix} 1 & 4 \\ -1 & 1 \end{pmatrix}, \quad \mathbf{d} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{A} \mathbf{x} \leq \mathbf{d}.$$

The stationarity condition from [Theorem 26.1](#) is

$$\nabla f(\mathbf{x}) + \mathbf{A}^\top \boldsymbol{\mu} = \begin{pmatrix} 2(x_1 - 2) \\ 4(x_2 - 1) \end{pmatrix} + \begin{pmatrix} 1 & -1 \\ 4 & 1 \end{pmatrix} \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Complementarity gives

$$\mu_1 (x_1 + 4x_2 - 3) = 0, \quad \mu_2 (-x_1 + x_2) = 0, \quad \text{and} \quad \mu_1, \mu_2 \geq 0.$$

We check the possible active sets:

1. If $\mu_1 = 0$ and $\mu_2 = 0$, then $x_1 = 2$ and $x_2 = 1$. This candidate is infeasible.
2. If $\mu_1 = 0$, then $-x_1 + x_2 = 0$. Then

$$x_1 = \frac{4}{3}, \quad x_2 = \frac{4}{3}, \quad \text{and} \quad \mu_2 = -\frac{4}{3}.$$

This candidate violates the first inequality and has a negative multiplier.

3. If $\mu_2 = 0$, then $3 - x_1 - 4x_2 = 0$. Then

$$x_1 = \frac{5}{3}, \quad x_2 = \frac{1}{3}, \quad \text{and} \quad \mu_1 = \frac{2}{3}.$$

This candidate is feasible and has a nonnegative multiplier! Let's table it for now.

4. If $3 - x_1 - 4x_2 = 0$, then $x_1 - x_2 = 0$. Then

$$x_1 = x_2 = \frac{3}{5}, \quad \mu_1 = \frac{22}{25}, \quad \text{and} \quad \mu_2 = -\frac{48}{25} < 0,$$

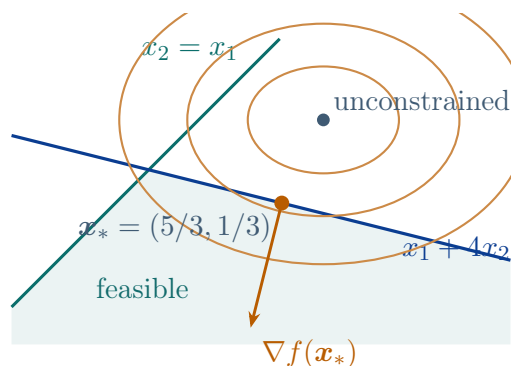
so the multiplier condition fails.

Because the objective is strictly convex and the feasible set is convex, this feasible KKT point is the unique global minimizer:

$$\mathbf{x} = \begin{pmatrix} 5/3 \\ 1/3 \end{pmatrix}.$$

The active-set geometry for this example appears in [Figure 8](#). The unconstrained minimizer lies outside the feasible wedge, while the constrained minimizer is the tangency point on $x_1 + 4x_2 = 3$; the other inequality is slack there and therefore has multiplier zero.

In two dimensions this active-set enumeration is feasible; in high dimensions it becomes prohibitively expensive. Avoid it when out doing fieldwork.



an active constraint supports the first objective contour that reaches the feasible set

FIGURE 8

Lecture 27 — Thursday, November 13, 2014

The Quadratic Problem

We focus on our beloved quadratic objective function

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top \mathbf{H} \mathbf{x} - \mathbf{c}^\top \mathbf{x}.$$

We are interested in the inequality-constrained problem $\min\{f(\mathbf{x}) \mid \mathbf{A}\mathbf{x} \leq \mathbf{d}\}$. Equality constraints $\mathbf{B}\mathbf{x} = \mathbf{b}$ can first be handled by the null-space substitution $\mathbf{x} = \mathbf{x}_0 + \mathbf{Z}\mathbf{w}$, leaving an equivalent inequality-constrained problem in $\mathbf{w}$.

The conditions from [Theorem 26.1](#) specialized to the quadratic inequality-constrained problem are

$$\begin{aligned} \mathbf{H}\mathbf{x}^* + \mathbf{A}^\top \boldsymbol{\mu}^* - \mathbf{c} &= \mathbf{0}_n & \text{and} & & \mathbf{A}\mathbf{x}^* - \mathbf{d} &\leq \mathbf{0}_p, \\ \mu_i^*(\mathbf{A}_i \mathbf{x}^* - d_i) &= 0 \quad (i = 1, \dots, p) & \text{and} & & \boldsymbol{\mu}^* &\geq \mathbf{0}_p. \end{aligned}$$

Introduce a slack variable $\mathbf{z} = \mathbf{d} - \mathbf{A}\mathbf{x}$. The conditions become

$$\begin{aligned} \mathbf{H}\mathbf{x}^* + \mathbf{A}^\top \boldsymbol{\mu}^* - \mathbf{c} &= \mathbf{0}_n \\ \mathbf{A}\mathbf{x}^* + \mathbf{z} - \mathbf{d} &= \mathbf{0}_p \\ \mathbf{Z}\boldsymbol{\mu} &= \mathbf{0}_p, \end{aligned}$$

where $\mathbf{Z} = \text{diag}(z_1, \dots, z_p)$ and $\mathbf{D}_\mu = \text{diag}(\mu_1, \dots, \mu_p)$. Combine these equations into

$$\mathbf{F}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{z}) = \begin{pmatrix} \mathbf{H}\mathbf{x} + \mathbf{A}^\top \boldsymbol{\mu} - \mathbf{c} \\ \mathbf{A}\mathbf{x} + \mathbf{z} - \mathbf{d} \\ \mathbf{Z}\mathbf{D}_\mu \mathbf{1} \end{pmatrix} = \mathbf{0}_{n+2p}.$$

We only know one way to (iteratively) solve $\mathbf{F}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{z}) = \mathbf{0}_{n+2p}$, and that is Newton's method. Generically, we have

$$\begin{pmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu}_{k+1} \\ \mathbf{z}_{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_k \\ \boldsymbol{\mu}_k \\ \mathbf{z}_k \end{pmatrix} - [\mathbf{F}'(\mathbf{x}_k, \boldsymbol{\mu}_k, \mathbf{z}_k)]^{-1} \mathbf{F}(\mathbf{x}_k, \boldsymbol{\mu}_k, \mathbf{z}_k).$$

The Jacobian is

$$\mathbf{J}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{z}) = \begin{pmatrix} \mathbf{H} & \mathbf{A}^\top & \mathbf{0}_{n \times p} \\ \mathbf{A} & \mathbf{0}_{p \times p} & \mathbf{I}_p \\ \mathbf{0}_{p \times n} & \mathbf{Z} & \mathbf{D}_\mu \end{pmatrix}.$$

The Newton step solves

$$\begin{pmatrix} \mathbf{H} & \mathbf{A}^\top & \mathbf{0}_{n \times p} \\ \mathbf{A} & \mathbf{0}_{p \times p} & \mathbf{I}_p \\ \mathbf{0}_{p \times n} & \mathbf{Z} & \mathbf{D}_\mu \end{pmatrix} \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\mu} \\ \Delta \mathbf{z} \end{pmatrix} = - \begin{pmatrix} \mathbf{H}\mathbf{x} + \mathbf{A}^\top \boldsymbol{\mu} - \mathbf{c} \\ \mathbf{A}\mathbf{x} + \mathbf{z} - \mathbf{d} \\ \mathbf{Z}\mathbf{D}_\mu \mathbf{1} \end{pmatrix} \equiv - \begin{pmatrix} \mathbf{r}_d \\ \mathbf{r}_p \\ \mathbf{r}_\mu \end{pmatrix},$$

where

$$\begin{pmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\mu} \\ \Delta \mathbf{z} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\mu}_{k+1} \\ \mathbf{z}_{k+1} \end{pmatrix} - \begin{pmatrix} \mathbf{x}_k \\ \boldsymbol{\mu}_k \\ \mathbf{z}_k \end{pmatrix}.$$

The complementarity equation $z_i \mu_i = 0$ is delicate, because we also need $z_i, \mu_i \geq 0$. This motivates the central path. The set of points $(\mathbf{x}^\tau, \boldsymbol{\mu}^\tau, \mathbf{z}^\tau)$ with $\mathbf{z}, \boldsymbol{\mu} > \mathbf{0}_p$ is called the **central path** if

$$\mathbf{F}(\mathbf{x}^\tau, \boldsymbol{\mu}^\tau, \mathbf{z}^\tau) = \begin{pmatrix} \mathbf{0}_n \\ \mathbf{0}_p \\ \tau \mathbf{1} \end{pmatrix}.$$

Under the usual convexity, strict-feasibility, and existence assumptions, accumulation points of central-path solutions as $\tau \rightarrow 0$ satisfy the KKT system for the original problem. A common choice is to set

$$\kappa = \frac{\mathbf{z}^\top \boldsymbol{\mu}}{p} \quad \text{and} \quad \tau = \sigma \kappa,$$

where $\sigma \in [0, 1]$. The quantity κ is called the **complementarity measure**.

Lecture 28 — Friday, November 14, 2014

We are trying to deal with the condition $z_i, \mu_i \geq 0$. Using the slack variable $\mathbf{z} = \mathbf{d} - \mathbf{Ax}$, the conditions are:

$$\begin{aligned}\mathbf{H}\mathbf{x}^* + \mathbf{A}^\top \boldsymbol{\mu}^* - \mathbf{c} &= \mathbf{0}_n \\ \mathbf{Ax}^* + \mathbf{z} - \mathbf{d} &= \mathbf{0}_p \\ \mu_i z_i &= 0 \quad \text{for all } i \\ \mu_i, z_i &\geq 0\end{aligned}$$

Define

$$\mathbf{F}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{z}) = \begin{pmatrix} \mathbf{H}\mathbf{x} + \mathbf{A}^\top \boldsymbol{\mu} - \mathbf{c} \\ \mathbf{Ax} + \mathbf{z} - \mathbf{d} \\ \mathbf{ZD}_\mu \mathbf{1} \end{pmatrix},$$

where $\mathbf{Z} = \text{diag}(z_1, \dots, z_p)$ and $\mathbf{D}_\mu = \text{diag}(\mu_1, \dots, \mu_p)$. Rather than solving $\mathbf{F}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{z}) = \mathbf{0}_{n+2p}$ directly, we solve the central-path equations

$$\mathbf{F}(\mathbf{x}^\tau, \boldsymbol{\mu}^\tau, \mathbf{z}^\tau) = \begin{pmatrix} \mathbf{0}_n \\ \mathbf{0}_p \\ \tau \mathbf{1} \end{pmatrix}$$

for small $\tau > 0$.

Define

$$\bar{\mathbf{F}}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{z}) = \mathbf{F}(\mathbf{x}, \boldsymbol{\mu}, \mathbf{z}) - \begin{pmatrix} \mathbf{0}_n \\ \mathbf{0}_p \\ \tau \mathbf{1} \end{pmatrix}.$$

Write

$$\bar{\mathbf{F}}(\mathbf{x}_k, \boldsymbol{\mu}_k, \mathbf{z}_k) = \begin{pmatrix} \mathbf{r}_d \\ \mathbf{r}_p \\ \mathbf{r}_\mu \end{pmatrix}.$$

To calculate the Newton step, solve

$$\begin{pmatrix} \mathbf{H} & \mathbf{A}^\top & \mathbf{0}_{n \times p} \\ \mathbf{A} & \mathbf{0}_{p \times p} & \mathbf{I}_p \\ \mathbf{0}_{p \times n} & \mathbf{Z} & \mathbf{D}_\mu \end{pmatrix} \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\mu} \\ \Delta \mathbf{z} \end{pmatrix} = - \begin{pmatrix} \mathbf{r}_d \\ \mathbf{r}_p \\ \mathbf{r}_\mu \end{pmatrix}.$$

We determine τ using the complementarity measure $\kappa = \mathbf{z}^\top \boldsymbol{\mu} / p$, choosing $\tau = \sigma \kappa$ for some

$\sigma \in [0, 1]$. If $\mathbf{D}_\mu$ is invertible, then the Newton system can be reduced to

$$\begin{pmatrix} \mathbf{H} & \mathbf{A}^\top \\ \mathbf{A} & -\mathbf{D}_\mu^{-1}\mathbf{Z} \end{pmatrix} \begin{pmatrix} \Delta \mathbf{x} \\ \Delta \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} -\mathbf{r}_d \\ -\mathbf{r}_p + \mathbf{D}_\mu^{-1}\mathbf{r}_\mu \end{pmatrix}.$$

This is a smaller system.

Box Constraints

Consider the **box-constrained** minimization problem

$$\min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}.$$

This is a very hard problem. A feasible point is **nondegenerate** if, for each index i ,

$$(\nabla f(\mathbf{x}))_i = 0 \implies l_i < (\mathbf{x})_i < u_i.$$

Theorem 28.1 (Optimality conditions: box constraints) Assume f is continuously differentiable for the first-order statement and twice continuously differentiable near $\mathbf{x}^*$ for the second-order statements.

1. **(First-order necessary condition)**: if $\mathbf{x}^*$ is a local minimizer and $\mathbf{g}^* = \nabla f(\mathbf{x}^*)$, then

$$\begin{cases} (\mathbf{g}^*)_i = 0 & \text{if } l_i < (\mathbf{x}^*)_i < u_i \\ (\mathbf{g}^*)_i \leq 0 & \text{if } (\mathbf{x}^*)_i = u_i \\ (\mathbf{g}^*)_i \geq 0 & \text{if } (\mathbf{x}^*)_i = l_i \end{cases} \quad (*)$$

2. **(Second-order necessary condition)**: let

$$\text{Free}(\mathbf{x}^*) = \{i \mid l_i < (\mathbf{x}^*)_i < u_i\}.$$

If $\mathbf{x}^*$ is a local minimizer and $\mathbf{g}^* = \nabla f(\mathbf{x}^*)$, then the Hessian on the free variables, $\mathbf{H}^{\text{free}}$, is positive semidefinite.

3. **(Second-order sufficiency condition)**: if $\mathbf{x}_*$ is nondegenerate and $(*)$ holds and $\mathbf{H}^{\text{free}}$ is positive definite, then $\mathbf{x}_*$ is a local minimizer.

Example 28.2 Let

$$\mathbf{H} = \begin{pmatrix} -10 & 2 & 1 \\ 2 & 8 & 3 \\ 1 & 3 & 16 \end{pmatrix} \quad \text{and} \quad \text{Free} = \{2, 3\}.$$

If $x_1^* \in \{l_1, u_1\}$ while $x_2^* \in (l_2, u_2)$ and $x_3^* \in (l_3, u_3)$, then

$$\mathbf{H}^{\text{free}} = \begin{pmatrix} 8 & 3 \\ 3 & 16 \end{pmatrix}.$$

This free Hessian is positive definite.

Lecture 29 — Tuesday, November 18, 2014

Example 29.1 Let

$$\mathbf{l} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \quad \text{and} \quad \mathbf{x}_* = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix}.$$

If $\nabla f(\mathbf{x}_*) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, then $\mathbf{x}_*$ is nondegenerate: the only zero gradient component is the second one, and $(\mathbf{x}_*)_2 = 1.5$ is strictly inside its interval.

With the same bounds and $\mathbf{x}_*$, if $\nabla f(\mathbf{x}_*) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, then $\mathbf{x}_*$ is degenerate, because the first gradient component is zero while $(\mathbf{x}_*)_1 = l_1$ is on the boundary.

The box-constrained minimization problem tends to be very difficult to solve. Two standard approaches are active-set methods and interior-point methods. In an **active-set approach**, set

$$\text{Free}(\mathbf{x}) = \{i \mid l_i < (\mathbf{x})_i < u_i\}.$$

Define also the active set

$$\text{Act}(\mathbf{x}) = \{i \mid (\mathbf{x})_i = l_i \text{ or } (\mathbf{x})_i = u_i\}.$$

Starting from a feasible $\mathbf{x}_0$, a typical active-set iteration is:

1. At iterate $\mathbf{x}_k$, fix the active components and solve the reduced subproblem in the free variables.
2. Compute a feasible trial point with lower objective value.
3. If the first-order box-constrained optimality conditions are satisfied, stop.
4. Otherwise, modify the working active set (usually by releasing one active bound or adding a newly hit bound) and continue.

In a simplified setting, we seek $\mathbf{x}_{k+1}$ such that

$$f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k) \quad \text{and often} \quad \text{Free}(\mathbf{x}_k) \subseteq \text{Free}(\mathbf{x}_{k+1}).$$

Example 29.2 For intuition, let $\mathbf{x} = (x_1, x_2, x_3)^\top$ represent age, height, and weight within box bounds. Suppose three feasible candidates are

$$\mathbf{x}^A = (35, 1.70, 180)^\top, \quad \mathbf{x}^B = (30, 1.70, 170)^\top, \quad \text{and} \quad \mathbf{x}^C = (25, 1.60, 160)^\top.$$

If the sets of free indices at these points are

$$\text{Free}(\mathbf{x}^A) = \{1, 3\}, \quad \text{Free}(\mathbf{x}^B) = \{3\}, \quad \text{and} \quad \text{Free}(\mathbf{x}^C) = \{1, 2, 3\},$$

then the active-set method interprets this as follows: at $\mathbf{x}^B$ two variables are on bounds, so only one variable moves freely; after releasing active bounds we can move to a point such as $\mathbf{x}^C$ where all components are free. This is the mechanism by which the method updates working sets while continuing to decrease the objective.

In an **interior-point method**, each iteration stays strictly inside the box, with $\mathbf{l} < \mathbf{x}_k < \mathbf{u}$, and the iterates approach a solution $\mathbf{x}^*$. A typical trust-region subproblem has the form

$$\min_{\mathbf{d}} \left\{ \mathbf{d}^\top \nabla f(\mathbf{x}_k) + \frac{1}{2} \mathbf{d}^\top (\mathbf{H}_k + \mathbf{C}_k) \mathbf{d} \mid \|\mathbf{D}_k^{-1} \mathbf{d}\| \leq \Delta_k \right\},$$

where $\mathbf{C}_k$ and $\mathbf{D}_k$ are diagonal matrices. Interior-point methods are very new and very exciting, and are being actively researched. Unfortunately they lie far beyond the understanding of a mere undergrad, so we will not pursue them further.

Actually, by 2014 interior-point methods were already quite mature. For problems with box constraints (which, contrary to these notes, are unusually *easy* to deal with), methods that exploit the constraints such as the L-BFGS-B algorithm are standard, and interior-point methods are

usually overkill.

To solve a quadratic program

$$\min \left\{ \mathbf{f}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \mathbf{H} \mathbf{x} \mid \mathbf{A} \mathbf{x} \leq \mathbf{b}, \mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \right\},$$

in MATLAB, use

```
[x, fval] = quadprog(H, f, A, b, Aeq, beq, l, u)
```

For the more general constrained nonlinear problem

$$\min \{ h(\mathbf{x}) \mid \mathbf{A} \mathbf{x} \leq \mathbf{b}, \mathbf{A}_{eq} \mathbf{x} = \mathbf{b}_{eq}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \},$$

use

```
[x, fval] = fmincon(fun, x0, A, b, Aeq, beq, l, u)
```

where x_0 is the initial value.

Penalty Functions

One **penalty function** approach replaces equality constraints by a penalty term:

$$\min_{\mathbf{x}} \{ f(\mathbf{x}) \mid \mathbf{A} \mathbf{x} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \} \rightsquigarrow \min \left\{ f(\mathbf{x}) + \frac{1}{2\mu} \|\mathbf{A} \mathbf{x} - \mathbf{b}\|^2 \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \right\}.$$

Taking μ small makes violations of $\mathbf{A} \mathbf{x} = \mathbf{b}$ expensive.

Lecture 30 — Thursday, November 20, 2014

Consider the box-constrained problem

$$\min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{Ax} = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}.$$

A quadratic penalty replaces the equality constraint by a term in the objective:

$$\min_{\mathbf{x}} \left\{ p(\mathbf{x}) := f(\mathbf{x}) + \frac{1}{2\mu} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \right\}.$$

The Hessian of the penalized objective is

$$\nabla^2 p(\mathbf{x}) = \nabla^2 f(\mathbf{x}) + \frac{1}{\mu} \mathbf{A}^\top \mathbf{A}.$$

For any finite $\mu > 0$, a minimizer of the penalized problem need not satisfy $\mathbf{Ax} = \mathbf{b}$ exactly. Violations become expensive as $\mu \downarrow 0$; under standard assumptions ensuring feasibility, existence, and boundedness of the relevant minimizers, their accumulation points solve the original constrained problem. The same idea applies to nonlinear equality constraints,

$$\min_{\mathbf{x}} \{f(\mathbf{x}) \mid g(\mathbf{x}) = \mathbf{b}, \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\},$$

by replacing $\|\mathbf{Ax} - \mathbf{b}\|_2^2$ with $\|g(\mathbf{x}) - \mathbf{b}\|_2^2$.

5. Portfolio Optimization Models

The Efficient Frontier

Let $S_1, \dots, S_n$ be risky assets, let μ_i be the expected return on asset S_i , and let σ_{ij} be the covariance between the returns of S_i and S_j . Set

$$\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^\top, \quad \boldsymbol{\Sigma} = (\sigma_{ij})_{i,j=1}^n, \quad \text{and} \quad \mathbf{1} = (1, \dots, 1)^\top.$$

The covariance matrix $\boldsymbol{\Sigma}$ is always symmetric positive semidefinite. For the closed-form derivations below, we assume it is positive definite. If x_i is the fraction of wealth invested in asset S_i , then the budget constraint is

$$\mathbf{1}^\top \mathbf{x} = 1.$$

The expected return and variance of the portfolio are

$$\mu_p = \boldsymbol{\mu}^\top \mathbf{x} \quad \text{and} \quad \sigma_p^2 = \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x}.$$

With positive definite $\boldsymbol{\Sigma}$, unrestricted portfolio weights, and compatible targets on the upper efficient frontier, **Markowitz's model** can be described in three corresponding ways. The **variance-efficient problem** fixes a target expected return and minimizes risk:

$$\min_{\mathbf{x}} \{ \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} \mid \boldsymbol{\mu}^\top \mathbf{x} = \mu_p, \mathbf{1}^\top \mathbf{x} = 1 \}.$$

The **expected-return-efficient problem** fixes a target risk and maximizes expected return:

$$\max_{\mathbf{x}} \{ \boldsymbol{\mu}^\top \mathbf{x} \mid \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} \leq \sigma_p^2, \mathbf{1}^\top \mathbf{x} = 1 \}.$$

At a nondegenerate point on the upper frontier, the risk bound is active. The **parameter-efficient problem** (or **hybrid problem**) combines the two objectives:

$$\min_{\mathbf{x}} \left\{ -t \boldsymbol{\mu}^\top \mathbf{x} + \frac{1}{2} \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} \mid \mathbf{1}^\top \mathbf{x} = 1 \right\},$$

where $t \geq 0$ is fixed. When $t = 0$, only variance is minimized; as t increases, expected return receives more weight.

Assume $\boldsymbol{\mu}$ is not a multiple of $\mathbf{1}$. For the parameter-efficient model, the first-order condition gives

$$\boldsymbol{\Sigma} \mathbf{x} - t \boldsymbol{\mu} = \lambda \mathbf{1}.$$

Since $\boldsymbol{\Sigma}$ is nonsingular,

$$\mathbf{x} = t \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} + \lambda \boldsymbol{\Sigma}^{-1} \mathbf{1}.$$

Using $\mathbf{1}^\top \mathbf{x} = 1$,

$$\lambda = \frac{1 - t \mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{1}}.$$

Substituting this expression for λ gives an affine parametrization of the optimal portfolio:

$$\mathbf{x} = t \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} + \frac{1 - t \mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{1}} \boldsymbol{\Sigma}^{-1} \mathbf{1} = \underbrace{\frac{\boldsymbol{\Sigma}^{-1} \mathbf{1}}{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{1}}}_{=: \mathbf{h}_0} + \underbrace{\left(\boldsymbol{\Sigma}^{-1} \boldsymbol{\mu} - \frac{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}{\mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{1}} \boldsymbol{\Sigma}^{-1} \mathbf{1} \right)}_{=: \mathbf{h}_1} t = \mathbf{h}_0 + \mathbf{h}_1 t.$$

Thus the expected return is

$$\mu_p = \boldsymbol{\mu}^\top \mathbf{x} = \boldsymbol{\mu}^\top \mathbf{h}_0 + t \boldsymbol{\mu}^\top \mathbf{h}_1 = \alpha_0 + \alpha_1 t,$$

where $\alpha_0 = \boldsymbol{\mu}^\top \mathbf{h}_0$ and $\alpha_1 = \boldsymbol{\mu}^\top \mathbf{h}_1$. The variance is

$$\sigma_p^2 = (\mathbf{h}_0 + \mathbf{h}_1 t)^\top \boldsymbol{\Sigma} (\mathbf{h}_0 + \mathbf{h}_1 t) = \mathbf{h}_0^\top \boldsymbol{\Sigma} \mathbf{h}_0 + 2t \mathbf{h}_1^\top \boldsymbol{\Sigma} \mathbf{h}_0 + t^2 \mathbf{h}_1^\top \boldsymbol{\Sigma} \mathbf{h}_1 = \beta_0 + 2\beta_1 t + \beta_2 t^2.$$

Here $\beta_0 = \mathbf{h}_0^\top \boldsymbol{\Sigma} \mathbf{h}_0$, $\beta_1 = \mathbf{h}_1^\top \boldsymbol{\Sigma} \mathbf{h}_0$, and $\beta_2 = \mathbf{h}_1^\top \boldsymbol{\Sigma} \mathbf{h}_1$. A direct calculation gives $\beta_1 = 0$ and $\beta_2 = \alpha_1$. Because $\boldsymbol{\mu}$ is not a multiple of $\mathbf{1}$, we have $\mathbf{h}_1 \neq \mathbf{0}_n$ and hence $\alpha_1 = \mathbf{h}_1^\top \boldsymbol{\Sigma} \mathbf{h}_1 > 0$. Therefore

$$\sigma_p^2 = \beta_0 + \alpha_1 t^2.$$

Since

$$t = \frac{\mu_p - \alpha_0}{\alpha_1} \quad \text{and} \quad t^2 = \frac{\sigma_p^2 - \beta_0}{\alpha_1},$$

the **efficient frontier** in mean–variance space is

$$\sigma_p^2 - \beta_0 = \frac{(\mu_p - \alpha_0)^2}{\alpha_1}.$$

The two branches of this parabola are shown in Figure 9.

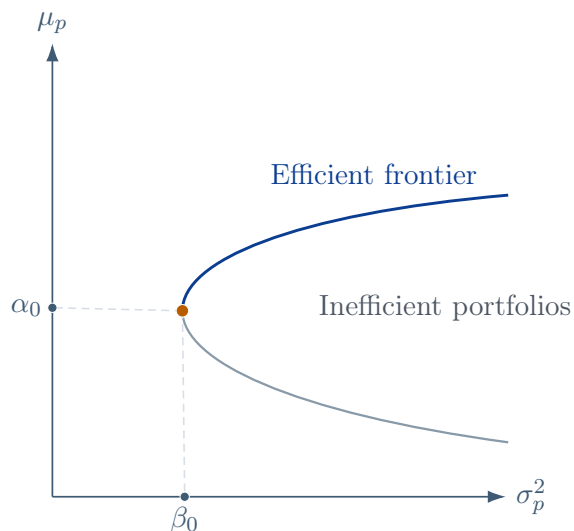


FIGURE 9

In Figure 9, the point (α_0, β_0) corresponds to the global minimum-variance portfolio. In practice, portfolio managers update their estimate of the minimum-variance portfolio every day. Those management fees aren't for nothing.

Lecture 31 — Friday, November 21, 2014

The same efficient frontier can be plotted in mean–standard deviation space. Since

$$\sigma_p = \sqrt{\beta_0 + \frac{(\mu_p - \alpha_0)^2}{\alpha_1}},$$

the upper branch may also be written as

$$\mu_p = \alpha_0 + \sqrt{\alpha_1(\sigma_p^2 - \beta_0)}.$$

Practical Portfolio Constraints

The basic Markowitz model is often extended by adding investment constraints. For example, Eva’s investment problem has the form

$$\min_{\mathbf{x}} \left\{ -t\boldsymbol{\mu}^\top \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} \mid \mathbf{A}_1 \mathbf{x} = \mathbf{b}_1 \right\}.$$

Additional equality constraints give

$$\min_{\mathbf{x}} \left\{ -t\boldsymbol{\mu}^\top \mathbf{x} + \frac{1}{2}\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x} \mid \mathbf{A}_1 \mathbf{x} = \mathbf{b}_1, \mathbf{A}_2 \mathbf{x} = \mathbf{d}_1 \right\}.$$

Closed-form solutions become less useful as these constraints become more detailed, and numerical quadratic programming methods are typically used.

Common practical constraints include the following:

1. **Box constraints:**

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}.$$

These enforce lower and upper bounds on each asset weight.

2. **Cardinality constraints:**

$$K_{\min} \leq \sum_{i=1}^n \delta_i \leq K_{\max},$$

where $\delta_i \in \{0, 1\}$ and known weight bounds $l_i \leq 0 \leq u_i$ are linked by

$$l_i \delta_i \leq x_i \leq u_i \delta_i.$$

Thus $\delta_i = 0$ forces $x_i = 0$, while $\delta_i = 1$ permits the asset's weight to lie within its bounds. These constraints bound the number of assets held. For instance, among five technology stocks, we might require holding at least two and at most three.

3. Trade count constraints:

$$T_{\min} \leq \sum_{i=1}^n \sigma_i \leq T_{\max},$$

where $\sigma_i \in \{0, 1\}$ and known trade bounds $L_i \leq 0 \leq U_i$ satisfy

$$L_i \sigma_i \leq x_i - (\mathbf{x}_0)_i \leq U_i \sigma_i.$$

Thus $\sigma_i = 0$ forbids a trade in asset i , while $\sigma_i = 1$ permits one within the stated bounds. These constraints bound the number of positions changed from an existing portfolio $\mathbf{x}_0$.

- 4. Pair trading constraints:** For each asset index i , the variables $(\mathbf{x}_0)_i$ and x_i are treated as a paired state: $(\mathbf{x}_0)_i$ is the pre-trade holding and x_i is the post-trade holding. Constraints on these pairs can restrict allowed transitions (for example, no trade, increase, or decrease), which gives direct control over how each position is adjusted during rebalancing.

Capital Asset Pricing Model (CAPM)

Now suppose that, in addition to the n risky assets, there is one risk-free asset with return r . Let x_{n+1} be the fraction of wealth invested in the risk-free asset. Since the risk-free asset has zero variance and zero covariance with the risky assets,

$$\mu_p = \boldsymbol{\mu}^\top \mathbf{x} + r x_{n+1} \quad \text{and} \quad \sigma_p^2 = \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times n} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} = \mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x}.$$

Write $\mathbf{a} = \boldsymbol{\mu} - r\mathbf{1}$, and assume $\mathbf{a} \neq \mathbf{0}_n$ and

$$D := \mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{a} > 0.$$

The second assumption ensures that the upper capital market line meets the fully invested risky-asset frontier at a positive value of the parameter t . The parameter-efficient problem becomes

$$\min_{\mathbf{x}, x_{n+1}} \left\{ -t(\boldsymbol{\mu}^\top \mathbf{x} + r x_{n+1}) + \frac{1}{2} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times n} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} \mid \mathbf{1}^\top \mathbf{x} + x_{n+1} = 1 \right\}.$$

The first-order condition is

$$\begin{pmatrix} \Sigma \mathbf{x} - t\boldsymbol{\mu} \\ -tr \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{1} \\ 1 \end{pmatrix}.$$

Thus $\lambda = -tr$ and

$$\Sigma \mathbf{x} = t(\boldsymbol{\mu} - r\mathbf{1}) \quad \text{and} \quad \mathbf{x} = t\Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1}).$$

The budget constraint gives

$$x_{n+1} = 1 - \mathbf{1}^\top \mathbf{x} = 1 - t\mathbf{1}^\top \Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1}).$$

Consequently,

$$\mu_p - r = t(\boldsymbol{\mu} - r\mathbf{1})^\top \Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1}) \quad \text{and} \quad \sigma_p^2 = t^2(\boldsymbol{\mu} - r\mathbf{1})^\top \Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1}).$$

Eliminating t yields the **capital market line**:

$$\mu_p - r = \sigma_p \left[(\boldsymbol{\mu} - r\mathbf{1})^\top \Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1}) \right]^{1/2}.$$

The **fully invested tangency portfolio** is the point on this line with no wealth in the risk-free asset. Under the CAPM equilibrium interpretation it is identified with the aggregate market portfolio, so we denote it by $\mathbf{x}_m$. It occurs when

$$t_m = \frac{1}{\mathbf{1}^\top \Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1})},$$

so that

$$\mathbf{x}_m = t_m \Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1}) = \frac{\Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1})}{\mathbf{1}^\top \Sigma^{-1}(\boldsymbol{\mu} - r\mathbf{1})}.$$

Let $\mu_m = \boldsymbol{\mu}^\top \mathbf{x}_m$ and $\sigma_m^2 = \mathbf{x}_m^\top \Sigma \mathbf{x}_m$. Since $\mathbf{1}^\top \mathbf{x}_m = 1$,

$$\sigma_m^2 = \mathbf{x}_m^\top \Sigma \mathbf{x}_m = t_m(\boldsymbol{\mu} - r\mathbf{1})^\top \mathbf{x}_m = t_m(\mu_m - r).$$

It follows that the capital market line can be written as

$$\mu_p = r + \left(\frac{\mu_m - r}{\sigma_m} \right) \sigma_p.$$

Lecture 32 — Tuesday, November 25, 2014

The CAPM efficient frontier with one risk-free asset is described by

$$\min_{\mathbf{x}, x_{n+1}} \left\{ -t \begin{pmatrix} \boldsymbol{\mu}^\top & r \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix}^\top \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times n} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ x_{n+1} \end{pmatrix} \mid \mathbf{1}^\top \mathbf{x} + x_{n+1} = 1 \right\}.$$

As derived previously,

$$\mu_p - r = \sigma_p \left[(\boldsymbol{\mu} - r\mathbf{1})^\top \boldsymbol{\Sigma}^{-1} (\boldsymbol{\mu} - r\mathbf{1}) \right]^{1/2}$$

or, equivalently,

$$\mu_p = r + \left(\frac{\mu_m - r}{\sigma_m} \right) \sigma_p = r + \frac{\sigma_p}{\sigma_m} (\mu_m - r).$$

Figure 10 compares the capital market line with the efficient frontier for risky assets.

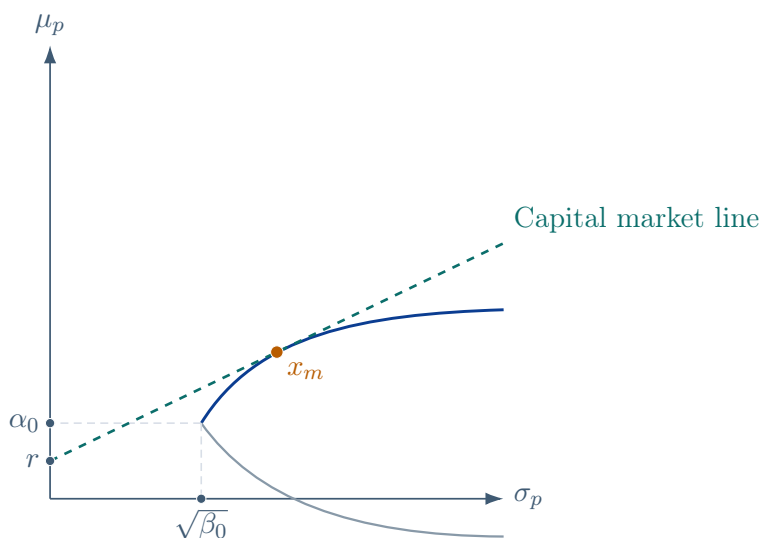


FIGURE 10

At the market portfolio x_m in Figure 10, we have $x_{n+1} = 0$. If short selling of the risk-free asset is prohibited, then $x_{n+1} \geq 0$, so the efficient frontier consists of two pieces: the capital market line from the risk-free asset to x_m , followed by the efficient frontier for risky assets from x_m onward. This combined frontier is continuous and differentiable at x_m .

Note that with a risk-free asset and unrestricted borrowing or lending at the risk-free rate, the efficient frontier in mean–standard deviation space is the capital market line. If borrowing is

restricted by $x_{n+1} \geq 0$, then the line stops at the market portfolio.

In the form

$$\mu_p = r + \frac{\sigma_p}{\sigma_m}(\mu_m - r),$$

r is the risk-free return, $\mu_m - r$ is the **market premium**, and σ_p/σ_m is the portfolio's risk relative to the market portfolio. This ratio scales the market premium. For example, if the market return is 3%, the risk-free return is 1%, and $\sigma_p/\sigma_m = 1.5$, then

$$\mu_p = 1 + 1.5(3 - 1) = 4\%.$$

The Sharpe Ratio

The slope of the capital market line is the **Sharpe ratio**

$$R = \frac{\mu_p - r}{\sigma_p},$$

which measures excess expected return per unit of standard deviation. For the efficient frontier for risky assets,

$$\mu_p = \alpha_0 + \sqrt{\alpha_1(\sigma_p^2 - \beta_0)}.$$

Hence

$$\frac{\partial \mu_p}{\partial \sigma_p} = \frac{\alpha_1 \sigma_p}{\sqrt{\alpha_1(\sigma_p^2 - \beta_0)}}.$$

The tangent line to the frontier at the market-portfolio point (σ_m, μ_m) is

$$\mu_p = \mu_m + \frac{\alpha_1 \sigma_m}{\sqrt{\alpha_1(\sigma_m^2 - \beta_0)}}(\sigma_p - \sigma_m).$$

The risk-free return is the intercept of this tangent line at $\sigma_p = 0$. Since

$$\mu_m - \alpha_0 = \sqrt{\alpha_1(\sigma_m^2 - \beta_0)},$$

we obtain

$$r = \alpha_0 - \frac{\alpha_1 \beta_0}{\mu_m - \alpha_0}.$$

Conversely, given $r < \alpha_0$,

$$\mu_m = \alpha_0 + \frac{\alpha_1 \beta_0}{\alpha_0 - r}.$$

Given a positive slope $R > \sqrt{\alpha_1}$ of the capital market line, we can equivalently recover

$$\mu_m = \alpha_0 + \frac{\alpha_1 \sqrt{\beta_0}}{\sqrt{R^2 - \alpha_1}} \quad \text{and} \quad r = \alpha_0 - \sqrt{\beta_0(R^2 - \alpha_1)}.$$

Thus the risk-free return, the market portfolio, and the slope of the capital market line determine one another.

For a fixed risk-free return r , maximizing the Sharpe ratio means solving

$$\max_{\mathbf{x}} \left\{ \frac{\boldsymbol{\mu}^\top \mathbf{x} - r}{(\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x})^{1/2}} \mid \mathbf{1}^\top \mathbf{x} = 1 \right\}.$$

Theorem 32.1 Assume $\boldsymbol{\Sigma}$ is positive definite, $\mathbf{a} = \boldsymbol{\mu} - r\mathbf{1} \neq \mathbf{0}_n$, and $D = \mathbf{1}^\top \boldsymbol{\Sigma}^{-1} \mathbf{a} > 0$. Then the unique fully invested portfolio that maximizes the Sharpe ratio is

$$\mathbf{x}_m = \frac{\boldsymbol{\Sigma}^{-1} \mathbf{a}}{D}.$$

It is also the unique solution of the parameter-efficient Markowitz problem with $t = t_m = 1/D$.

Proof. For every $\mathbf{x}$ with $\mathbf{1}^\top \mathbf{x} = 1$, the Cauchy–Schwarz inequality in the $\boldsymbol{\Sigma}$ inner product gives

$$(\mathbf{a}^\top \mathbf{x})^2 = ((\boldsymbol{\Sigma}^{-1/2} \mathbf{a})^\top (\boldsymbol{\Sigma}^{1/2} \mathbf{x}))^2 \leq (\mathbf{a}^\top \boldsymbol{\Sigma}^{-1} \mathbf{a})(\mathbf{x}^\top \boldsymbol{\Sigma} \mathbf{x}).$$

Consequently, every feasible portfolio has Sharpe ratio at most $\sqrt{\mathbf{a}^\top \boldsymbol{\Sigma}^{-1} \mathbf{a}}$. Equality with a positive numerator holds precisely when $\mathbf{x}$ is a positive multiple of $\boldsymbol{\Sigma}^{-1} \mathbf{a}$. The budget constraint forces that multiple to be $1/D$, so $\mathbf{x}_m$ is the unique maximizer.

Finally, with $t_m = 1/D$,

$$\boldsymbol{\Sigma} \mathbf{x}_m = t_m(\boldsymbol{\mu} - r\mathbf{1}) = t_m \boldsymbol{\mu} - t_m r \mathbf{1}.$$

This is the first-order condition for the parameter-efficient Markowitz problem, with multiplier $-t_m r$. Its objective is strictly convex because $\boldsymbol{\Sigma}$ is positive definite, so the feasible stationary point $\mathbf{x}_m$ is its unique global minimizer. $\square$

Appendix: Lecture and Tutorial Index

1. Lecture 1 — Tuesday, September 09, 2014
2. Lecture 2 — Thursday, September 11, 2014
3. Lecture 3 — Friday, September 12, 2014
4. Lecture 4 — Tuesday, September 16, 2014
5. Lecture 5 — Thursday, September 18, 2014
6. Lecture 6 — Friday, September 19, 2014
7. Lecture 7 — Tuesday, September 23, 2014
8. Lecture 8 — Thursday, September 25, 2014
9. Lecture 9 — Friday, September 26, 2014
10. Lecture 10 — Tuesday, September 30, 2014
11. Lecture 11 — Thursday, October 02, 2014
12. Lecture 12 — Friday, October 03, 2014
13. Lecture 13 — Tuesday, October 07, 2014
14. Lecture 14 — Thursday, October 09, 2014
15. Lecture 15 — Friday, October 10, 2014
16. Lecture 16 — Tuesday, October 14, 2014
17. Lecture 19 — Tuesday, October 21, 2014
18. Lecture 20 — Tuesday, October 28, 2014
19. Lecture 21 — Thursday, October 30, 2014
20. Lecture 22 — Friday, October 31, 2014
21. Lecture 23 — Tuesday, November 04, 2014
22. Lecture 24 — Thursday, November 06, 2014
23. Lecture 25 — Friday, November 07, 2014
24. Lecture 26 — Tuesday, November 11, 2014
25. Lecture 27 — Thursday, November 13, 2014
26. Lecture 28 — Friday, November 14, 2014

- 27. Lecture 29 — Tuesday, November 18, 2014
- 28. Lecture 30 — Thursday, November 20, 2014
- 29. Lecture 31 — Friday, November 21, 2014
- 30. Lecture 32 — Tuesday, November 25, 2014