12 Lecture 12, Feb 11

Announcements

- HW3 due Tue Feb 16 @ 11:59PM.
- HW4 posted. Due Tue Feb 23 @ 11:59PM.
- Quiz 2 returned (8.0 ± 2.1). Q1 1pt, Q2 1pt, Q3 1pt, Q4 1pt, Q5 2pt, Q6 4pt.
- HW2 graded. Feedback:
  - Cheating is not tolerated. From syllabus, "... *giving or receiving answers or code to or from another student is cheating* ..." First time, homework score is set to zero. Second time, disciplinary action.
  - Please check the solution sketch to make sure you learn something. `http://hua-zhou.github.io/teaching/biostatm280-2016winter/schedule.html`

Last time

- Applications of eigen-decomposition and SVD.
- Power algorithm and *QR iteration* for top eigen-pairs.
- *QR algorithm* for symmetric eigen-decomposition.
- Golub-Kahan-Reinsch algorithm SVD.

Today

- Iterative methods for eigen-problem and SVD.
- Jacobi algorithm for eigen-decomposition (parallel computing).
- Misc. topics: generalized eigen-problem, variants of least squares.
- Concluding remarks of numerical linear algebra.
- Optimization: overview.

Lanczos/Arnoldi iterative method for top eigen-pairs

- Motivation
  - Consider the Google PageRank problem. We want to find the top left eigenvector of the transition matrix $\boldsymbol{P}$. Direct methods such as (unsymmetric) QR or SVD takes forever. Iterative methods such as power method is feasible. However power method may take a huge number of iterations.
  - Consider adjusting for confounding by PCA in modern GWAS (genome-wide association studies). We want to find the top singular values/vectors of a genotype matrix $\boldsymbol{X} \in \mathbb{R}^{n \times p}$, where $n \sim 10^3$ and $p \sim 10^6$.
- *Krylov subspace methods* are the state-of-art iterative method for obtaining the top eigen-values/vectors or singular values/vectors of large *sparse* or *structured* matrices.
- Lanczos method: top eigen-pairs of a large *symmetric* matrix.
- Arnoldi method: top eigen-pairs of a large *asymmetric* matrix.
- Both methods are also adapted to obtain top singular values/vectors of large sparse or structured matrices.
- We will give an overview of these methods together with the conjugate gradient method for solving large linear system.
- `eigs()` and `svds()` in `Matlab` and `Julia` are wrappers of the `ARPACK` package, which implements Lanczos and Arnoldi methods. In R, try to construct sparse matrix using the `Matrix` package (by Doug Bates) and try the `irlba` package. `http://cran.r-project.org/web/packages/irlba/index.html`

Jacobi method for symmetric eigen-decomposition (KL 8.2)

Assume $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric and we seek the eigen-decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$.

- Idea: Systematically reduce off-diagonal entries

$$\text{off}(\boldsymbol{A}) = \sum_i \sum_{j \neq i} a_{ij}^2$$

  by Jacobi rotations.

- Jacobi/Givens rotations:

$$\boldsymbol{J}(p,q,\theta) = \begin{pmatrix} 1 & & 0 & & 0 & & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & & \cos(\theta) & & \sin(\theta) & & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & & -\sin(\theta) & & \cos(\theta) & & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & & 0 & & 0 & & 1 \end{pmatrix},$$

  $\boldsymbol{J}(p,q,\theta)$ is orthogonal.

- Consider $\boldsymbol{B} = \boldsymbol{J}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{J}$. $\boldsymbol{B}$ preserves the symmetry and eigenvalues of $\boldsymbol{A}$.

  Taking

$$\begin{cases} \tan(2\theta) = 2a_{pq}/(a_{qq} - a_{pp}) & \text{if } a_{pp} \neq a_{qq} \\ \theta = \pi/4 & \text{if } a_{pp} = a_{qq} \end{cases}$$

  forces $b_{pq} = 0$.

- Since orthogonal transform preserves Frobenius norm, we have

$$b_{pp}^2 + b_{qq}^2 = a_{pp}^2 + a_{qq}^2 + 2a_{pq}^2.$$

  (Just check the 2-by-2 block)

- Since $\|\boldsymbol{A}\|_{\mathrm{F}} = \|\boldsymbol{B}\|_{\mathrm{F}}$, this implies that the off-diagonal part

$$\mathrm{off}(\boldsymbol{B}) = \mathrm{off}(\boldsymbol{A}) - 2a_{pq}^2$$

  is decreased whenever $a_{pq} \neq 0$.

- One Jacobi rotation costs $O(n)$ flops.

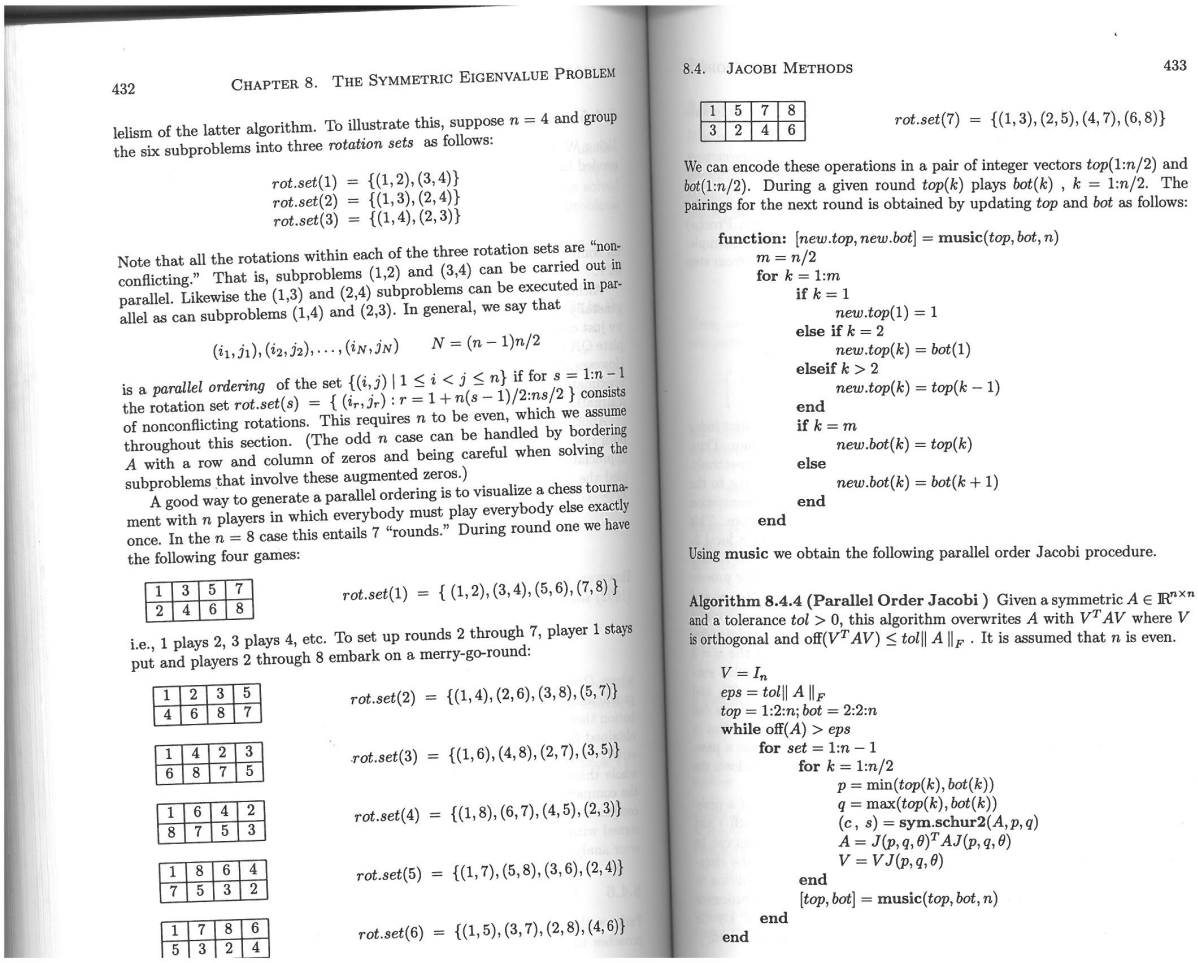
- *Classical Jacobi*: search for the largest $|a_{ij}|$ at each iteration.

- $\mathrm{off}(\boldsymbol{A}) \leq n(n-1)a_{ij}^2$ and $\mathrm{off}(\boldsymbol{B}) = \mathrm{off}(\boldsymbol{A}) - 2a_{ij}^2$ together implies

$$\mathrm{off}(\boldsymbol{B}) \leq \left(1 - \frac{2}{n(n-1)}\right) \mathrm{off}(\boldsymbol{A}).$$

  So Jacobi method converges in $O(n^2)$ iterations.

- In practice, cyclic-by-row implementation, to avoid the costly $O(n^2)$ search in the classical Jacobi.
- Jacobi method attracts a lot recent attention because of its rich inherent parallelism.
- *Parallel Jacobi*: "merry-go-round" to generate parallel ordering.



lelism of the latter algorithm. To illustrate this, suppose $n = 4$ and group the six subproblems into three *rotation sets* as follows:

$$\begin{aligned} rot.set(1) &= \{(1,2),(3,4)\} \\ rot.set(2) &= \{(1,3),(2,4)\} \\ rot.set(3) &= \{(1,4),(2,3)\} \end{aligned}$$

Note that all the rotations within each of the three rotation sets are "nonconflicting." That is, subproblems (1,2) and (3,4) can be carried out in parallel. Likewise the (1,3) and (2,4) subproblems can be executed in parallel as can subproblems (1,4) and (2,3). In general, we say that

$$(i_1,j_1),(i_2,j_2),\ldots,(i_N,j_N) \qquad N = (n-1)n/2$$

is a *parallel ordering* of the set $\{(i,j) \mid 1 \le i < j \le n\}$ if for $s = 1{:}n-1$ the rotation set $rot.set(s) = \{\,(i_r,j_r) : r = 1+n(s-1)/2{:}ns/2\,\}$ consists of nonconflicting rotations. This requires n to be even, which we assume throughout this section. (The odd n case can be handled by bordering A with a row and column of zeros and being careful when solving the subproblems that involve these augmented zeros.)

A good way to generate a parallel ordering is to visualize a chess tournament with n players in which everybody must play everybody else exactly once. In the $n = 8$ case this entails 7 "rounds." During round one we have the following four games:

1	3	5	7
2	4	6	8

$rot.set(1) = \{\,(1,2),(3,4),(5,6),(7,8)\,\}$

i.e., 1 plays 2, 3 plays 4, etc. To set up rounds 2 through 7, player 1 stays put and players 2 through 8 embark on a merry-go-round:

1	2	3	5
4	6	8	7

$rot.set(2) = \{(1,4),(2,6),(3,8),(5,7)\}$

1	4	2	3
6	8	7	5

$rot.set(3) = \{(1,6),(4,8),(2,7),(3,5)\}$

1	6	4	2
8	7	5	3

$rot.set(4) = \{(1,8),(6,7),(4,5),(2,3)\}$

1	8	6	4
7	5	3	2

$rot.set(5) = \{(1,7),(5,8),(3,6),(2,4)\}$

1	7	8	6
5	3	2	4

$rot.set(6) = \{(1,5),(3,7),(2,8),(4,6)\}$

1	5	7	8
3	2	4	6

$rot.set(7) = \{(1,3),(2,5),(4,7),(6,8)\}$

We can encode these operations in a pair of integer vectors $top(1{:}n/2)$ and $bot(1{:}n/2)$. During a given round $top(k)$ plays $bot(k)$, $k = 1{:}n/2$. The pairings for the next round is obtained by updating top and bot as follows:

```
function: [new.top, new.bot] = music(top, bot, n)
    m = n/2
    for k = 1:m
        if k = 1
            new.top(1) = 1
        else if k = 2
            new.top(k) = bot(1)
        elseif k > 2
            new.top(k) = top(k − 1)
        end
        if k = m
            new.bot(k) = top(k)
        else
            new.bot(k) = bot(k + 1)
        end
    end
```

Using **music** we obtain the following parallel order Jacobi procedure.

Algorithm 8.4.4 (Parallel Order Jacobi) Given a symmetric $A \in \mathbb{R}^{n\times n}$ and a tolerance $tol > 0$, this algorithm overwrites A with V^TAV where V is orthogonal and $\text{off}(V^TAV) \le tol \| A \|_F$. It is assumed that n is even.

```
V = I_n
eps = tol‖ A ‖_F
top = 1:2:n; bot = 2:2:n
while off(A) > eps
    for set = 1:n − 1
        for k = 1:n/2
            p = min(top(k), bot(k))
            q = max(top(k), bot(k))
            (c , s) = sym.schur2(A, p, q)
            A = J(p, q, θ)^T AJ(p, q, θ)
            V = VJ(p, q, θ)
        end
        [top, bot] = music(top, bot, n)
    end
end
```

Generalized eigen-problem

- Generalized eigen-problem: $\boldsymbol{Ax} = \lambda \boldsymbol{Bx}$, where $\boldsymbol{A}$ psd and $\boldsymbol{B}$ pd.
- Applications: canonical correlation analysis (CCA), partial least squares (PLS), sliced inverse regression (SIR).

- Method 1: $\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x}$. Non-symmetric eigen-problem ☹.
- Method 2: Cholesky $\boldsymbol{B} = \boldsymbol{L}\boldsymbol{L}^\intercal$. Then $\boldsymbol{L}^{-1}\boldsymbol{A}\boldsymbol{L}^{-T}\boldsymbol{y} = \lambda\boldsymbol{y}$ where $\boldsymbol{y} = \boldsymbol{L}^\intercal\boldsymbol{x}$.
- Method 3 (most numerically stable, $\boldsymbol{B}$ can be rank deficient): QZ algorithm.
- `eig()` and `qz()` in `Matlab` and `Julia` implement QZ. No native function in `R`? Check the `geneig` package. `https://cran.r-project.org/web/packages/geigen/geigen.pdf`

Generalized singular value decomposition

- $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{p\times n}$. Then there exists orthogonal $\boldsymbol{U} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{p\times p}$ and an invertible $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} \boldsymbol{U}^T\boldsymbol{A}\boldsymbol{X} &= \boldsymbol{C} = \text{diag}(c_1, \ldots, c_n), \quad c_i \geq 0 \\ \boldsymbol{V}^T\boldsymbol{B}\boldsymbol{X} &= \boldsymbol{S} = \text{diag}(s_1, \ldots, s_q), \quad s_i \geq 0, \end{aligned}$$

  where $q = \min\{p, n\}$.
- Applications: quadratically inequality-constrained least squares problem (LSQI).
- `gsvd()` in `Matlab` implements generalized SVD. No native function in `R`?

In the zoo of least squares (self-study)

Weighted least squares

- In weighted least squares, we minimize $\sum_{i=1}^n w_i(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2$, where $w_i > 0$ are observation weights.
- Let $\boldsymbol{W} = \text{diag}(w_1, \ldots, w_n)$. Then the criterion is $\|\boldsymbol{W}^{1/2}\boldsymbol{y} - \boldsymbol{W}^{1/2}\boldsymbol{X}\boldsymbol{\beta}\|_2^2$, which can be solved by standard methods for least squares with $\tilde{\boldsymbol{y}} = \boldsymbol{W}^{1/2}\boldsymbol{y}$ and $\tilde{\boldsymbol{X}} = \boldsymbol{W}^{1/2}\boldsymbol{X}$.

General least squares

- In Aitken model: $E(\boldsymbol{y}) = \boldsymbol{X}\boldsymbol{\beta}$, $\text{Cov}(\boldsymbol{y}) = \sigma^2\boldsymbol{V}$, where $\boldsymbol{V}$ is a positive semidefinite matrix. We minimize the generalized least squares criterion

$$(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T\boldsymbol{M}^-(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}),$$

where $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ is some positive semidefinite matrix, e.g., $\boldsymbol{M} = \boldsymbol{V}$ for non-singular $\boldsymbol{V}$ or $\boldsymbol{M} = \boldsymbol{V} + \boldsymbol{X}\boldsymbol{X}^T$ for singular $\boldsymbol{V}$.

- Let $\boldsymbol{M} = \boldsymbol{B}\boldsymbol{B}^T$ for some $\boldsymbol{B} \in \mathbb{R}^{n \times n}$ (e.g., the Cholesky factor). One approach is to minimize

$$\|\boldsymbol{B}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})\|_2^2.$$

  Unfortunately, when $\boldsymbol{B}$ is poorly conditioned (or even not invertible), the procedure produces a poor solution.

- Paige's method. The generalized least squares problem is equivalent to

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{v}^T\boldsymbol{v} \\ \text{subject to} \quad & \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{B}\boldsymbol{v} = \boldsymbol{y}. \end{aligned}$$

  To solve this problem, first compute the QR of $\boldsymbol{X}$

$$\boldsymbol{X} = (\boldsymbol{Q}_1, \boldsymbol{Q}_2) \begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{pmatrix}.$$

  Compute another QR for the (flat) matrix $\boldsymbol{Q}_2^T\boldsymbol{B}$ such that

$$\boldsymbol{Q}_2^T\boldsymbol{B} = (\boldsymbol{0}, \boldsymbol{S}) \begin{pmatrix} \boldsymbol{Z}_1^T \\ \boldsymbol{Z}_2^T \end{pmatrix},$$

  where $\boldsymbol{S}$ is upper triangular and $(\boldsymbol{Z}_1, \boldsymbol{Z}_2) \in \mathbb{R}^{n \times n}$ is orthogonal. Then the constraint becomes

$$\begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{0} \end{pmatrix} \boldsymbol{\beta} + \begin{pmatrix} \boldsymbol{Q}_1^T\boldsymbol{B}\boldsymbol{Z}_1 & \boldsymbol{Q}_1^T\boldsymbol{B}\boldsymbol{Z}_2 \\ \boldsymbol{0} & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \boldsymbol{Z}_1^T\boldsymbol{v} \\ \boldsymbol{Z}_2^T\boldsymbol{v} \end{pmatrix} = \begin{pmatrix} \boldsymbol{Q}_1^T\boldsymbol{y} \\ \boldsymbol{Q}_2^T\boldsymbol{y} \end{pmatrix}.$$

  From the bottom half we can solve for $\boldsymbol{v}$ from the equation (how?)

$$\boldsymbol{S}\boldsymbol{Z}_2^T\boldsymbol{v} = \boldsymbol{Q}_2^T\boldsymbol{y}.$$

  Then we solve for $\boldsymbol{\beta}$ from the equation

$$\boldsymbol{R}_1\boldsymbol{\beta} = \boldsymbol{Q}_1^T\boldsymbol{y} - (\boldsymbol{Q}_1^T\boldsymbol{B}\boldsymbol{Z}_1\boldsymbol{Z}_1^T + \boldsymbol{Q}_1^T\boldsymbol{B}\boldsymbol{Z}_2\boldsymbol{Z}_2^T)\boldsymbol{v} = \boldsymbol{Q}_1^T\boldsymbol{y} - \boldsymbol{Q}_1^T\boldsymbol{B}\boldsymbol{Z}_2(\boldsymbol{Z}_2^T\boldsymbol{v}).$$

- Paige's method also works for singular $\boldsymbol{X}$ and $\boldsymbol{B}$ (using QR with column pivoting).

- MATLAB's `lscov()` function implements Paige's method for singular covariance $\boldsymbol{V}$. No R implementation (?)

Ridge regression

- In ridge regression, we minimize

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_2^2,$$

  where λ is a tuning parameter.

- Ridge regression by augmented linear regression. Ridge regression problem is equivalent to

$$\left\| \begin{pmatrix} \boldsymbol{y} \\ \boldsymbol{0}_p \end{pmatrix} - \begin{pmatrix} \boldsymbol{X} \\ \sqrt{\lambda}\boldsymbol{I}_p \end{pmatrix} \boldsymbol{\beta} \right\|_2^2.$$

  Therefore any methods for linear regression can be applied.

- Ridge regression by method of normal equation. The normal equation for the ridge problem is

$$(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}_p)\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{y}.$$

  Therefore Cholesky or sweep can be used.

- Ridge regression by SVD. If we obtain the (thin) SVD of $\boldsymbol{X}$

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}_{p\times p}\boldsymbol{V}^T.$$

  Then the normal equation reads

$$(\boldsymbol{\Sigma}^2 + \lambda\boldsymbol{I}_p)\boldsymbol{V}^T\boldsymbol{\beta} = \boldsymbol{\Sigma}\boldsymbol{U}^T\boldsymbol{y}$$

  and we get

$$\widehat{\boldsymbol{\beta}}(\lambda) = \sum_{i=1}^{p} \frac{\sigma_i \boldsymbol{u}_i^T\boldsymbol{y}}{\sigma_i^2 + \lambda}\boldsymbol{v}_i = \sum_{i=1}^{r} \frac{\sigma_i \boldsymbol{u}_i^T\boldsymbol{y}}{\sigma_i^2 + \lambda}\boldsymbol{v}_i, \quad r = \operatorname{rank}(\boldsymbol{X}).$$

  It is clear that

$$\lim_{\lambda\to 0} \widehat{\boldsymbol{\beta}}(\lambda) = \widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}$$

  and $\|\widehat{\boldsymbol{\beta}}(\lambda)\|_2$ is monotone decreasing as λ increases.

- Only one SVD is needed for all λ (!), in contrast to the method of augmented linear regression, Cholesky, or sweep.

Least squares over a sphere

- Ridge regression "shrinks" the solution via penalty. Alternatively we can simply fit a least squares problem subject to the constraint that the solution lives in a sphere

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{\beta}\|_2 \leq \alpha. \end{aligned}$$

- Suppose we obtain the (thin) SVD $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}_{p\times p}\boldsymbol{V}^T$. If the ordinary least squares solution

$$\widehat{\boldsymbol{\beta}}_{\text{OLS}} = \sum_{i=1}^{r} \frac{\boldsymbol{u}_i^T\boldsymbol{y}}{\sigma_i}\boldsymbol{v}_i$$

has ℓ_2 norm less than α, then we are done. If not, we use the method of Lagrangian multipliers

$$\psi(\boldsymbol{\beta}, \lambda) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \frac{\lambda}{2}(\|\boldsymbol{\beta}\|_2^2 - \alpha^2).$$

Setting the gradient to 0, we have the shifted normal equation

$$(\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}_p)\boldsymbol{\beta} = \boldsymbol{X}^T\boldsymbol{y},$$

which has solution

$$\widehat{\boldsymbol{\beta}}(\lambda) = \sum_{i=1}^{r} \frac{\sigma_i\boldsymbol{u}_i^T\boldsymbol{y}}{\sigma_i^2 + \lambda}\boldsymbol{v}_i.$$

We need to choose the λ such that $\|\widehat{\boldsymbol{\beta}}(\lambda)\|_2 = \alpha$. That is we need to find the (unique) zero of the function

$$f(\lambda) = \|\widehat{\boldsymbol{\beta}}(\lambda)\|_2^2 - \alpha^2 = \sum_{i=1}^{r}\left(\frac{\sigma_i\boldsymbol{u}_i^T\boldsymbol{y}}{\sigma_i^2 + \lambda}\right)^2 - \alpha^2.$$

This is easily achieved by Newton's or other methods.

Least squares with equality constraints

- In many applications, there are *a priori* constraints on the regression parameters. Let's consider how to solve linear regression with equality constraints (LSE)

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \\ \text{subject to} \quad & \boldsymbol{B}\boldsymbol{\beta} = \boldsymbol{d}. \end{aligned}$$

- LSE by QR. First compute QR of $\boldsymbol{B}^T \in \mathbb{R}^{p\times m}$

$$\boldsymbol{B}^T = \boldsymbol{Q}\begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{0} \end{pmatrix}$$

and set

$$\boldsymbol{X}\boldsymbol{Q} = (\boldsymbol{X}_1, \boldsymbol{X}_2) \quad \text{and} \quad \boldsymbol{Q}^T\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix}.$$

Then the original minimization problem becomes

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{y} - \boldsymbol{X}_1\boldsymbol{\beta}_1 - \boldsymbol{X}_2\boldsymbol{\beta}_2\|_2^2 \\ \text{subject to} \quad & \boldsymbol{R}^T\boldsymbol{\beta}_1 = \boldsymbol{d}. \end{aligned}$$

Now $\boldsymbol{\beta}_1$ is determined from the constraint $\boldsymbol{R}^T\boldsymbol{\beta}_1 = \boldsymbol{d}$ and $\boldsymbol{\beta}_2$ is solved from the unconstrained least squares problem

$$\text{minimize} \quad \|(\boldsymbol{y} - \boldsymbol{X}_1\boldsymbol{\beta}_1) - \boldsymbol{X}_2\boldsymbol{\beta}_2\|_2^2.$$

Finally we recover the solution from

$$\boldsymbol{\beta} = \boldsymbol{Q}\begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \end{pmatrix}.$$

- LSE by augmented system. Define the Lagrangian function

$$\phi(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 - \boldsymbol{\lambda}^T(\boldsymbol{B}\boldsymbol{\beta} - \boldsymbol{d}).$$

Setting gradient to zero yields

$$\begin{aligned} \boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{B}^T\boldsymbol{\lambda} &= \boldsymbol{X}^T\boldsymbol{y} \\ \boldsymbol{B}\boldsymbol{\beta} &= \boldsymbol{d}, \end{aligned}$$

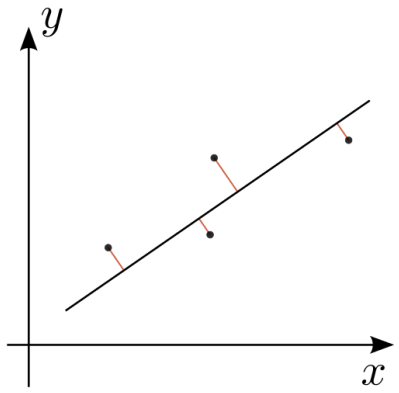
suggesting the augmented system

$$\begin{pmatrix} \boldsymbol{X}^T\boldsymbol{X} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta} \\ -\boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}^T\boldsymbol{y} \\ \boldsymbol{d} \end{pmatrix}.$$

This linear system is non-singular when $\boldsymbol{X}$ and $\boldsymbol{B}$ have full rank and can be solved by Cholesky, sweep, and so on.

- LSE by generalized SVD.

Total least squares (TLS)



TLS considers the case both predictors and observations are subject to errors. It is solved by SVD. Read KL 9.3.6 if interested.

Tikhonov regularization

Tikhonov regularization is an extension of the ridge regression

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{B}\boldsymbol{\beta}\|_2^2,$$

where $\boldsymbol{B} \in \mathbb{R}^{m\times p}$ is a fixed regularization matrix and λ is a tuning parameter. It is solved by the generalized singular value decomposition (GSVD).

Least squares with quadratic inequality constraint (LSQI)

Least squares with quadratic inequality constraint (LSQI) minimizes the least squares criterion over a hyper-ellipsoid:

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \\ \text{subject to} \quad & \|\boldsymbol{B}\boldsymbol{\beta}\|_2 \leq \alpha, \end{aligned}$$

where $\boldsymbol{B} \in \mathbb{R}^{m \times p}$ is a fixed regularization matrix. It is solved by the generalized singular value decomposition (GSVD). See Golub and Van Loan (1996, Section 2.1.1).

Concluding remarks on numerical linear algebra

- Numerical linear algebra forms the building blocks of most computation we do.
- Be flop and memory aware.

  > The form of a mathematical expression and the way the expression should be evaluated in actual practice may be quite different.

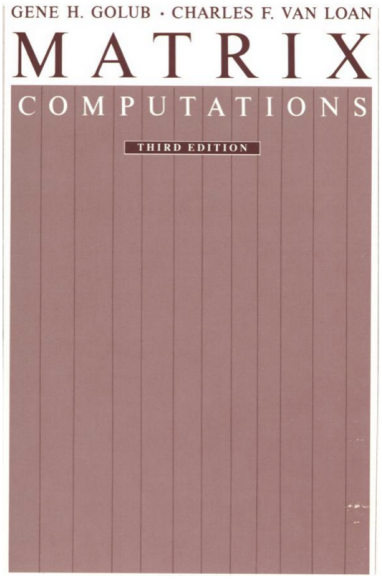
- Be alert to problem structure and make educated choice of software/algorithm.

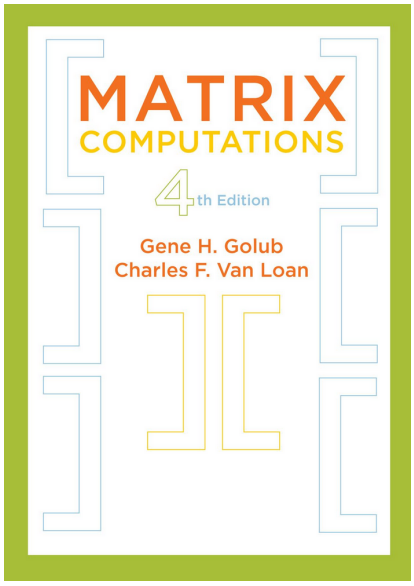
  > The structure should be exploited whenever solving a problem.

- Do not write your own matrix computation routines unless for good reason. Utilize `BLAS` and `LAPACK` as much as possible!
- In contrast, for optimization, often we need to devise problem specific optimization routines, or even "mix and match" them.

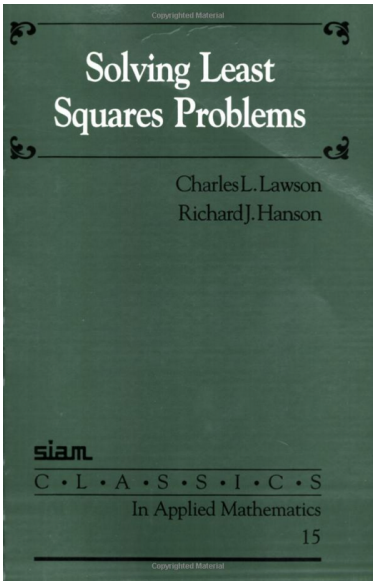
Reference books on numerical linear algebra

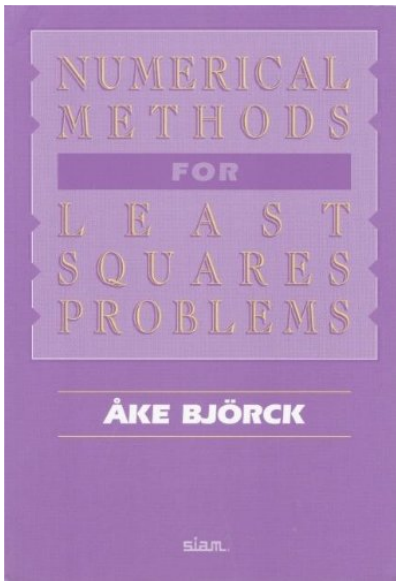
- Golub and Van Loan (1996): "Bible" in numerical linear algebra. Good for reference.



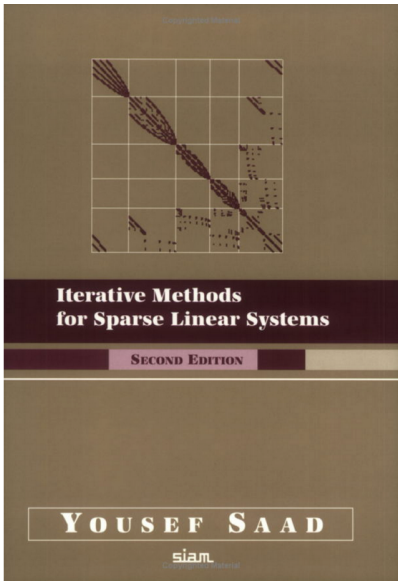


- Lawson and Hanson (1987) and Björck (1996): classical monographs on solving least squares problems.





- Saad (2003): standard reference for iterative methods



MLE (as a motivation for optimization)

A great idea due to Fisher in 20s, and made rigorous by Cramer and others in 40s.

- Notations:
  - Density: $f(\boldsymbol{x}|\boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subset \mathbb{R}^p$
  - Log-likelihood function: $L(\boldsymbol{\theta}) = \ln f(\boldsymbol{x}|\theta)$
  - (Column) Gradient/score vector: $\nabla L(\boldsymbol{\theta}) \in \mathbb{R}^{p \times 1}$
  - Differential: $dL(\boldsymbol{\theta}) = [\nabla L(\boldsymbol{\theta})]^\intercal \in \mathbb{R}^{1 \times p}$
  - Hessian: $d^2 L(\boldsymbol{\theta}) = \nabla^2 L(\boldsymbol{\theta})$
  - Observed information matrix: $-d^2 L(\boldsymbol{\theta})$
  - Expected (Fisher) information matrix: $\boldsymbol{I}(\boldsymbol{\theta}) = \mathbf{E}_{\boldsymbol{\theta}}[-d^2 L(\boldsymbol{\theta})]$
  - Given iid observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ from $f(\cdot|\boldsymbol{\theta})$,

$$L_n(\boldsymbol{\theta}) \;\; = \;\; \sum_{i=1}^{n} \ln f(\boldsymbol{x}_i|\boldsymbol{\theta})$$

  - Maximum likelihood estimator (MLE):

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \text{argmax}_{\boldsymbol{\theta}} \, L_n(\boldsymbol{\theta})$$

- Consistency of MLE
  - Under the true parameter value $\boldsymbol{\theta}_0$,

$$\begin{aligned} & M_n(\boldsymbol{\theta}) = \frac{1}{n}[L_n(\boldsymbol{\theta}) - L_n(\boldsymbol{\theta}_0)] \\ \rightarrow \quad & M(\boldsymbol{\theta}) = \mathbf{E}_{\boldsymbol{\theta}_0}[\ln f(\boldsymbol{X}|\boldsymbol{\theta}) - \ln f(\boldsymbol{X}|\boldsymbol{\theta}_0)] \end{aligned}$$

    for all $\boldsymbol{\theta}$ almost surely.
  - Note that $M(\boldsymbol{\theta})$ is the negative Kullback-Leibler divergence between distribution at $\boldsymbol{\theta}$ and distribution at $\boldsymbol{\theta}_0$.

    Assuming *identifiability*, by the *information inequality*, $M(\boldsymbol{\theta})$ achieves maximum uniquely at $\boldsymbol{\theta}_0$. We hope the MLE

$$\hat{\boldsymbol{\theta}}_n = \text{argmax}_{\boldsymbol{\theta}} \, \boldsymbol{M}_n(\boldsymbol{\theta})$$

    converges to

$$\boldsymbol{\theta}_0 = \text{argmax}_{\boldsymbol{\theta}} \, \boldsymbol{M}(\boldsymbol{\theta}).$$

  - Need *uniform convergence* of $M_n(\boldsymbol{\theta})$ to $M(\boldsymbol{\theta})$, i.e.,

$$\sup_{\Theta} |M_n(\boldsymbol{\theta}) - M(\boldsymbol{\theta})|$$

    converges to 0 in probability. A set of sufficient conditions for uniform convergence:
    - compactness of the parameter space $\boldsymbol{\Theta}$
    - continuity of $M(\boldsymbol{\theta})$ in $\boldsymbol{\theta}$ for any $\boldsymbol{x}$
    - $M(\boldsymbol{\theta})$ dominated by an integrable function
  - Example of non-uniform convergence: $f_n(x) = 1_{\{n,n+1\}}$ (or a triangle on $[n, n+1]$ if we want f_n to be continuous). $f_n \to f \equiv 0$ pointwise but not uniformly.
- Asymptotic normality of MLE
  - Assume $\hat{\boldsymbol{\theta}}_n$ is consistent for $\boldsymbol{\theta}_0$.

– Taylor expansion on $\mathbf{0}_p = \frac{1}{n}\nabla L_n(\hat{\boldsymbol{\theta}}_n)$ gives

$$\begin{aligned}\mathbf{0}_p &= \frac{1}{n}\nabla L_n(\boldsymbol{\theta}_0) + \left[\frac{1}{n}d^2 L_n(\boldsymbol{\theta}_0)\right](\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \\ &\quad + \frac{1}{2}[\boldsymbol{I}_p \otimes (\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0)^{\intercal}]\left[\frac{1}{n}Dd^2 L_n(\tilde{\boldsymbol{\theta}}_n)\right](\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0),\end{aligned}$$

where $\tilde{\boldsymbol{\theta}}_n$ is somewhere between $\boldsymbol{\theta}_0$ and $\hat{\boldsymbol{\theta}}_n$. If $\frac{1}{n}Dd^2 L_n(\tilde{\boldsymbol{\theta}}_n) = O_p(1)$ (bounded in probability), then the third term is $o_p(1)(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0)$ and

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) = \left[-\frac{1}{n}d^2 L_n(\boldsymbol{\theta}_0) + o_p(1)\right]^{-1}\frac{\sqrt{n}}{n}\nabla L_n(\boldsymbol{\theta}_0).$$

Now

* $-\frac{1}{n}d^2 L_n(\boldsymbol{\theta}_0) + o_p(1) \to \mathbf{E}_{\boldsymbol{\theta}_0}[-d^2 L(\boldsymbol{\theta}_0)] = \boldsymbol{I}(\boldsymbol{\theta}_0)$ almost surely by the law of large number.
* $n^{-1/2}\nabla L_n(\boldsymbol{\theta}_0)$ converges to a multivariate normal with mean $\mathbf{0}_p$ and variance

$$\mathbf{E}_{\boldsymbol{\theta}_0}[\nabla L(\boldsymbol{\theta}_0) dL(\boldsymbol{\theta}_0)],$$

which equals $\boldsymbol{I}(\boldsymbol{\theta}_0)$ under exchangeability of integral and differentiation.

Then by the Slutsky theorem,

$$\begin{aligned}&\sqrt{n}(\hat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_0) \\ \to\ & N_p\left(\mathbf{0}_p, \boldsymbol{I}^{-1}(\boldsymbol{\theta}_0)\cdot\mathbf{E}_{\boldsymbol{\theta}_0}[\nabla \ln f(\boldsymbol{\theta}_0) d\ln f(\boldsymbol{\theta}_0)]\cdot\boldsymbol{I}^{-1}(\boldsymbol{\theta}_0)\right) \\ =\ & N_p\left(\mathbf{0}_p, \boldsymbol{I}^{-1}(\boldsymbol{\theta}_0)\right)\end{aligned}$$

in distribution.

– In practice, we can estimate the variance by

* Fisher information matrix $\boldsymbol{I}^{-1}(\hat{\boldsymbol{\theta}})$,
* observed information matrix $[-(1/n)d^2 L_n(\hat{\boldsymbol{\theta}})]^{-1}$, or
* the sandwich estimator

- Asymptotic efficiency of MLE.

"Cramer-Rao theorem" says the variance of any unbiased estimator is "at least" $(nI(\boldsymbol{\theta}_0))^{-1}$ (the difference is psd). So MLE has the smallest asymptotic variance within the class of unbiased estimators.

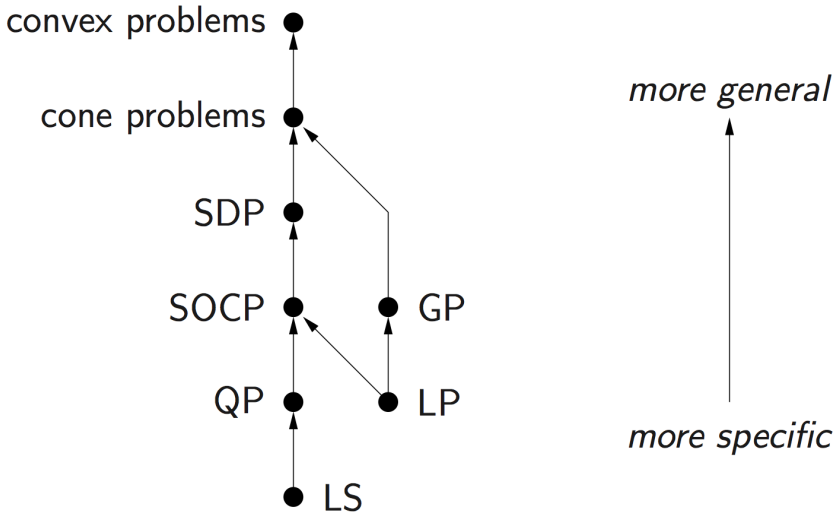
Hierarchy of optimization problems

Difficulty of optimization problems *in general*

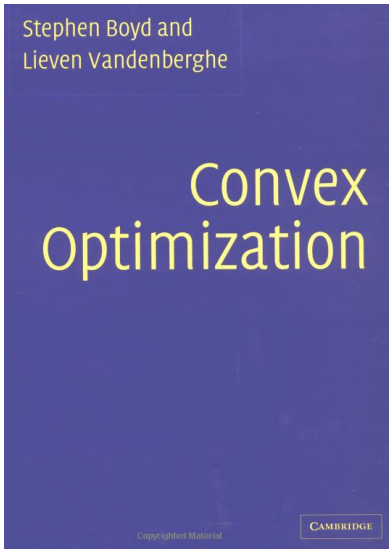
Harder	Easier
discrete (combinatorial) optimization	continuous optimization
non-smooth	smooth
non-convex	convex
constrained	un-constrained
inequality constraint	equality constrained

Convex optimization

- *Extremely important* skill to recognize or transform to convex problems



  - Examples: ℓ_∞ regression, ℓ_1 regression, quantile regression, and many more.
  - *Convex Optimization* by Boyd and Vandenberghe and accompanying slides `http://www.stanford.edu/~boyd/cvxbook/`



  - Lecture videos:
    `http://www.stanford.edu/class/ee364a/videos.html`
    `http://www.stanford.edu/class/ee364b/videos.html`

- UCLA courses by Lieven Vandenberghe: EE236A (Linear Programming), EE236B (Convex Optimization), EE236C (Optimization Methods for Large-scale Systems).

- Convex programming (LS, LP, QP, GP, SOCP, SDP) is almost becoming a technology (`Cplex`, `Gurobi`, `Mosek`, `cvx`, `Matlab`, `JuliaOpt`, ...), just like numerical linear algebra libraries BLAS and LAPACK.

- Non-convex optimization still occurs in many natural statistical applications. Statisticians have specialized tools to deal with them (Fisher scoring method, EM algorithm, simulated annealing, ...)

Unconstrained optimization (KL 11.2)

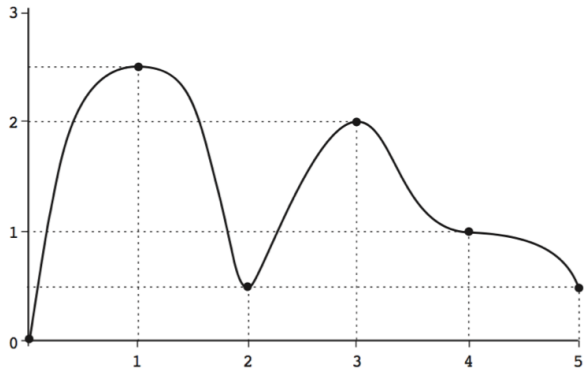


Figure 1 Unconstrained optimization in one variable

- Possible confusion:
  - We (statisticians) talk about *maximization*: max $L_n(\boldsymbol{\theta})$.
  - People talk about *minimization* in the optimization world: $\min_{\boldsymbol{x}} f(\boldsymbol{x})$.
- Fundamental questions: When does a function have minimum? How do we tell whether a point is minimum?
- When does a function have a minimum?

  (Weierstrass) A continuous function $f(x)$ defined on a compact (closed and bounded) set is bounded below and attains its minimum.
- None of the Weierstrass conditions can be taken out.
  - $f(x) = x$, $x \in (-\infty, \infty)$. Non-compact support.
  - $f(x) = \tan(x)$, $x \in (-\pi/2, \pi/2)$. Non-compact support.
  - $f(x) = x$, $x \in (-1, 1)$ and $f(-1) = f(1) = 0$. The minimum not attained by the *discontinuous* function f.
- None of the Weierstrass conditions are necessary. $f(x) = x$, $x \in [0, 2)$, $f(x) = 1$, $x \in (2, \infty)$.

- Coercive function: $\{\boldsymbol{x} \in U : f(\boldsymbol{x}) \leq f(\boldsymbol{y})\}$ is compact for all $\boldsymbol{y} \in U$.

  Weierstrass theorem also holds for a coercive function defined on a possibly open U.

- Necessary conditions for a local minimum.

  Assume f has a local minimum at interior point $\boldsymbol{y} \in U$.

  - (Fermat) If f is differentiable, then $\nabla f(\boldsymbol{x})$ vanishes at $\boldsymbol{y}$.
  - If f is twice differentiable, then $d^2 f(\boldsymbol{y})$ is psd.

- Points with $\nabla f(\boldsymbol{x}) = \mathbf{0}$ are called *stationary points* or *critical points*. Most optimization algorithms try to find the stationary points of the function and then check sufficient condition.

- Counter-examples to necessary conditions. (1) $f(x) = x^3$ has zero gradient at 0, which is not local minimum. (2) $f(x) = |x|$ has local minimum at 0, where the gradient does not exist.

- (A first-order sufficient condition; first derivative test) Suppose f is differentiable in a ball $B(\boldsymbol{y})$ around an interior point $\boldsymbol{y}$, and $\langle \nabla f(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{y} \rangle \geq 0$ for all $\boldsymbol{x} \in B(\boldsymbol{y})$, then $\boldsymbol{y}$ is a local minimum.

- (A second-order sufficient condition; second derivative test) If $\nabla f(\boldsymbol{y}) = \mathbf{0}$ and $d^2(\boldsymbol{y})$ is pd, then y is a strict local minimum.

- Remark: In case $d^2 f(\boldsymbol{y})$ is neither positive definite nor negative definite but non-singular, $\boldsymbol{y}$ is a *saddle point*, i.e., a stationary point that is neither a local minimum nor a local maximum. In case $d^2 f(\boldsymbol{y})$ is singular, we cannot tell.

- Example: $f_1(x, y) = x^4 + y^4$, $f_2(x, y) = -x^4 - y^4$, $f_3(x, y) = x^3 + y^3$. Origin is a stationary (critical) point and the Hessian $d^2 f_i(0, 0) = \mathbf{0}_{2\times 2}$ is singular. Origin is a minimum, maximum, and a saddle point respectively.

Convexity and global optima

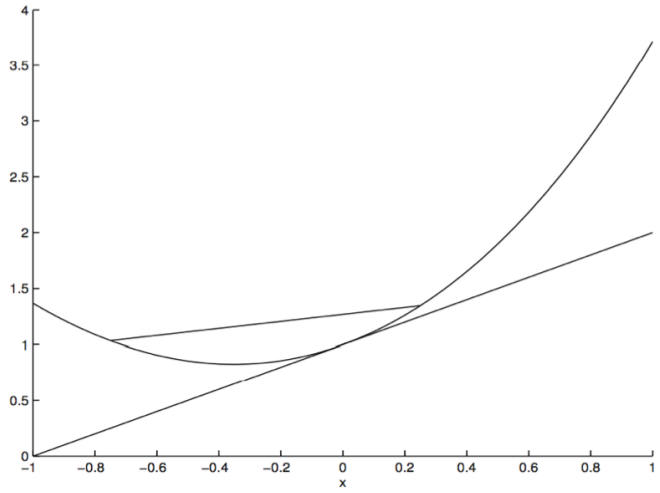


FIGURE 11.2. Plot of the Convex Function $e^x + x^2$

- $f : U \mapsto \mathbb{R}$ is *convex* if
  - U is a convex set ($\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in U$ for all $\boldsymbol{x}, \boldsymbol{y} \in U$ and $\lambda \in (0,1)$), and
  - $f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y})$, for all $\boldsymbol{x}, \boldsymbol{y} \in U$ and $\lambda \in (0,1)$.

  f is *strictly convex* if the inequality if strict for all $\boldsymbol{x} \neq \boldsymbol{y} \in U$ and λ.

- (*Supporting hyperplane inequality*) A differentiable function f is convex if and only if $f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle$ for all $\boldsymbol{x}, \boldsymbol{y} \in U$.

- (Second-order condition for convexity) A twice differentiable function f is convex if and only if $d^2 f(\boldsymbol{x})$ is psd for all $\boldsymbol{x} \in U$.

  It is strictly convex if $d^2 f(\boldsymbol{x})$ is pd for all $\boldsymbol{x} \in U$.

- (Convexity and global optima) Suppose f is a convex function on a convex set U.

  1. Any stationary point $\boldsymbol{y}$ is a global minimum. (By supporting hyperplane inequality, $f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle = f(\boldsymbol{y})$ for all $\boldsymbol{x} \in U$.)
  2. Any local minimum is a global minimum.

3. The set of (global) minima $\{\boldsymbol{x} \in U : f(\boldsymbol{x}) = f(\boldsymbol{y})\}$ is convex.
4. If f is strictly convex, then the global minimum, if exists, is unique.

- Example: Least squares estimate. $f(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$ has Hessian $d^2 f = \boldsymbol{X}^\intercal \boldsymbol{X}$ which is psd. So f is convex and any stationary point (solution to the normal equation) is a global minimum. When $\boldsymbol{X}$ is rank deficient, the set of solutions is convex.

- (Jensen's inequality) W a random variable taking values in U and h is convex on U. Then
$$\mathbf{E}[h(W)] \geq h[\mathbf{E}(W)],$$
provided both expectations exist. For a strictly convex h, equality holds if and only if $W = \mathbf{E}(W)$ almost surely.

  Proof: supporting hyperplane inequality taking $\boldsymbol{x} = \boldsymbol{W}$ and $\boldsymbol{y} = E(\boldsymbol{W})$.

- (Information inequality) Let f and g be two densities with respect to a common measure μ. $h, g > 0$ almost everywhere relative to μ. Then
$$\mathbf{E}_f(\ln f) \geq \mathbf{E}_f(\ln g),$$
with equality if and only if $f = g$ almost everywhere on μ.

  Proof: Apply Jensen's inequality to the convex function $-\ln(t)$ and random variable $W = g(x)/f(x)$.
  Applications: M-estimation, EM algorithm.

Optimization with equality constraints (KL 11.3)

Consider the equality constrained minimization problem

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) = 0, i = 1, \ldots, m \\ & \boldsymbol{x} \in U \subset \mathbb{R}^n. \end{aligned}$$

We write

$$g(\boldsymbol{x}) = \begin{pmatrix} g_1(\boldsymbol{x}) \\ \vdots \\ g_m(\boldsymbol{x}) \end{pmatrix} \in \mathbb{R}^m \text{ and } Dg(\boldsymbol{x}) = \begin{pmatrix} dg_1(\boldsymbol{x}) \\ \cdots \\ dg_m(\boldsymbol{x}) \end{pmatrix} \in \mathbb{R}^{m \times n}.$$

- Method of Lagrange multiplier. *Lagrangian* function

$$L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^{\mathsf{T}} g(\boldsymbol{x}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}).$$

  Strategy for finding the equality constrained minimum: find the stationary point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ of the Lagrangian,

$$\begin{aligned} \nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}) &= \nabla f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\boldsymbol{x}) = \mathbf{0}_n \\ g(\boldsymbol{x}) &= \mathbf{0}_m. \end{aligned}$$

- Intuition: Null space of the matrix Dg is the tangent space. Movement along the tangent space does not change constraint function values. We need the ∇f to be orthogonal to the tangent space. In other words, ∇f is in the column space of $[Dg]^T$.

- Intuition of the Lagrange multiplier method: Hill climb along a trail which is a contour line of the constraint function. We feel effortless exactly when the direction of our movement is perpendicular to the steepest ascent direction of the hill. In other words, steepest ascent direction of the constraint function aligns with that of the hill.

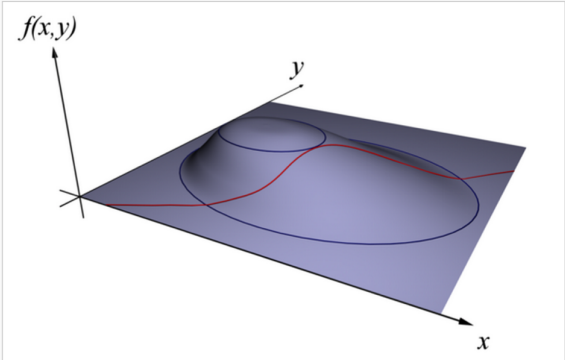


Figure 1: Find *x* and *y* to maximize $f(x, y)$ subject to a constraint (shown in red) $g(x, y) = c$.

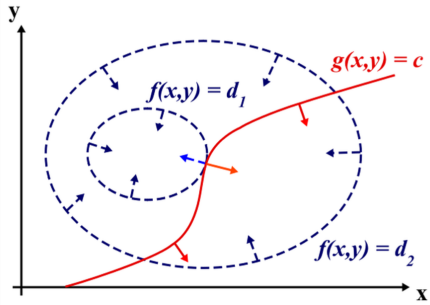


Figure 2: Contour map of Figure 1. The red line shows the constraint $g(x, y) = c$. The blue lines are contours of $f(x, y)$. The point where the red line tangentially touches a blue contour is our solution.Since $d_1 > d_2$, the solution is a maximization of $f(x, y)$

- (Necessary condition for a constrained local minimum) Assume conditions (i) $g(\boldsymbol{y}) = \mathbf{0}_m$, (2) f and g are differentiable in some n-ball $B(\boldsymbol{y})$, (iii) $Dg(\boldsymbol{y}) \in \mathbb{R}^{m\times n}$ is continuous at $\boldsymbol{y}$, (iv) $Dg(\boldsymbol{y})$ has full row rank, (v) $f(\boldsymbol{x}) \geq f(\boldsymbol{y})$ for any $\boldsymbol{x} \in B(\boldsymbol{y})$ satisfying $g(\boldsymbol{x}) = \mathbf{0}_m$ ($\boldsymbol{y}$ a local minimum subject to constraints). Then there exists $\boldsymbol{\lambda} \in \mathbb{R}^m$ satisfying $\nabla f(\boldsymbol{y}) + \sum_{i=1}^m \lambda_i \nabla g_i(\boldsymbol{y}) = \mathbf{0}_n$, i.e., $(\boldsymbol{y}, \boldsymbol{\lambda})$ is a stationarity point of the Lagrangian $L(\boldsymbol{x}, \boldsymbol{\lambda})$. In other words, there exists $\boldsymbol{\lambda} \in \mathbb{R}^m$, such that $\nabla L(\boldsymbol{y}, \boldsymbol{\lambda}) = \mathbf{0}_{m+n}$.

- (Sufficient condition for a constrained local minimum) (i) f twice differentiable at $\boldsymbol{y}$, (ii) g twice differentiable at $\boldsymbol{y}$, (iii) the Jacobian matrix $Dg(\boldsymbol{y}) \in \mathbb{R}^{m\times n}$ has full row rank m, (iv) it is a stationarity point of the Lagrangian at a given $\boldsymbol{\lambda} \in \mathbb{R}^m$, (v) $\boldsymbol{u}^\top d^2 f(\boldsymbol{y}) \boldsymbol{u} > 0$ for all $\boldsymbol{u} \neq \mathbf{0}_n$ satisfying $[Dg(\boldsymbol{y})]\boldsymbol{u} = \mathbf{0}_m$ (tangent vectors). Then $\boldsymbol{y}$ is a strict local minimum of f under constraint $g(\boldsymbol{y}) = \mathbf{0}_m$.

- Check condition (v). Condition (v) is equivalent to the "bordered determinantal criterion"

$$(-1)^m \det \begin{pmatrix} \mathbf{0}_{m\times m} & \boldsymbol{B}_r \\ \boldsymbol{B}_r^\top & \boldsymbol{A}_{rr} \end{pmatrix} > 0$$

  for $r = m+1, \ldots, n$, where

  - $\boldsymbol{A}_{rr}$ is the top left r-by-r block of $d^2 f(\boldsymbol{y}) + \sum_{i=1}^m \lambda_i d^2 g_i(\boldsymbol{y})$
  - $\boldsymbol{B}_r \in \mathbb{R}^{m\times r}$ is the first r columns of the $Dg(\boldsymbol{y})$.

- (Sufficient condition for a global constrained minimum) Lagrangian first order condition + convexity of the Lagrangian on U.

- (Interpretation of the Lagrange multipliers $\boldsymbol{\lambda}$). Consider $\min f(\boldsymbol{x})$ subject to some resource constraint $g(\boldsymbol{x}) = \boldsymbol{b}$. Consider the solution $\boldsymbol{x}^*(\boldsymbol{b})$ as a function of $\boldsymbol{b}$. Then it can be shown that

$$\frac{\partial f(\boldsymbol{x}^*(\boldsymbol{b}))}{\partial b_j} = \lambda_j.$$

  That's why the score test is classically called the Lagrange multiplier test.

- Example: Linearly constrained least squares solution. $\min \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2$ subject to linear constrained $\boldsymbol{V}\boldsymbol{\beta} = \boldsymbol{d}$. Form the Lagrangian

$$L(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \boldsymbol{\lambda}^\top(\boldsymbol{V}\boldsymbol{\beta} - \boldsymbol{d}).$$

Stationary condition says

$$\begin{aligned} \boldsymbol{X}^\top \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{X}^\top \boldsymbol{y} + \boldsymbol{V}^\top \boldsymbol{\lambda} &= \mathbf{0}_p \\ \boldsymbol{V}\boldsymbol{\beta} &= \boldsymbol{d} \end{aligned}$$

or equivalently

$$\begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{X} & \boldsymbol{V}^\top \\ \boldsymbol{V} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{d} \end{pmatrix},$$

which can be solved by say sweeping on

$$\begin{pmatrix} \boldsymbol{X}^\top \boldsymbol{X} & \boldsymbol{V}^\top & \boldsymbol{X}^\top \boldsymbol{y} \\ \boldsymbol{V} & \mathbf{0} & \boldsymbol{d} \\ \boldsymbol{y}\boldsymbol{X}^\top & \boldsymbol{d}^\top & \boldsymbol{y}^\top \boldsymbol{y} \end{pmatrix},$$

or Cholesky or QR.

Optimization with both equality and inequality constraints (KL 11.4)

Consider the constrained minimization problem

$$\begin{aligned} \text{minimize} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & g_i(\boldsymbol{x}) = 0, i = 1, \ldots, p \\ & h_j(\boldsymbol{x}) \leq 0, i = 1, \ldots, q \\ & \boldsymbol{x} \in U \subset \mathbb{R}^n. \end{aligned}$$

- Lagrangian function:

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{p} \lambda_i g_i(\boldsymbol{x}) + \sum_{j=1}^{q} \mu_j h_j(\boldsymbol{x}).$$

- Karush-Kuhn-Tucker (KKT) necessary condition: If (1) $\boldsymbol{y}$ is a local constrained minimum and (2) satisfies certain constraint qualifications (Kuhn-Tucker, Mangasarian-Fromovitz), then

  1. (Lagrangian stationarity condition) there exist $\boldsymbol{\lambda} \in \mathbb{R}^p$, $\boldsymbol{\mu} \in \mathbb{R}^q$ such that

$$\nabla f(\boldsymbol{y}) + \sum_{i=1}^{p} \lambda_i \nabla g_i(\boldsymbol{y}) + \sum_{j=1}^{q} \mu_j \nabla h_j(\boldsymbol{y}) = \mathbf{0},$$

2. (Complementary slackness) $\mu_j = 0$ if $h_j(\boldsymbol{y}) < 0$ and $\mu_j > 0$ otherwise.

- Sufficient condition: KKT + second order condition.
- Global minimum: KKT conditions + convexity.
- Read KL Section 11.4 for more details. KKT is "one of the great triumphs of 20th century applied mathematics".

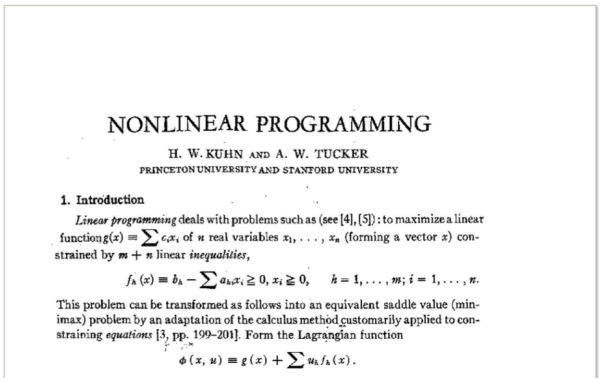
1. ^ Kuhn, H. W.; Tucker, A. W. (1951). "Nonlinear programming". *Proceedings of 2nd Berkeley Symposium*. Berkeley: University of California Press. pp. 481–492. MR47303
2. ^ W. Karush (1939). *Minima of Functions of Several Variables with Inequalities as Side Constraints*. M.Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago, Chicago, Illinois.

Nonlinear Programming

H. W. Kuhn, and A. W. Tucker

Source: Proc. Second Berkeley Symp. on Math. Statist. and Prob. (Univ. of Calif. Press, 1951), 481-492.

First Page: Hide



NONLINEAR PROGRAMMING

H. W. KUHN AND A. W. TUCKER
PRINCETON UNIVERSITY AND STANFORD UNIVERSITY

1. Introduction

Linear programming deals with problems such as (see [4], [5]): to maximize a linear function $g(x) = \sum c_i x_i$ of n real variables $x_1, \ldots, x_n$ (forming a vector x) constrained by $m + n$ linear *inequalities*,

$$f_h(x) = b_h - \sum a_{hi} x_i \geqq 0, \quad x_i \geqq 0, \qquad h = 1, \ldots, m; \; i = 1, \ldots, n.$$

This problem can be transformed as follows into an equivalent saddle value (min-imax) problem by an adaptation of the calculus method customarily applied to constraining *equations* [3, pp. 199–201]. Form the Lagrangian function

$$\phi(x, u) = g(x) + \sum u_h f_h(x).$$