EECS 275 Matrix Computation

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

Lecture 9

Overview

- Least squares minimization
- Regression
- Regularization

Reading

- Chapter 11 of *Numerical Linear Algebra* by Llyod Trefethen and David Bau
- Chapter 5 of *Matrix Computations* by Gene Golub and Charles Van Loan
- Chapter 4 of *Matrix Analysis and Applied Linear Algebra* by Carl Meyer
- Chapter 11 and Chapter 15 of *Matrix Algebra From a Statistician's Perspective* by David Harville

Matrix differentiation

- First order differentiation of linear form:

$$\mathbf{a}^\top \mathbf{x} = \mathbf{x}^\top \mathbf{a} = \sum_i a_i x_i$$

$$\frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^\top \mathbf{x}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial x_1} \\ \vdots \\ \frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial x_n} \end{bmatrix} = \mathbf{a}$$

$$\frac{\partial x_i}{\partial x_j} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

$$\frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial x_j} = \frac{\partial (\sum_i a_i x_i)}{\partial x_j} = \sum_i a_i \frac{\partial x_i}{\partial x_j} = a_j$$

- Likewise

$$\frac{\partial \mathbf{x}^\top \mathbf{a}}{\partial \mathbf{x}^\top} = \mathbf{a}^\top$$

$$\frac{\partial (A\mathbf{x})}{\partial \mathbf{x}^\top} = A \qquad \frac{\partial (A\mathbf{x})^\top}{\partial \mathbf{x}} = A^\top$$

Matrix differentiation (cont'd)

- First order differentiation of quadratic form:

$$\mathbf{x}^\top A\mathbf{x} = \sum_{i,k} a_{ik}x_ix_k$$

$$\frac{\partial \mathbf{x}^\top A\mathbf{x}}{\partial \mathbf{x}} = (A + A^\top)\mathbf{x}$$

$$\frac{\partial(x_ix_k)}{\partial x_j} = \begin{cases} 2x_j & \text{if } i = k = j \\ x_i & \text{if } k = j, i \neq j \\ x_k & \text{if } i = j, k \neq j \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned}
\frac{\partial \mathbf{x}^\top A\mathbf{x}}{\partial x_j} &= \frac{\partial \sum_{i,k} a_{ik}x_ix_k}{\partial x_j} \\
&= \frac{\partial(a_{jj}x_j^2 + \sum_{i\neq j} a_{ij}x_ix_j + \sum_{k\neq j} a_{jk}x_jx_k + \sum_{i\neq j,k\neq j} a_{ik}x_ix_k)}{\partial x_j} \\
&= a_{jj}\frac{\partial x_j^2}{\partial x_j} + \sum_{i\neq j} a_{ij}\frac{\partial(x_ix_j)}{\partial x_j} + \sum_{k\neq j} a_{jk}\frac{\partial(x_jx_k)}{\partial x_j} + \sum_{i\neq j,k\neq j} a_{ik}\frac{\partial(x_ix_k)}{\partial x_j} \\
&= 2a_{jj}x_j + \sum_{i\neq j} a_{ij}x_i + \sum_{k\neq j} a_{jk}x_k + 0 \\
&= \sum_i a_{ij}x_i + \sum_k a_{jk}x_k
\end{aligned}$$

Matrix differentiation (cont'd)

- First order differentiation of quadratic form:

$$\frac{\partial \mathbf{x}^\top A \mathbf{x}}{\partial \mathbf{x}} = (A + A^\top)\mathbf{x}$$

- Let W be a symmetric matrix, it can be easily shown that

$$\begin{aligned}
\tfrac{\partial}{\partial \mathbf{s}}(\mathbf{x} - A\mathbf{s})^\top W(\mathbf{x} - A\mathbf{s}) &= -2A^\top W(\mathbf{x} - A\mathbf{s}) \\
\tfrac{\partial}{\partial \mathbf{s}}(\mathbf{x} - \mathbf{s})^\top W(\mathbf{x} - \mathbf{s}) &= -2W(\mathbf{x} - \mathbf{s}) \\
\tfrac{\partial}{\partial \mathbf{x}}(\mathbf{x} - A\mathbf{s})^\top W(\mathbf{x} - A\mathbf{s}) &= 2W(\mathbf{x} - A\mathbf{s})
\end{aligned}$$

Matrix differentiation (cont'd)

- Second order derivative of quadratic form:

$$\frac{\partial^2(\mathbf{x}^\top A\mathbf{x})}{\partial x_s \partial x_j} = \sum_i a_{ij}\frac{\partial x_j}{\partial x_s} + \sum_k a_{jk}\frac{\partial x_k}{\partial x_s} = a_{sj} + a_{js}$$

$$\frac{\partial^2(\mathbf{x}^\top A\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}^\top} = A + A^\top$$

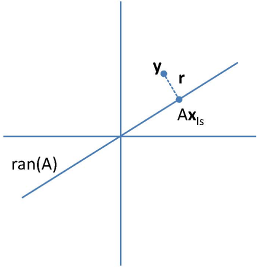
- Recall

$$f(\mathbf{x}) \approx f(\mathbf{a}) + J(\mathbf{a})(\mathbf{x}-\mathbf{a}) + \frac{1}{2}(\mathbf{x}-\mathbf{a})^\top H(\mathbf{a})(\mathbf{x}-\mathbf{a})$$

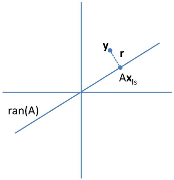
- See *The Matrix Cookbook* by Kaare Petersen and Michael Pedersen (`http://matrixcookbook.com`) for details

Overdetermined linear equations

- Consider $\mathbf{y} = A\mathbf{x}$ where $A \in \mathbb{R}^{m \times n}$ is skinny, i.e., $m > n$
- One can approximately solve $y \approx A\mathbf{x}$, and define residual or error $\mathbf{r} = A\mathbf{x} - \mathbf{y}$
- Find $\mathbf{x} = \mathbf{x}_{ls}$ that minimizes $\|\mathbf{r}\|$
- $\mathbf{x}_{ls}$ is the least squares solution
- Geometric interpretation: $A\mathbf{x}_{ls}$ is the point in $\text{ran}(A)$ that is closest to $\mathbf{y}$, i.e., $A\mathbf{x}_{ls}$ is the projection of $\mathbf{y}$ onto $\text{ran}(A)$



Least squares minimization



- Minimize norm of residual squared

$$\begin{aligned} \mathbf{r} &= A\mathbf{x} - \mathbf{y} \\ \|\mathbf{r}\|^2 &= \mathbf{x}^\top A^\top A\mathbf{x} - 2\mathbf{y}^\top A\mathbf{x} + \mathbf{y}^\top \mathbf{y} \end{aligned}$$

- Set gradient with respect to $\mathbf{x}$ to zero

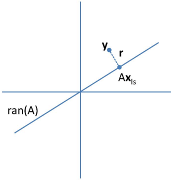
$$\nabla_{\mathbf{x}} \|\mathbf{r}\|^2 = 2A^\top A\mathbf{x} - 2A^\top \mathbf{y} = 0 \Rightarrow \; A^\top A\mathbf{x} = A^\top \mathbf{y}$$

(also known as normal equations)

- Assume $A^\top A$ is invertible, we have

$$\begin{aligned} \mathbf{x}_{ls} &= (A^\top A)^{-1} A^\top \mathbf{y} \\ A\mathbf{x}_{ls} &= A(A^\top A)^{-1} A^\top \mathbf{y} \end{aligned}$$

Least squares minimization



$$\mathbf{y} = A\mathbf{x} \;\Rightarrow\; \mathbf{x}_{ls} = (A^\top A)^{-1} A^\top \mathbf{y}$$

- $\mathbf{x}_{ls}$ is linear function of $\mathbf{y}$
- $\mathbf{x}_{ls} = A^{-1}\mathbf{y}$ if A is square
- $\mathbf{x}_{ls}$ solves $\mathbf{y} = A\mathbf{x}_{ls}$ if $\mathbf{y} \in \mathrm{ran}(A)$
- $A^\dagger = (A^\top A)^{-1} A^\top$ is called pseudo inverse or Moore-Penrose inverse
- $A^\dagger$ is a left inverse of (full rank, skinny) A:

$$A^\dagger A = (A^\top A)^{-1} A^\top A = I$$

- $A(A^\top A)^{-1} A^\top$ is the projection matrix

Orthogonality principle

- Optimal residual

$$\mathbf{r} = A\mathbf{x}_{ls} - \mathbf{y} = (A(A^\top A)^{-1}A^\top - I)\mathbf{y}$$

which is orthogonal to $\mathrm{ran}(A)$:

$$\langle \mathbf{r}, A\mathbf{z} \rangle = \mathbf{y}^\top (A(A^\top A)^{-1}A^\top - I)^\top A\mathbf{z} = 0$$

for all $\mathbf{z} \in \mathbb{R}^n$

- Since $\mathbf{r} = A\mathbf{x}_{ls} - \mathbf{y} \perp A(\mathbf{x} - \mathbf{x}_{ls})$ for any $\mathbf{x} \in \mathrm{ran}(A)$, we have

$$\|A\mathbf{x} - \mathbf{y}\|^2 = \|(A\mathbf{x}_{ls} - \mathbf{y}) + A(\mathbf{x} - \mathbf{x}_{ls})\|^2 = \|A\mathbf{x}_{ls} - \mathbf{y}\|^2 + \|A(\mathbf{x} - \mathbf{x}_{ls})\|^2$$

which means for $\mathbf{x} \neq \mathbf{x}_{ls}$, $\|A\mathbf{x} - \mathbf{y}\| > \|A\mathbf{x}_{ls} - \mathbf{y}\|$

- Can be further simplified via QR decomposition

Least squares minimization and orthogonal projection

- Recall if $\mathbf{u} \in \mathbb{R}^m$, then $P = \frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}}$ is an orthogonal projection
- Given a point $\mathbf{x} = \mathbf{x}_\parallel + \mathbf{x}_\perp$, its projection is

$$P_\mathbf{u}\mathbf{x} = \mathbf{u}\mathbf{u}^\top\mathbf{x}_\parallel + \mathbf{u}\mathbf{u}^\top\mathbf{x}_\perp = \mathbf{x}_\parallel$$

- Generalize to orthogonal projections on a subspace spanned by a set of orthonormal basis $A = [\mathbf{u}_1, \ldots, \mathbf{u}_r]$

$$P_A = AA^\top$$

- In general, we need a normalization term for orthogonal projection if $\mathbf{u}_1, \ldots, \mathbf{u}_r$ is not orthonormal basis,

$$P_A = A(A^\top A)^{-1}A^\top$$

- Given $A = U\Sigma V^\top$, it follows that $P_A = UU^\top$ by least squares minimization

Least squares estimation

- Numerous applications in inversion, estimation and reconstruction problems have the form

$$\mathbf{y} = A\mathbf{x} + \mathbf{v}$$

  - $\mathbf{x}$ is what we want to estimate or reconstruct
  - $\mathbf{y}$ is our sensor measurements
  - $\mathbf{v}$ is unknown noise or measurement error
  - i-th row of A characterizes i-th sensor
- Least squares estimation: choose $\hat{\mathbf{x}}$ that minimizes $\|A\hat{\mathbf{x}} - \mathbf{y}\|$, i.e., deviation between
  - what we actually observe $\mathbf{y}$, and
  - what we would observe if $\mathbf{x} = \hat{\mathbf{x}}$, and there were no noise ($\mathbf{v} = 0$)

  least squares estimate is

$$\hat{\mathbf{x}} = (A^\top A)^{-1} A^\top \mathbf{y}$$

Best linear unbiased estimator (BLUE)

- Linear estimator with noise: $\mathbf{y} = A\mathbf{x} + \mathbf{v}$ with A is a full rank and skinny
- A linear estimator of form $\hat{\mathbf{x}} = B\mathbf{y}$, is unbiased if $\hat{\mathbf{x}} = \mathbf{x}$ whenever $\mathbf{v} = 0$ (no estimator error when $\mathbf{v} = 0$)
- Equivalent to $BA = I$, i.e., B is the left inverse of A
- Estimator error of unbiased linear estimator is

$$\mathbf{x} - \hat{\mathbf{x}} = \mathbf{x} - B(A\mathbf{x} + \mathbf{v}) = -B\mathbf{v}$$

- It follows that $A^{\dagger} = (A^{\top}A)^{-1}A^{\top}$ is the smallest left inverse of A such that for any B with $BA = I$, we have

$$\sum_{i,j} B_{ij}^2 \geq \sum_{i,j} A_{ij}^{\dagger\,2}$$

i.e., least squares provides the best linear unbiased estimator (BLUE)

Pseudo inverse via regularization

- For $\mu > 0$, let $\mathbf{x}_\mu$ be unique minimizer of

$$\|A\mathbf{x} - \mathbf{y}\|^2 + \mu\|\mathbf{x}\|^2 = \left\| \begin{bmatrix} A \\ \sqrt{\mu} I \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \right\|^2 = \left\| \widetilde{A}\mathbf{x} - \widetilde{\mathbf{y}} \right\|^2$$

  thus

$$\begin{aligned} \mathbf{x}_\mu &= (\widetilde{A}^\top \widetilde{A})^{-1} \widetilde{A}^\top \widetilde{\mathbf{y}} \\ &= (A^\top A + \mu I)^{-1} A^\top \mathbf{y} \end{aligned}$$

  is called regularized least squares solution for $A\mathbf{x} \approx \mathbf{y}$
- Also called Tikhonov (Tychonov) regularization (ridge regression in statistics)
- As $A^\top A + \mu I > 0$ and so is invertible, then we have

$$\lim_{\mu \to \mathbf{0}} \mathbf{x}_\mu = A^\dagger \mathbf{y}$$

  and

$$\lim_{\mu \to \mathbf{0}} (A^\top A + \mu I)^{-1} A^\top = A^\dagger$$

Minimizing weighted-sum objective

- Two (or more) objectives:
  - want $J_1 = \|A\mathbf{x} - \mathbf{y}\|^2$ small
  - and also $J_2 = \|F\mathbf{x} - \mathbf{g}\|^2$ small
- Consider minimize a weighted-sum objective

$$\|A\mathbf{x}-\mathbf{y}\|^2+\mu\|F\mathbf{x}-\mathbf{g}\|^2 = \left\| \begin{bmatrix} A \\ \sqrt{\mu}F \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{y} \\ \sqrt{\mu}\mathbf{g} \end{bmatrix} \right\|^2 = \left\| \widetilde{A}\mathbf{x} - \widetilde{\mathbf{y}} \right\|^2$$

- Thus, the least squares solution is

$$\mathbf{x} = (\widetilde{A}^\top \widetilde{A})^{-1}\widetilde{A}^\top \widetilde{\mathbf{y}} = (A^\top A + \mu F^\top F)^{-1}(A^\top \mathbf{y} + \mu F^\top \mathbf{g})$$

- Widely used function approximation, regression, optimization, image processing, computer vision, control, machine learning, graph theory, etc.

Least squares data fitting

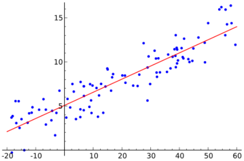
- Linear regression: Model one scalar y in terms of linear combination of $t_1, \ldots, t_n$

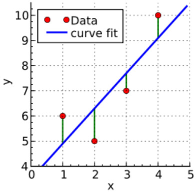
$$y = \alpha_0 + \alpha_1 t_1 + \cdots + \alpha_n t_n = \sum_{j=1}^{n+1} \alpha_i t_j$$

where α_j are unknown parameters or coefficients

- For a set of m data points, $\{(\mathbf{t}_i, y_i)\}$, $\mathbf{t} \in \mathbb{R}^n$, want to minimize

$$\sum_{i=1}^{m} (y_i - \sum_{j=1}^{n+1} t_{ij}\alpha_j)^2$$





Least squares data fitting

- For a set of training data, $\{(\mathbf{t}_i, y_i)\}$, we form $\mathbf{y}$ and A
- In matrix form, denote A by $m \times (n+1)$ matrix with each row an input vector, and $\mathbf{x} \in \mathbb{R}^{n+1}$,

$$\mathbf{y} = A\mathbf{x} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{bmatrix} \quad A = \begin{bmatrix} 1 & t_{11} & t_{12} & \dots & t_{1n} \\ 1 & t_{21} & t_{22} & \dots & t_{2n} \\ 1 & \cdots & & & \\ 1 & t_{m1} & t_{m2} & \dots & t_{mn} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}$$

and thus we obtain the coefficients α_i from $\mathbf{x}$, where

$$\mathbf{x} = A^\dagger \mathbf{y} = (A^\top A)^{-1} A^\top \mathbf{y}$$

and

$$y = \alpha_0 + \alpha_1 t_1 + \cdots + \alpha_n t_n = \sum_{j=1}^{n+1} \alpha_i t_j$$

Least squares data fitting (cont'd)

- Estimate the relationship of weight loss (y) and storage time (t_1) and storage temperature (t_2) with $y = \alpha_0 + \alpha_1 t_1 + \alpha_2 t_2$

Time	1	1	1	2	2	2	3	3	3
Temp	-10	-5	0	-10	-5	0	-10	-5	0
Loss	.15	.18	.20	.17	.19	.22	.20	.23	.25

- Least squares solution is found by

$$A = \begin{bmatrix} 1 & 1 & -10 \\ 1 & 1 & -5 \\ \ldots & & \\ 1 & 3 & 0 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} .15 \\ .18 \\ \ldots \\ .25 \end{bmatrix}$$

- Using MATALB: $\mathbf{x} = A \backslash \mathbf{y} = [.174 \quad .025 \quad .005]^\top$

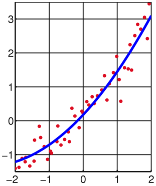
$$y = .174 + .025 t_1 + .005 t_2$$

Least squares polynomial fitting

- Fit polynomial of degree $n-1$, $n \leq m$

$$y = p(t) = \alpha_0 + \alpha_1 t + \alpha_2 t^2 + \cdots + \alpha_{n-1} t^{n-1}$$

  with data (y_i, t_i)
- Basis functions are $f_j(t) = t^{j-1}$, $j = 1, \ldots, n$ (using geometric progression)
  - Straight line: $p(t) = \alpha_0 + \alpha_1 t_1$
  - Quadratic: $p(t) = \alpha_o + \alpha_1 t_1 + \alpha_2 t_2^2$



- Cubic, quartic, and higher polynomials

Least squares polynomial fitting

- Matrix A has form $A_{ij} = t_i^{j-1}$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_m \end{bmatrix} \quad A = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \cdots & & & & \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \cdots \\ \alpha_{n-1} \end{bmatrix}$$

  (called a Vandermonde matrix)
- See also kernel regression and splines

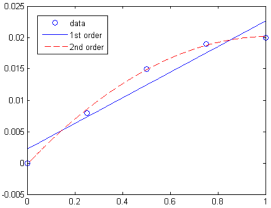
Least squares polynomial fitting (cont'd)

- Estimate the relationship between range of height of a missile

Position	0	250	500	750	1000
Height	0	8	15	19	20

$$A = \begin{bmatrix} 1 & .25 & .25^2 \\ 1 & .5^2 & .5^2 \\ 1 & .75 & .75^2 \\ 1 & 1 & 1^2 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 0 \\ 0.008 \\ 0.015 \\ 0.019 \\ 0.02 \end{bmatrix}$$

$$f(t) = -0.0002286 + 0.03983t - 0.01943t^2$$



Applications

- Thin plate spline: model/morph non-rigid motion