EE445 Mod1-Lec4: Linear Algebra IV

References:

- [VMLS]: Chapters 8, 10

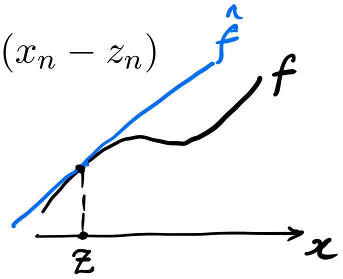
Linear/affine function models, $\mathbf{R}^n \mapsto \mathbf{R}^m$

- in many applications, relations between n-vectors and m-vectors are approximated as linear or affine: *linearization*
- sometimes the approximation is excellent, and holds over large ranges of the variables (e.g., electromagnetics,...)
- sometimes the approximation is reasonably good over smaller ranges (e.g., aircraft dynamics,...)

First-order Taylor approximation, $m = 1$

- suppose $f : \mathbf{R}^n \mapsto \mathbf{R}$. first-order Taylor approximation of f near point z:

$$\begin{aligned} \hat{f}(x) &= f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + \ldots + \frac{\partial f}{\partial x_n}(z)(x_n - z_n) \\ &= f(z) + \nabla f(z)^T (x - z) \end{aligned}$$



- where $\nabla f(z) = \left(\frac{\partial f}{\partial x_1}(z), \; \ldots \;, \frac{\partial f}{\partial x_n}(z) \right)$ is the gradient
- $\hat{f}$ is an affine function of x
- second-order Taylor adds quadratic term using second derivative (the *Hessian* matrix):

$$\nabla^2 f(z) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(z) & \ldots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(z) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(z) & \ldots & \frac{\partial^2 f}{\partial x_n^2}(z) \end{bmatrix}$$

First-order Taylor approximation, $m = 1$

- suppose $f : \mathbf{R}^n \mapsto \mathbf{R}$. first-order Taylor approximation of f near point z:

$$
\begin{aligned}
\hat{f}(x) &= f(z) + \frac{\partial f}{\partial x_1}(z)(x_1 - z_1) + \ldots + \frac{\partial f}{\partial x_n}(z)(x_n - z_n) \\
&= f(z) + \nabla f(z)^T (x - z)
\end{aligned}
$$

ex: $f(x_1, x_2) = x_1^2 + x_2^3$

$\nabla^2 f = \begin{bmatrix} 2 & 0 \\ 0 & 6x_2 \end{bmatrix}$

- where $\nabla f(z) = \left(\frac{\partial f}{\partial x_1}(z), \ \ldots \ , \frac{\partial f}{\partial x_n}(z) \right)$ is the gradient
- $\hat{f}$ is an affine function of x
- second-order Taylor adds quadratic term using second derivative (the *Hessian* matrix):

$$
\nabla^2 f(z) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(z) & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n}(z) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(z) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(z) \end{bmatrix}_{n \times n}
$$

ex: $f(x_1, x_2) = x_1^2 + x_2^3 + x_1 x_2$

$\nabla^2 f = \begin{bmatrix} 2 & 1 \\ 1 & 6x_2 \end{bmatrix}$

First-order Taylor approximation, $m > 1$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \longrightarrow \begin{bmatrix} f_1(x) \\ \vdots \\ f_m(x) \end{bmatrix} = f(x)$$

- suppose $f : \mathbf{R}^n \mapsto \underline{\mathbf{R}^m}$ is differentiable
- first-order Taylor approximation $\hat{f}$ of f near z, for $i = 1, \ldots, m$:

$$\begin{aligned} \hat{f}_i(x) &= f_i(z) + \frac{\partial f_i}{\partial x_1}(z)(x_1 - z_1) + \ldots + \frac{\partial f_i}{\partial x_n}(z)(x_n - z_n) \\ &= f_i(z) + \nabla f_i(z)^T(x - z) \end{aligned}$$

$$\begin{bmatrix} \hat{f}_1(x) \\ \vdots \\ \hat{f}_m(x) \end{bmatrix} = \begin{bmatrix} \nabla f_1(z)^T \\ \vdots \\ \nabla f_m(z)^T \end{bmatrix} (x - z) + \begin{bmatrix} f_1(z) \\ \vdots \\ f_m(z) \end{bmatrix}$$

- putting these together for $i = 1, \ldots, m$:

$$\hat{f}(x) = f(z) + Df(z)(x - z)$$

- $Df(z)$ is the $m \times n$ derivative or *Jacobian* matrix of f at z
- $\hat{f}(x)$ is an affine function of x, and an approximation of $f(x)$ for x near z

$$\hat{f}(x) = (Df(z))\,\underline{x} + \left(f(z) - Df(z)\,z\right)$$

Regression model

- recall: regression model: $\hat{y} = x^T\beta + v$
  - x: n-vector of features/regressors
  - β: n-vector of model parameters, v is offset parameter
  - (scalar) $\hat{y}$ is our prediction of y

- now suppose we have N samples $x^{(1)}, \ldots, x^{(N)}$ and corresponding $y^{(1)}, \ldots, y^{(N)}$
- and predictions: $\hat{y}^{(i)} = (x^{(i)})^T\beta + v$

- write as: $\hat{y} = X^T\beta + v\mathbf{1}$
  - X is feature matrix with columns $x^{(1)}, \ldots, x^{(N)}$
  - $\hat{y}$ is N-vector of predictions $\hat{y}^{(1)}, \ldots, \hat{y}^{(N)}$
  - *prediction error* (vector) is $y - \hat{y} = y - X^T\beta - v\mathbf{1}$

Regression model

- recall: regression model: $\hat{y} = x^T\beta + v$
  - x: n-vector of features/regressors
  - β: n-vector of model parameters, v is offset parameter
  - (scalar) $\hat{y}$ is our prediction of y

- now suppose we have N samples $x^{(1)}, \ldots, x^{(N)}$ and corresponding $y^{(1)}, \ldots, y^{(N)}$
- and predictions: $\hat{y}^{(i)} = (x^{(i)})^T\underline{\beta} + \underline{v}$

- write as: $\hat{y} = \underline{X^T}\beta + v\mathbf{1}$
  - X is feature matrix with columns $\underline{x^{(1)}}, \ldots, \underline{x^{(N)}}$
  - $\hat{y}$ is N-vector of predictions $\hat{y}^{(1)}, \ldots, \hat{y}^{(N)}$
  - *prediction error* (vector) is $y - \hat{y} = y - X^T\beta - v\underline{\mathbf{1}}$

Systems of linear equations

- the simplest problem with a linear model is: a system of m linear equations in n variables:

$$\begin{aligned} A_{11}x_1 + \ldots + A_{1n}x_n &= b_1 \\ &\vdots \quad \vdots \\ A_{m1}x_1 + \ldots + A_{mn}x_n &= b_m \end{aligned}$$

- express compactly as: $\underline{Ax = b}$
- will later see an approximate version: make $\|Ax - b\|^2$ small (cf. lectures on regression)

$Ax^{(1)} = b^{(1)}$

$Ax^{(2)} = b^{(2)}$

$\vdots$

- multiple sets of linear eq.'s with same A: $AX = B$

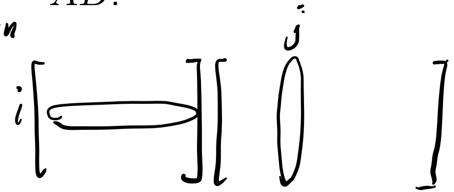
$$\underset{m\times n}{A}\underset{n\times p}{\left[x^{(1)} \cdots x^{(p)}\right]} = \underset{m\times p}{\left[b^{(1)} \cdots b^{(p)}\right]}$$

Matrix multiplication reminder

- multiplying $m \times p$ matrix A and $p \times n$ matrix B to get $C = AB$:

  m×n

$$C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj}$$



- special cases:
  - inner product: $a^T b$
  - outer product:

$$ab^T = \begin{bmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \dots & a_2 b_n \\ \vdots & \vdots & & \vdots \\ a_m b_1 & a_m b_2 & \dots & a_m b_n \end{bmatrix}_{m \times n}$$

Properties

- properties:
  - $(AB)C = A(BC) = ABC$
  - $A(B + C) = AB + AC$
  - $(AB)^T = B^T A^T$
  - $AB = BA$ does NOT hold in general
- block matrices multiplied similarly (provided all products make sense):

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

- column interpretation of matrix product: $B = \begin{bmatrix} | & | & & | \\ b_1 & b_2 & \dots & b_n \\ | & | & & | \end{bmatrix}$, then

$$AB = A \begin{bmatrix} b_1 & b_2 & \dots & b_n \end{bmatrix} = \begin{bmatrix} Ab_1 & Ab_2 & \dots & Ab_n \end{bmatrix}$$

Inner product interpretation, Gram matrix

- with a_i denoting rows of A and b_j denoting columns of B:

$$AB = \begin{bmatrix} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_n \\ a_2^T b_1 & & & \\ & & \ddots & \\ & & & a_m^T b_n \end{bmatrix}$$

all inner products of rows of A & col's of B arranged in a matrix

- let C be $m \times n$ with columns $c_1, \ldots, c_n$, the *Gram matrix* of C is

$\|c_1\|^2$

$$G = C^T C = \begin{bmatrix} c_1^T c_1 & c_1^T c_2 & \cdots & c_1^T c_n \\ \vdots & & \ddots & \\ & & & c_n^T c_n \end{bmatrix}_{n \times n}$$

- if $C^T C = I$, what does this mean about columns of C? $c_1, \ldots, c_n$ are orthonormal

Outer product interpretation

- Gram matrix example:
  suppose $m \times n$ matrix C gives the membership of m items in n groups:

$$C_{ij} = \begin{cases} 1 & \text{item } i \text{ is in group } j \\ 0 & \text{item } i \text{ is not in group } j \end{cases}$$

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & & & 5 & \end{array} \\ i & \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

  here C^TC gives: $(C^TC)_{ij} = c_i^T c_j = \#$ of items in both groups i & j

- outer product interpretation of $\underset{m\times n}{A}\underset{n\times p}{B}$: $(C^TC)_{ii} = \#$ of items in group i

$$\begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} \begin{bmatrix} - b_1^T - \\ \vdots \\ - b_n^T - \end{bmatrix} = a_1 b_1^T + \cdots + a_n b_n^T = \sum_{i=1}^{n} a_i b_i^T$$

Composition of linear functions

- consider $f : \mathbf{R}^p \mapsto \mathbf{R}^m$ with $f(u) = Au$, and $g : \mathbf{R}^n \mapsto \mathbf{R}^p$ with $g(v) = Bv$
- $h : \mathbf{R}^n \mapsto \mathbf{R}^m$ with $h(x) = f(g(x))$ can be expressed as

$$h(x) = A(Bx) = (AB)x$$

$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \xrightarrow{D_n} \begin{bmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{bmatrix}$$

- example: 2nd-difference matrix

$$D_n x = (x_2 - x_1, \ldots, x_n - x_{n-1}), \quad D_{n-1} y = (y_2 - y_1, \ldots, y_{n-1} - y_{n-2})$$

then $D_{n-1} D_n$ gives $\quad (x_1 - 2x_2 + x_3, \ldots, x_{n-2} - 2x_{n-1} + x_n)$

$$\underset{3\times 5}{\begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}} = \underset{3\times 4}{\begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}} \underset{4\times 5}{\begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}}$$

Gram-Schmidt in matrix notation

- run Gram–Schmidt on columns $a_1, \ldots, a_k$ of $n \times k$ matrix A
- if columns are linearly independent, get orthonormal $q_1, \ldots, q_k$

$Q = [q_1 \cdots q_k]$

- define matrix Q with columns q_i; then $Q^T Q = I$
- from G-S algorithm:

$$\begin{aligned} a_i &= (q_1^T a_i) q_1 + \ldots + (q_{i-1}^T a_i) q_{i-1} + \|\tilde{q}_i\| q_i \\ &= R_{1i} q_1 + \ldots + R_{ii} q_i \end{aligned}$$

with $R_{ij} = q_i^T a_j$ for $i < j$, and $R_{ii} = \|\tilde{q}_i\|$. let $R_{ij} = 0$ for $i > j$

QR factorization

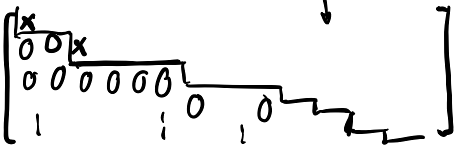
- $A = QR$ is called QR factorization

upper triangular

$$\underbrace{\begin{bmatrix} a_1 & a_2 & \dots & a_k \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} q_1 & q_2 & \dots & q_k \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \dots & r_{1k} \\ 0 & r_{22} & \dots & r_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{kk} \end{bmatrix}}_{R}$$

- $Q^T Q = I_k$, columns of Q are orthonormal basis for the range of A (denoted $\mathcal{R}(A)$)

- modified G-S: if $\tilde{q}_j = 0$, skip to next vector a_{j+1} and continue. on exit:
  - $q_1, \dots, q_r$ are ortho basis for $\mathcal{R}(A)$ (hence $r = \mathbf{Rank}(A)$)
  - R is $r \times k$ in *upper staircase* form:



Matrix Rank

- define rank of $A \in \mathbf{R}^{m \times n}$ as

range = lin. comb. of col's
= span $\{a_1, \cdots, a_n\}$

$$\mathbf{Rank}(A) = \dim \mathcal{R}(A)$$

- $\mathbf{Rank}(A)$ is maximum number of independent columns of A
  - to see this: if columns of A are independent, then number of columns r is the rank, since columns are a basis for the range
  - and if not, there must be one column lin. dependent on others, so remove it, repeat if needed
  - all other independent sets of columns must have no more than r elements.
- $\mathbf{Rank}(A) = \mathbf{Rank}(A^T)$, can prove using QR

Conservation of dimension

$\mathcal{N}(A) = \{x \mid Ax = 0\}$ nullspace $\qquad A \in \mathbb{R}^{m \times n}, \quad x \in \mathbb{R}^n$

$\mathcal{R}(A) = \{y \mid y = Ax \text{ for some } x \in \mathbb{R}^n\}$

$$\underbrace{\dim \mathcal{R}(A)}_{= \text{Rank}(A)} + \underline{\dim \mathcal{N}(A)} = \underline{n}$$

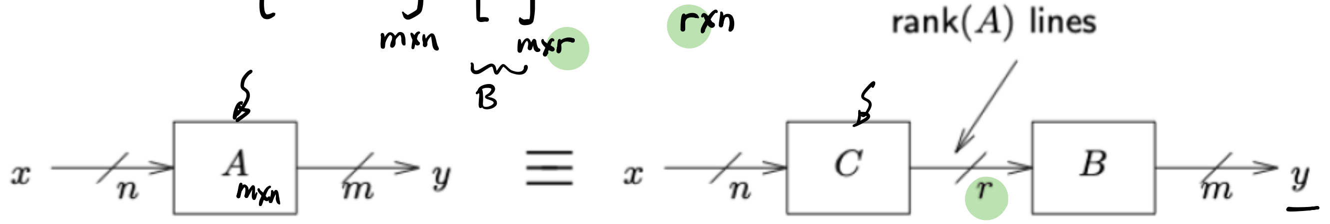
- $\mathbf{Rank}(A)$ is dimension of set 'hit' by the mapping $y = Ax$
- $\dim \mathcal{N}(A)$ is dimension of set of x 'crushed' to zero by $y = Ax$
- *conservation of dimension*: each dimension of input is either crushed to zero or ends up in output
- proof using QR

"Coding" interpretation of rank

$$\mathbf{Rank}(\underline{BC}) \le \min\{\mathbf{Rank}(B), \mathbf{Rank}(C)\}$$

- hence if $A = BC$ with $B \in \mathbf{R}^{m\times r}$, $C \in \mathbf{R}^{r\times n}$, then $\mathbf{Rank}(A) \le r$
- converse: if $\mathbf{Rank}(A) = r$ then $A \in \mathbf{R}^{m\times n}$ factors as $A = BC$ with $B \in \mathbf{R}^{m\times r}$, $C \in \mathbf{R}^{r\times n}$
- $\mathbf{Rank}(A) = r$ is <u>minimum size vector needed to faithfully reconstruct y from x</u>

$$A = \underset{m\times n}{\begin{bmatrix} \quad & \quad \end{bmatrix}} = \underbrace{\underset{m\times r}{\begin{bmatrix} \, \end{bmatrix}}}_{B}\ \overbrace{\underset{r\times n}{\begin{bmatrix} \quad & \quad \end{bmatrix}}}^{C}$$

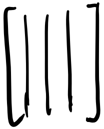


related to: dimensionality reduction methods (later)

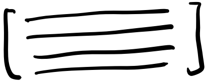
Full-rank matrices

for $A \in \mathbf{R}^{m \times n}$ we always have $\mathbf{Rank}(A) \leq \min\{m, n\}$. we say A is *full-rank* if equality holds

- for **square** matrices, full-rank means non-singular
- for **tall** matrices $(m \geq n)$, means columns are independent



- for **wide** matrices $(m \leq n)$, means rows are independent



Matrix inverse: left-inverse

- a matrix X that satisfies $XA = I$ is called a *left-inverse* of A
- example: the matrix

$$A = \begin{bmatrix} -3 & -4 \\ 4 & 6 \\ 1 & 1 \end{bmatrix}$$

has different left inverses:

$$B = \frac{1}{9}\begin{bmatrix} -11 & -1 & 16 \\ 7 & 8 & -11 \end{bmatrix} \qquad C = \frac{1}{2}\begin{bmatrix} 0 & -1 & 6 \\ 0 & 1 & -4 \end{bmatrix}$$

- if A has a left-inverse, its columns are lin. independent
- to see this: if $Ax = 0$ and $CA = I$, then $\quad 0 = C0 = C(Ax) = (CA)x = x$
- can use left-inverse (when exists) to solve $Ax = b$: $\quad Cb = C(Ax) = (CA)x = x$

$\hookrightarrow x = Cb$