Matrix Methods MIT 18.065

in Data Analysis, Signal Processing, and Machine Learning

2020. 11. 29

Sungchan Yi

Matrix Multiplication.

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 7 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix}}_{U} = l_1 u_1^T + l_2 u_2^T$$

(columns of L: l_1, l_2; rows of U: u_1^T, u_2^T)

$$= \begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 2 & 3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 3 \\ 4 & 6 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 7 \end{bmatrix}$$

Eigenvalue of Rotations

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad \det R_\theta = 1, \quad R_\theta \in O(2)$$

$$R_{\frac{\pi}{2}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = A \qquad p_A(t) = \lambda^2 + 1 = 0$$

$\therefore \lambda = \pm i$ (Imaginary)

Reflections in $\mathbb{R}^n$

· Reflection always has an axis
(Some hyperplane of $n-1$ dimension)

Take $n-1$ orthonormal vectors of this hyperplane
: $u_1, u_2, \cdots, u_{n-1}$

Then the reflection does not change the vector.
$\therefore S u_i = u_i$ for $i = 1, 2, \cdots, n-1$

$\Rightarrow$ 1 is an eigenvalue. (multiplicity $n-1$)

$\exists v \in (\text{hyperplane})^\perp$ then $Sv = -v$.

$\Rightarrow$ -1 is an eigenvalue

$\therefore$ S has a full set of independent eigenvectors.

S is diagonalizable

$S \sim \operatorname{diag}(-1, 1, \cdots, 1)$, $\det S = -1$.

Orthogonal Matrices have Orthogonal Eigenvectors

Proof. Let $Q \in O(n)$, λ, μ be eigenvalues of Q, where $\lambda \neq \mu$.

$\exists v, w \in \mathbb{C}^n$ s.t $Qv = \lambda v$, $Qw = \mu w$

We show that $v^* w = 0$.

$$Qv = \lambda v \Rightarrow v^* Q^* = \bar{\lambda} v^*$$
$$\Rightarrow v^* Q^* Q w = \bar{\lambda} v^* Q w$$
$$\Rightarrow v^* w = \bar{\lambda} \mu v^* w$$

Since eigenvalues of orthogonal matrices have length 1, $\bar{\lambda}\mu \neq 0$ and $v^* w = 0$ ■

$(\det Q = \pm 1)$

Singular Value Decomposition.

$A \in M_{m,n}(F)$

$$\begin{cases} Av_1 = \sigma_1 u_1 \\ Av_2 = \sigma_2 u_2 \\ \quad \vdots \\ Av_r = \sigma_r u_r \end{cases} \qquad (v_i \in F^n,\ u_i \in F^m) \quad (\sigma_1 \geq \sigma_2 \geq \cdots > 0)$$

$$\underset{m\times n}{A} \underbrace{\begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix}}_{V,\ n\times r} = \underbrace{\begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}}_{U,\ m\times r} \underset{r\times r}{\Sigma}$$

$\Rightarrow AV = U\Sigma$.

Because V is orthogonal, $A = U\Sigma V^T$

Such U, V exist because AA^T, A^TA are positive semi-definite
→ Has orthogonal eigenvectors. (basis)

Singular values of A are $\sqrt{\text{eigenvalues}}$ of AA^T

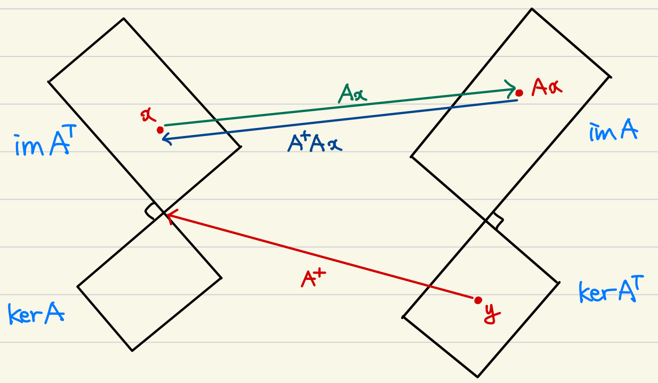
$$AA^T = U\Sigma V^T V \Sigma^T U^T = U\Sigma\Sigma^T U^T$$
$$A^TA = V\Sigma^T U^T U \Sigma V^T = V\Sigma^T\Sigma V^T$$

$r = \operatorname{rk} A$, and $\sigma_{r+1}, \cdots, \sigma_{\min(m,n)} = 0$

$$\Rightarrow \underset{m\times n}{A} = \underset{m\times m}{U} \ \underset{m\times n}{\Sigma} \ \underset{n\times n}{V^T}$$

For positive semi-definite matrices, SVD is equal to diagonalization.

Four Subspaces.



$A^+Ax = x$ for all $x \in \operatorname{im} A^T$

but $A^+y = 0$ for all $y \in \ker A^T$

If A is invertible, $A^+ = A^{-1}$.

If A is not invertible, ··· SVD!

$A = U\Sigma V^T$

$\Rightarrow A^+ = U\Sigma^+ V^T$

$$= U \begin{bmatrix} 1/\sigma_1 & & & \\ & \ddots & & \\ & & 1/\sigma_r & \\ & & & 0 \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix} V^T$$

Matrix Norms.

Spectral Norm. (Maximum singular value)

$$\|A\|_2 = \sigma_1 = \max_{x \in V} \frac{\|Ax\|_2}{\|x\|_2}$$

Frobenius Norm

$$\|A\|_F = \sqrt{\sum_{ij} |a_{ij}|^2} = \sqrt{tr(A^*A)} = \sqrt{\sum_i \sigma_i}$$

Nuclear Norm (Sum of singular values)

$$\|A\|_N = \sum_i |\sigma_i|$$

Least Squares

$A : m \times n$ and $rkA = r$

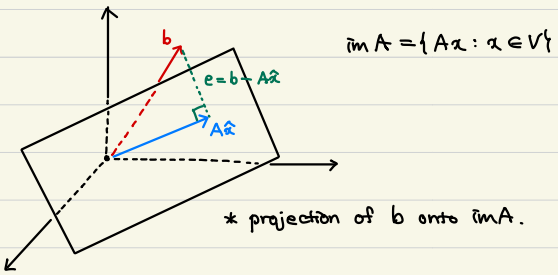
Solve $Ax = b$!

If $m = n = r$, $x = A^{-1}b$.

Otherwise, we minimize $\|Ax - b\|_2^2$

$$\|Ax-b\|_2^2 = (Ax-b)^T(Ax-b)$$
$$= x^TA^TAx - 2b^TAx + b^Tb$$

$\Rightarrow$ has minimum at $A^TA\hat{x} = A^Tb$



But this only works when A has independent columns.

$\rightarrow A^TA$ is invertible.

Proof. A has independent columns $\Rightarrow \ker A = 0$.

Suppose $A^TAx = 0$ then $Ax \in \ker A^T$.

since $\ker A^T \perp \text{im}A$, and $Ax \in \text{im}A$,

$Ax \in \ker A^T \cap \text{im}A = \{0\}$. $\therefore Ax = 0$.

But $\ker A = 0$ thus $x = 0$.

$\therefore \ker A^TA = 0$ and A^TA is invertible.

If $\ker A = 0$, (or $rkA = r = n$)

$A^+ = (A^TA)^{-1}A^T$

(Note that AA^T is not invertible)

If A has independent rows, AA^T is invertible.

$A^+ = A^T(AA^T)^{-1}$

Low Rank Approximation.

Minimize $\|A - A_k\|_2$ over A_k subject to $rkA_k = k$.

Eckart-Young-Mirsky Theorem.

Let $A = U\Sigma V^T \in M_{m\times n}(\mathbb{R})$

Define $U_1 \in M_{m\times k}(\mathbb{R})$, $\Sigma_1 \in M_{k,k}(\mathbb{R})$.

$V_1 \in M_{n\times k}(\mathbb{R})$ as

$$U = \begin{bmatrix} U_1 & * \end{bmatrix} \quad \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & * \end{bmatrix} \quad V = \begin{bmatrix} V_1 & * \end{bmatrix}$$

Then, $A_k = U_1\Sigma_1 V_1^T$.

Proof. WLOG. $m \geq n$.

Let $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_k, \cdots, \sigma_m)$

$$A = \sum_{i=1}^{n} \sigma_i u_i v_i^T, \quad A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Thus $\|A - A_k\|_2 = \left\| \sum_{i=k+1}^{n} \sigma_i u_i v_i^T \right\|_2 = \sigma_{k+1}$.

Suppose $B = XY^T$ where X, Y have k columns.

$\Rightarrow rkB = k$.

We show that $\|A - A_k\|_2 = \sigma_{k+1} \leq \|A - B\|_2$

Since Y has k columns, there exists a non-trivial linear combination of the first $k+1$ columns of V,

$w = c_1v_1 + \cdots + c_{k+1}v_{k+1}$ such that $Y^Tw = 0$.

WLOG, scale $\|w\|_2 = 1$ $(c_1^2 + \cdots + c_{k+1}^2 = 1)$

Then, $\|A - B\|_2^2 \geq \|(A-B)w\|_2^2$

$$= \|Aw\|_2^2$$
$$= c_1^2\sigma_1^2 + \cdots + c_{k+1}^2\sigma_{k+1}^2$$
$$= \sigma_{k+1}^2 + c_1^2(\sigma_1^2 - \sigma_{k+1}^2) + \cdots$$
$$\geq \sigma_{k+1}^2 \quad \blacksquare$$

10. Survey of Difficulties with $Ax=b$

⓪ Always works. $x = A^+ b$.

① Good normal case : Square matrix

Condition number σ_1/σ_n is OK. $x = A \backslash b$

② $m > n$ = rank $A^TA\hat{x} = A^Tb$

A^TA is not to big (reasonable to compute)

③ $m < n$ (Not enough equations - Deep Learning)

· minimum norm A^+b (ℓ^2 norm)

· min ℓ^1 norm → pick some algorithm and solve.

"Good algorithm" ⇒ Good results in test data.

under determined / many solutions

④ Columns are nearly almost linearly dependent

(In bad condition) → Gram Schmidt

$A = QR$ (Q: orthogonal, R: triangular)

• Column pivoting : reordering columns

If pivot is too slow, then reorder the rows. $PA = LU$

⑤ Nearly singular / Inverse problems ⇒ Add penalty.

· Minimize $\|Ax-b\|^2 + \delta^2\|x\|^2$ (try to make it invertible)

⑥ Too big → Iterative methods (conjugate gradient)

⑦ Way too big (Not in core, giant problem)

→ Randomized numerical linear algebra

Sample the columns/rows of A …

SVD : $A = U\Sigma V^T \rightarrow A^+ = V\Sigma^+ U^T$

Nearly singular → small σ_i s

→ Σ^{-1} has large singular values.

ℓ^2 norm : minimize $\|Ax-b\|^2 + \delta^2\|x\|^2$

Choose $\delta > 0$ → Solvable problem.

$$\begin{bmatrix} A \\ \delta I \end{bmatrix} x = \begin{bmatrix} b \\ 0 \end{bmatrix}$$ (Augmented system)

$A^* x = b^*$ ← Apply least squares here.

• What should δ be?

• What happens as $\delta \to 0$?

$(A^*)^T A^* \hat{x} = (A^*)^T b^*$

$$\Rightarrow \begin{bmatrix} A^T & \delta I \end{bmatrix} \begin{bmatrix} A \\ \delta I \end{bmatrix} \hat{x} = \begin{bmatrix} A^T & \delta I \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix}$$

$\Rightarrow (A^TA + \delta^2 I)\hat{x} = A^Tb$ … (*)

↳ positive definite (invertible)

Suppose A is 1×1 $A = [\sigma]$ $b = [b]$

Then (*) $\Rightarrow (\sigma^2+\delta^2)x = \sigma b$ $x = \frac{\sigma}{\sigma^2+\delta^2} b$

(i) $\sigma > 0$: as $\delta \to 0$, $x \to \frac{b}{\sigma}$ OK.

(ii) $\sigma = 0$: $x = 0$

… Similar to pseudoinverses. $\delta \to 0$, $x \to A^+b$!!!

For any A. $(A^TA + \delta^2 I)^{-1}A^T \to A^+$ as $\delta \to 0$

(Note that we are staying with ℓ^2 norms)

pf) $A = U\Sigma V^T$. $A^TA = V\Sigma^T U^T U \Sigma V^T = V\Sigma^T\Sigma V^T$

$A^TA + \delta^2 I = V\Sigma^T\Sigma V^T + \delta^2 VV^T = V(\Sigma^T\Sigma + \delta^2 I)V^T$

Let $D = \Sigma^T\Sigma + \delta^2 I$ (diagonal, D^{-1} exists).

$\therefore (A^TA + \delta^2 I)^{-1}A^T = VD^{-1}V^T V \Sigma^T U^T = VD^{-1}\Sigma^T U^T$

Observe that $D = (d_{ij})_{n\times n}$ where

$$d_{ij} = \begin{cases} \sigma_i^2 + \delta^2 & \text{if } i=j\le r \\ 0 & \text{otherwise} \end{cases}$$

therefore $D^{-1}\Sigma^T = (c_{ij})_{n\times m}$ where

$$c_{ij} = \begin{cases} \sigma_i / (\sigma_i^2 + \delta^2) & \text{if } i=j\le r \\ 0 & \text{otherwise} \end{cases}$$

Since $\lim_{\delta\to 0} \frac{\sigma_i}{\sigma_i^2+\delta^2} = \frac{1}{\sigma_i}$, $D^{-1}\Sigma^T \to \Sigma^+$

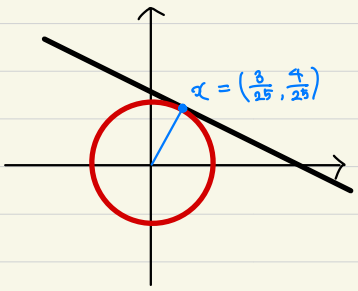
$\therefore (A^TA + \delta^2 I)^{-1}A^T \to V\Sigma^+ U^T = A^+$ as $\delta \to 0$ ∎

11. Minimizing $\|x\|$ Subject to $Ax=b$

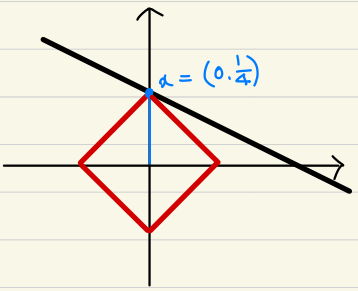
Suppose we want to minimize $\|x\|$,

with $3x_1 + 4x_2 = 1$ (Geometrically)

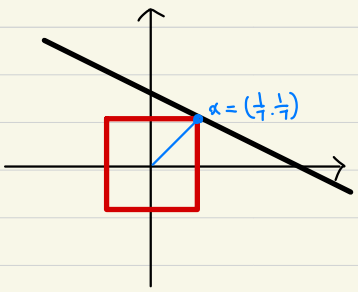
ℓ_2 norm



ℓ_1 norm



ℓ_∞ norm



* Gram - Schmidt

1. Standard way $A \to QR$

2. Column exchanges

3. Krylov - Arnoldi - Lanczos

Let $A = [a_1 \cdots a_n]$.

Gram-Schmidt $A = QR$ where Q is orthogonal

· What is R?

→ Combinations of q_i will give columns of A.

→ Since Q is orthogonal, $Q^{-1} = Q^T$.

$\therefore R = Q^T A = \begin{bmatrix} - q_1^T - \\ \vdots \\ - q_n^T - \end{bmatrix} \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix}$

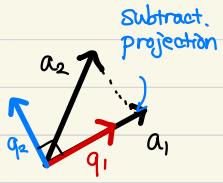
Then (i,j)-component of R should be $q_i^T a_j$

(dot product of q_i and a_j)

Gram-Schmidt Process.

For given n vectors $a_1, \cdots, a_n$.

Let $q_1 = \frac{a_1}{\|a_1\|}$



$u_2 = a_2 - \text{proj}_{q_1} a_2 \Rightarrow q_2 = \frac{u_2}{\|u_2\|}$

$= a_2 - (a_2^T q_1) q_1$

Repeat similarly, ···

$u_3 = a_3 - (a_3^T q_1) q_1 - (a_3^T q_2) q_2$

$\Rightarrow q_3 = \frac{u_3}{\|u_3\|}$

$\therefore u_i = a_i - (a_i^T q_1) q_1 - \cdots - (a_i^T q_{i-1}) q_{i-1}$

$\Rightarrow q_i = \frac{u_i}{\|u_i\|}$ for $i = 1, \cdots, n$.

Check that $q_i^T q_j = \delta_{ij}$ (Kronecker Delta)

2. Column pivoting possible

: How to decide q_2

→ Compute $a_i - (a_i^T q_1) q_1$ for all $i = 2, \cdots, n$ and set u_2 as the maximum..

Now $q_2 = \frac{u_2}{\|u_2\|}$.

Computing all $a_i - (a_i^T q_1) q_1$ isn't costly since $(a_i^T q_1) q_1$ was going to be subtracted anyway.

3. Krylov. : Large, sparse A.

Solve $Ax = b$ ...

Sparse → matrix × vector is easy.

Compute $b, Ab, A^2 b, \cdots, A^{j-1} b$. (iterate $Ax(A^i b)$)

Their combination gives the Krylov space K_j (dimension)

x_j = best vector in K_j (closest least squares solution)

Why is orthonormal basis so good?

Let $x = c_1 q_1 + c_2 q_2 + \cdots + c_n q_n = Q \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = Qc$.

Then $c = Q^{-1} x = Q^T x$. !!!

$q_i^T x = c_i q_i^T q_i = c_i$ (since $q_i^T q_j = 0$ if $i \neq j$)

Arnoldi (Any matrix) - Lanczos (Symmetric matrix) found a great way to orthogonalize the vectors in K_j. (Using standard Gram-Schmidt)

* Trefethen - Bau * Golub - VanLoan

12. Computing Eigenvalues and Singular Values

· Eigenvalues of A. — QR Method.

Let $A_i = Q_i R_i$ (orthogonalize, R: upper triangular)

Set $A_{i+1} = R_i Q_i$ for $i = 0, 1, \cdots$

Claim. $A_i \sim A_{i+1}$

Proof. $A_{i+1} = R_i Q_i = R_i A_i R_i^{-1}$ ■ ← Can't I use Q_i^{-1} ?

After each iteration, components below the diagonals tend to get smaller (in absolute value)

⇒ Then A_k is nearly upper triangular, and has λ_is on the main diagonal.

* Introducing shifts (shifting eigenvalues)

Shift by εI. : $A_i - \varepsilon I = Q_i R_i$, $A_{i+1} = R_i Q_i + \varepsilon I$

→ Eigenvectors do not change, eigenvalues are shifted.

Does $A_i \sim A_{i+1}$ still hold?

$A_{i+1} = R_i Q_i + \varepsilon I = R_i (A_i - \varepsilon I) R_i^{-1} + \varepsilon I$

$= R_i A_i R_i^{-1} - R_i (\varepsilon I) R_i^{-1} + \varepsilon I = R_i A_i R_i^{-1}$ ■

Since λ_n is usually computed first, ε can be set to λ_n.

Non-zeros directly under the main diagonal : Hessenburg Matrix.

* If A_i is nearly upper-triangular → easy to compute.

Computing Eigenvalues.

① Reduce A (by similarity transformation) to Hessenburg form

② Apply QR decomposition with shifts LAPACK

If A is symmetric, its Hessenburg form will be tridiagonal.

→ reduced order of operations.

Symmetric S. then find $Q = Q_0 Q_1 \cdots$ such that QSQ^{-1} is tridiagonal (and has same eigenvalues)

Multiplying orthogonal matrices does not change σ_is.

· σ_is are eigenvalues of $A^T A$.

Then for QA, $(QA)^T QA = A^T Q^T Q A = A^T A$.

thus the singular values of QA are eigenvalues of $A^T A$

Then if $A = U \Sigma V^T$, multiply left / right hand sides by orthogonal matrices Q_1, Q_2 so that

$A' = (Q_1 U) \Sigma (V^T Q_2)$ → bidiagonal !!

Now $(A')^T A$ is tridiagonal, now σ_i can be calculated using the QR decomposition

But these methods are $O(n^3)$

13. Randomized Matrix Multiplication

For matrices that are HUGE !!!

$$AB = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} \begin{bmatrix} - & b_1^T & - \\ & \vdots & \\ - & b_n^T & - \end{bmatrix} = a_1 b_1^T + \cdots + a_n b_n^T$$

IDEA: Choose columns randomly with probability $\sum p_i = 1$.

+ Choose the probability that minimize the variance

$p_i = \|a_i\| \|b_i^T\| / c$ (Normalize.) where $c = \sum_{i=1}^{n} \|a_i\| \|b_i^T\|$

↑ probability of choosing $a_i \times b_i^T$ (to multiply)

the probability is proportional to the norm.

s : number of samples

With probability p_i, choose $\frac{a_i b_i^T}{s p_i}$

Then the approximate AB is $\sum_{i=1}^{s} \frac{a_i b_i^T}{s p_i}$

Mean of 1 sample is $\sum_{i=1}^{n} p_i \frac{a_i b_i^T}{s p_i} = \sum_{i=1}^{n} \frac{a_i b_i^T}{s}$

→ s samples give $s \sum_{i=1}^{n} \frac{a_i b_i^T}{s} = \sum_{i=1}^{n} a_i b_i^T = AB$

Variance (will depend on p_i)

$$\underbrace{\sum_{i=1}^{n} \frac{\|a_i\|^2 \|b_i^T\|^2}{s p_i} - \frac{1}{s}\|AB\|_F^2}_{\text{variance.}} = \sum_{i=1}^{n} \frac{c}{s} \|a_i\| \|b_i^T\| - \frac{1}{s}\|AB\|_F^2$$

$$= \frac{1}{s}\left(c^2 - \|AB\|_F^2\right)$$

→ s samples : $c^2 - \|AB\|_F^2$ (with optimal p_i)

* How to choose p_i such that the variance is minimized?

Minimize $\sum_{i=1}^{n} \frac{\|a_i\|^2 \|b_i^T\|^2}{p_i}$ with the constraint $\sum p_i = 1$

Lagrange Multipliers.

Let $F = \sum_{i=1}^{n} \frac{\|a_i\|^2 \|b_i^T\|^2}{p_i} + \lambda(\sum p_i - 1)$

$\frac{\partial F}{\partial p_i} = -\frac{\|a_i\|^2 \|b_i^T\|^2}{p_i^2} + \lambda = 0 \Rightarrow p_i^2 = \frac{\|a_i\|^2 \|b_i^T\|^2}{\lambda}$ for each $i = 1, \cdots, n$

Since p_i s add up to 1, $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} \frac{\|a_i\| \|b_i^T\|}{\sqrt{\lambda}} = 1$.

$\therefore \sqrt{\lambda} = \sum_{i=1}^{n} \|a_i\| \|b_i^T\| = c$. $\therefore p = \|a_i\| \|b_i^T\| / \sqrt{\lambda} = \frac{\|a_i\| \|b_i^T\|}{c}$ ∎

14. Low Rank Changes in A and its Inverse

Start with I and perturb it by uv^T (rank 1 matrix).

What is the inverse of $I - uv^T$?

→ Matrix Inversion Formula

$$(I - uv^T)^{-1} = I + \frac{uv^T}{1 - v^T u}$$ ($v^T u$ → 1×1.)

→ The inverse also has rank 1 perturbation.

Proof). $(I - uv^T)\left(I + \frac{uv^T}{1 - v^T u}\right)$ ($v^T u$: 1×1 scalar.)

$= I + \frac{uv^T}{1 - v^T u} - uv^T - \frac{uv^T uv^T}{1 - v^T u}$

$= I + \frac{uv^T}{1 - v^T u} - uv^T - \frac{uv^T (v^T u)}{1 - v^T u}$

$= I + \frac{(1 - v^T u) uv^T}{1 - v^T u} - uv^T = I + uv^T - uv^T = I$ ∎

How about for rank k matrix ?

Let $U \in \mathbb{M}_{n \times k}$. $V \in \mathbb{M}_{k \times n}$ UV^T is rank k

* Sherman – Morrison – Woodbury

$$(I_n - UV^T)^{-1} = I_n + U\underbrace{(I_k - V^T U)^{-1}}_{k \times k \text{ inverse}} V^T$$

Proof) $(I_n - UV^T)\left(I_n + U(I_k - V^T U)^{-1} V^T\right)$

$= I_n - UV^T + U(I_k - V^T U)^{-1} V^T - UV^T U(I_k - V^T U)^{-1} V^T$

$= I_n - UV^T + U\{(I_k - V^T U)^{-1} - V^T U(I_k - V^T U)^{-1}\} V^T$

$= I_n - UV^T + U\{(I_k - V^T U)(I_k - V^T U)^{-1}\} V^T$

$= I_n - UV^T + UV^T = I_n$ ∎

Finally, (Woodbury Matrix Identity)

$$(A - UV^T)^{-1} = A^{-1} - A^{-1}U(I_k - V^T A^{-1} U)^{-1} V^T A^{-1}$$

To use the formula.

① Solve $(A - UV^T)x = b$

Suppose $Aw = b$ is solved for w

Also solve $Az = u$. for z

A = LU and back-substitute. (once)

$A[w \; z] = [b \; u]$

Now solve $(A - uv^T)x = b$. for x, quickly!

$\therefore x = w + \frac{zv^T w}{1 - v^T z}$

Proof). $(A - uv^T)\left(w + \frac{zv^T w}{1 - v^T z}\right)$ ($v^T z$: scalar)

$= Aw - uv^T w + A\frac{zv^T w}{1 - v^T z} - uv^T \frac{zv^T w}{1 - v^T z}$

$= b - uv^T w + \frac{uv^T w}{1 - v^T z} - (v^T z)\frac{uv^T w}{1 - v^T z}$

$= b - uv^T w + uv^T w = b$ ∎

② New measurement in least squares !

(OLD) $Ax = b$ ~normal~ $A^T A\hat{x} = A^T b$ (Recursive LSA)

Make 1 more measurement!

$\begin{bmatrix} A \\ v^T \end{bmatrix} x = \begin{bmatrix} b \\ b_{new} \end{bmatrix}$ ← 1 new row.

~normal~ $\begin{bmatrix} A^T & v \end{bmatrix} \begin{bmatrix} A \\ v^T \end{bmatrix} \hat{x}_{new} = \begin{bmatrix} A^T & v \end{bmatrix} \begin{bmatrix} b \\ b_{new} \end{bmatrix}$

This is actually $(A^T A + vv^T)\hat{x}_{new} = A^T b + v b_{new}$

$A^T A$ doesn't change ← vv^T rank 1 change in $A^T A$.

* Kalman Filter : Dynamic least squares.

· Idea of using the covariance matrix. (correlation)

· (Control Theory) State equation.

Least Squares : Standard.

Data not correlated, covariance matrix is I.

???

15. Matrices A(t) and Derivative dA/dt

Setup $A(t)$ is a matrix depending on t.

A is invertible and we know $\frac{dA}{dt}$. What is $\frac{dA^{-1}}{dt}$?

Notice that $B^{-1} - A^{-1} = B^{-1}(A - B)A^{-1}$

A−B and $B^{-1}-A^{-1}$ have the same rank. multiplying by invertible matrix doesn't change the rank.

$\Rightarrow \Delta A^{-1} = (A + \Delta A)^{-1}(-\Delta A)A^{-1}$

divide by Δt and let $\Delta t \to 0$. then

$$\frac{dA^{-1}}{dt} = -A^{-1}\frac{dA}{dt}A^{-1}$$

Compare with 1×1 case. $A(t) = t$.

* Changes in Eigenvalues

Let $A(t)x(t) = \lambda(t)\cdot x(t)$ (eigenvector of A)

$y^T(t)A(t) = \lambda(t)y^T(t)$ (eigenvector of A^T)

Norm given by $y^T x = 1$.

$\Rightarrow AX = X\Lambda$, $Y^TA = \Lambda Y^T$, $Y^TX = I$

Also, $y^TAx = y^T(\lambda x) = \lambda$. Take the derivative!

$\frac{d\lambda}{dt} = \frac{dy^T}{dt}Ax + y^T\frac{dA}{dt}x + y^TA\frac{dx}{dt}$ (Product rule)

$= \frac{dy^T}{dt}\lambda x + y^T\frac{dA}{dt}x + \lambda y^T\frac{dx}{dt}$

$= \lambda\left(\frac{dy^T}{dt}x + y^T\frac{dx}{dt}\right) + y^T\frac{dA}{dt}x$

$= \lambda\frac{d}{dt}(y^Tx) + y^T\frac{dA}{dt}x = y^T\frac{dA}{dt}x$ ($y^Tx = 1$, $\frac{d}{dt}(y^Tx) = 0$)

→ Does not involve $\frac{dy^T}{dt}$, $\frac{dx}{dt}$!

For symmetric matrix S. Note that uu^T is positive semi-definite.

Eigenvalues of $S + uu^T$?

Let λ_i, μ_i be eigenvalues of S, $S+uu^T$.

Then the eigenvalues interlace!

$\mu_1 \geq \lambda_1 \geq \mu_2 \geq \lambda_2 \geq \cdots \geq \cdots$

Repeat the process and make rank 1 changes.

let v_i be the eigenvalues of $S + uu^T + ww^T$. then

$v_1 \geq \mu_1 \geq v_2 \geq \mu_2 \geq \cdots$

thus we know that $\lambda_1 \geq v_3$!!!

Suppose $Su = \lambda_2 u$. (2nd unit eigenvector)

Look at $S + 20uu^T$.

$(S+20uu^T)u = Su + 20uu^Tu = (\lambda_2 + 20)u$

How does it not contradict the interlacing?

16. Derivatives of Inverse and Singular Values

* Changes in Singular Values.

From $AV = U\Sigma$, $\sigma(t) = u^T(t)A(t)v(t)$.

$\Rightarrow \frac{d\sigma}{dt} = \frac{du^T}{dt}Av + u^T\frac{dA}{dt}v + u^TA\frac{dv}{dt}$

$= \frac{du^T}{dt}\sigma u + u^T\frac{dA}{dt}v + \sigma v^T\frac{dv}{dt} = u^T\frac{dA}{dt}v$

($\because u^Tu = 1 \Rightarrow \frac{du^T}{dt}u + u^T\frac{du}{dt} = 0 \Rightarrow \frac{du^T}{dt}u = 0$)

We can't get exact formulas but we can get bounds.

* Interlacing.

Eigenvalues of S : $\lambda_1 \geq \lambda_2 \geq \cdots$

Eigenvalues of $S + kuu^T$: $\mu_1 \geq \mu_2 \geq \cdots$

$\mu_1 \geq \lambda_1 \geq \mu_2 \geq \lambda_2 \geq \cdots$

If $\mu_2 = \lambda_2 + k$, take k as large as μ_2 passes λ_1.

⇒ As k increases, μ_2 is now μ_1 and the inequality still holds.

* Weyl's Inequality

For symmetric S.T., $\lambda_{i+j-1}(S+T) \leq \lambda_i(S) + \lambda_j(T)$ (i-th eigenvalue of S)

Compressed Sensing

Nuclear norm $\|A\|_N = \sigma_1 + \sigma_2 + \cdots + \sigma_r$

Minimizing in L_1 norm makes the answer sparse.

17. Rapidly Decreasing Singular Values

What makes low rank matrices appear in problems?

Let $X \in M_{m,n}(\mathbb{R})$, singular values $\sigma_1, \cdots, \sigma_m$.

Fact : If k is the largest integer s.t $\sigma_{k+1} \leq 0 < \sigma_k$,

① $\operatorname{rk} X = k$

② $X = \sigma_1u_1v_1^T + \cdots + \sigma_ku_kv_k^T$

③ Rank Theorem.

Sending images. $2kn < n^2$ (SVD U, V, Σ; All entries)

Def X is a low rank matrix if $\operatorname{rk} X < \frac{n}{2}$.

often, we demand $k \ll \frac{n}{2}$

Rank 1 matrix → Highly aligned.

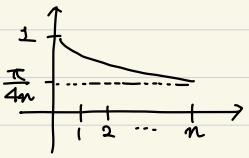
Triangular Flag  $= \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 1 & & 1 \end{bmatrix} = X$

$X^{-1} = \begin{bmatrix} 1 & & \\ -1 & \ddots & \\ & \ddots & 1 \end{bmatrix}$ $(X^TX)^{-1} = \begin{bmatrix} 1 & -1 & \\ -1 & 2 & \ddots \\ & \ddots & \frac{1}{2} \end{bmatrix}$

⇒ Singular values $\sigma_k = \left[2\sin\frac{\pi(2k-1)}{2(2n+1)}\right]^{-1}$

$\sigma_1 \approx \frac{2n}{\pi}$, ..., $\sigma_n \approx \frac{1}{2}$

Plot of σ_k/σ_1



Japan Flag

$\operatorname{rk}\left(\boxed{r}\right) \leq \operatorname{rk}\left(\boxed{}\right) + \operatorname{rk}\left(\boxed{}\right)$ (Same column/row space.) Use symmetry to cut down. ≤1

$\leq \operatorname{rk}\left(\boxed{}\right) + \operatorname{rk}\left(\boxed{}\right) + 1$

$\leq 2\left\lceil r\left(1 - \frac{1}{\sqrt{2}}\right)\right\rceil + 1$

Numerical rank for $0 < \varepsilon < 1$ (tolerance) is k if $\operatorname{rk}_\varepsilon X = k$

k is the largest integer s.t $\sigma_{k+1} \leq \varepsilon\sigma_1 < \sigma_k$

By definition, $\operatorname{rk}_0 X = \operatorname{rk} X$

Eckart-Young : σ_{k+1}, $\|X - X_k\|_2$

(X_k : best rank k matrix)

* Matrices of low numerical rank.

- All low rank matrices.
- Hilbert matrix (full rank but low numerical rank)

$H_{ij} = \frac{1}{i+j-1}$

· Vandermonde matrix $V = \begin{bmatrix} 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{bmatrix}$ $(x_i \in \mathbb{R})$

Often we want V^{-1}, but hard to compute

The world is Smooth. (Reade, 1983)

Ex). $p(x,y) := 1 + x + xy$

Define $x_{ij} = p(i,j) = 1 + i + ij$.

$$X = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix} + \begin{bmatrix} 1 \\ \vdots \\ n \end{bmatrix} \begin{bmatrix} \cdots \end{bmatrix} + \begin{bmatrix} 1 & 2 & \cdots & n \\ 2 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ n & \cdots & \cdots & n^2 \end{bmatrix}$$

rank 1 rank 1 rank 1

$\therefore$ rk $X = 3$.

In general. $p(x,y) = \sum_{s,t}^{n-1} a_{st} x^s y^t$

$x_{ij} = p(i,j)$ then rk $X \le n^2$

For Hilbert matrix, $f(x,y) = \frac{1}{x+y-1}$.

Approximate f by a polynomial $\Rightarrow |f(x,y) - p(x,y)| \le \frac{\varepsilon}{n}\|X\|_2$

Then matrix $Y = (y_{ij})$, where $y_{ij} = p(i,j)$

should have finite rank and $\|X - Y\|_2 \le \varepsilon \|X\|_2$

(will have low numerical rank)

Doesn't work well ···

rk $H_{1000} = 1000$, rk$_\varepsilon H_{1000} = 28$ $(\varepsilon = 10^{-15})$

But Reade's argument ... rk$_\varepsilon H_{1000} \le 719$

→ Not strong enough to show that H_{1000} is of low numerical rank.

* The world is Sylvester

Sylvester Equation.

Many matrices satisfy $AX - XB = C$ for some A.B.C

Want to find A.B.C → X is of low numerical rank.

$$\begin{bmatrix} 1/2 & & & \\ & 3/2 & & \\ & & \ddots & \\ & & & n-1/2 \end{bmatrix} H - H \begin{bmatrix} -1/2 & & & \\ & -3/2 & & \\ & & \ddots & \\ & & & -n+\frac{1}{2} \end{bmatrix} = \begin{bmatrix} 1 & & & 1 \\ 1 & & & \\ \vdots & & & \\ 1 & & & 1 \end{bmatrix}$$

$$\begin{bmatrix} x_1 & & \\ & \ddots & \\ & & x_n \end{bmatrix} V - V \begin{bmatrix} 0 & & & -1 \\ 1 & \ddots & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \cdots & * \\ \vdots & & \vdots \\ 0 & \cdots & * \end{bmatrix}$$

C happens to be low rank.

Thm If X satisfies $AX - XB = C$ for normal matrices A, B. and let $r = \text{rk}\, C$. then

$$\sigma_{rk+1} \le Z_k(E,F)\,\sigma_1$$

* Z : Zolotarev number, E, F : Eigenvalue set of A. B

Key : E and F are separated.

For H : $E \subseteq [\frac{1}{2}, n-\frac{1}{2}]$. $F \subseteq [-n+\frac{1}{2}, -\frac{1}{2}]$

$Z_k(E,F) \to 0$ as k increases.

→ rk$_\varepsilon H_{1000} \le 34$. (Huge improvement)

18. Counting Parameters in SVD, LU, QR.

Counting free parameters in matrix factorization.

L : $\frac{1}{2}n(n-1)$, U : $\frac{1}{2}n(n+1)$

Q (orthogonal) : $\frac{1}{2}n(n-1)$

Λ (diagonal) : n

X (eigenvectors) : $n^2 - n$ (Normalize them)

S (Symmetric) : $\frac{1}{2}n(n+1)$

Let A be an $n \times m$ matrix with rank r.

$r \times r$ Σ : has r parameters

$n \times r$ U : has r orthonormal n-dimensional vectors.

$(n-1) + (n-2) + \cdots + (n-r) = rn - \frac{1}{2}r(r+1)$

$r \times m$ V^T : has r orthonormal m-dimensional vectors.

$(m-1) + (m-2) + \cdots + (m-r) = rm - \frac{1}{2}r(r+1)$

$\Rightarrow$ Total : $(m+n)r - r^2$ ← $r \times r$ block; * : Automatically Decided.

* Saddle Point from constraints.

Positive definite S.

Minimize $\frac{1}{2}x^T S x$ with respect to $Ax = b$.

Lagrangian. $L(x, \lambda) = \frac{1}{2}x^T S x + \lambda^T (Ax - b)$ (minus is also OK.)

$x \in \mathbb{R}^n$, $\lambda \in \mathbb{R}^m$

$\frac{\partial L}{\partial x} = Sx - A^T\lambda = 0$. $\frac{\partial L}{\partial \lambda} = Ax - b = 0$.

$$\Rightarrow \begin{bmatrix} S & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

Not positive definite.

The solution $\begin{bmatrix} x \\ \lambda \end{bmatrix}$ is a saddle point of $L(x, \lambda)$

· KKT matrix / conditions

Saddle points from Rayleigh quotient. $R(x) = \frac{x^T S x}{x^T x}$

$\lambda_{max} = \max_{x \in V} R(x)$, at eigenvector x_k ($Sx_k = \lambda_{max} x_k$)

$\lambda_{min} = \min_{x \in V} R(x)$. at eigenvector y_k ($Sy_k = \lambda_{min} y_k$)

$\therefore \lambda_{min} \le R(x) \le \lambda_{max}$

At $x = v_k$ (k-th eigenvector), saddle point.

$R(v_k) = \lambda_k$, grad $R(v_k) = 0$

* Min-Max Principle : $\lambda_2 = \max_{V \subseteq \mathbb{R}^n} \min_{x \in V} \frac{x^T S x}{x^T x}$ (subspace)

· Read more about this.

20. Definitions & Inequalities

Expectation $\mu = E[X]$

Variance $\sigma^2 = E[(X-\mu)^2]$

$E[f(X)] = \sum_i p_i \cdot f(a_i)$

$\Rightarrow \sigma^2 = \sum_i p_i (x_i-\mu)^2 = \sum_i p_i x_i^2 - \mu^2 = E[X^2] - E[X]^2$

* Markov Inequality

Holds for nonnegative random variable X.

For $a>0$, $P(X \geq a) \leq \frac{E[X]}{a}$

Proof). $E[X] = \int_0^\infty x f(x)\,dx = \int_0^a x f(x)\,dx + \int_a^\infty x f(x)\,dx$

$\geq \int_a^\infty x f(x)\,dx \geq \int_a^\infty a f(x)\,dx = aP(X \geq a)$

* Chebyshev Inequality

For random variable X, $P(|X-\mu| \geq a) \leq \frac{Var(X)}{a^2}$

Proof). From Markov,

$P(|X-\mu| \geq a) = P((X-\mu)^2 \geq a^2) \leq \frac{E[(X-\mu)^2]}{a^2} = \frac{Var(X)}{a^2}$

* Covariance Matrix.

m experiments at once. Random variables X_i $(i=1, \cdots, m)$

$V = (v_{ij})_{m \times m}$ where $v_{ij} = Cov(X_i, X_j)$

And $Cov(X_i, X_j) = E[(X_i-\mu_i)(X_j-\mu_j)]$

Alternatively, let $X = (X_1, X_2, \cdots, X_m)$.

Then $\overline{X} = (\overline{X_1}, \overline{X_2}, \cdots, \overline{X_m})$ and

$V = E[(X-\overline{X})(X-\overline{X})^T]$ (always P.S.D.)

$= E[XX^T] - \overline{X}\,\overline{X}^T$

21. Minimizing a Function Step by Step.

Taylor Expansion. (Single variable)

$f(x+\Delta x) \approx f(x) + f'(x)\Delta x + \frac{1}{2} f''(x) \Delta x^2$

Taylor Expansion (Multi-variable)

$F(x+\Delta x) = F(x) + (\Delta x)^T \nabla F(x) + \frac{1}{2}(\Delta x)^T H \Delta x$

where H is the Hessian matrix.

Hessian matrix (i,j)-component : $\frac{\partial^2 F}{\partial x_i \partial x_j}$

and also H is symmetric.

Let $f : \mathbb{R}^n \to \mathbb{R}^n$, defined by

$f(x_1, \cdots, x_n) = f(\vec{x}) = (f_1(\vec{x}), \cdots, f_n(\vec{x}))$

where $f_i : \mathbb{R}^n \to \mathbb{R}$.

$f(x+\Delta x) = f(x) + J\Delta x$

where J is the Jacobian, which has $\frac{\partial f_i}{\partial x_j}$ as its (i,j)-component.

How would we solve $f = \vec{0}$ $\begin{cases} n \text{ equations} \\ n \text{ unknowns.} \end{cases}$

* Newton's Method.

$x_{i+1} = x_i - J(x_i)^{-1} f(x_i)$ for $i = 0, 1, \cdots$

Can be used to solve $f(\vec{x}) = 0$.

* Minimizing Functions.

Minimizing $F(x)=0$ is similar (!) to solving $\nabla F = 0$.

① Steepest Descent

$x_{i+1} = x_i - s_i \nabla F$ (linear convergence)

(Gradient is the steepest direction)

② Newton's Method. (Solve $\nabla F = 0$)

$x_{i+1} = x_i - H^{-1} \nabla F$ (Quadratic convergence)

* Convexity (for convex optimization)

A set K is a convex set iff for any $x, y \in K$ and any $t \in [0,1]$, $(1-t)x + ty \in K$.

For two convex sets A, B, $A \cup B$ is not convex, but $A \cap B$ is always convex.

Convex sets are usually the domain where we minimize the objective function.

A function f is convex iff for any $x, y \in dom(f)$ and any $t \in [0,1]$, $f((1-t)x+ty) \leq (1-t)f(x) + tf(y)$

For two convex functions f, g, $\min(f,g)$ is not convex. but $\max(f,g)$ is always convex.

Using derivatives,

$f : \mathbb{R} \to \mathbb{R}$, $f''(x) \geq 0$ then convex.

$f : \mathbb{R}^n \to \mathbb{R}$, $H \geq 0$ then convex. (P.S.D.)

22. Gradient Descent : Downhill to Minimum

Gradient Descent.

$x_{i+1} = x_i - \alpha \nabla f(x_i)$

α: learning rate. $\begin{cases} \text{If } \alpha \text{ is too big} \to \text{oscillation.} \\ \text{If } \alpha \text{ is too small} \to \text{long training.} \end{cases}$

Must choose the optimal α (step size) : exact line search.

Backtracking line search : use $\alpha, \frac{1}{2}\alpha, \frac{1}{32}\alpha, \cdots$

23. Accelerating Gradient Descent

With momentum, $\frac{1-b}{1+b} \to \frac{1-\sqrt{b}}{1+\sqrt{b}}$ (rate of decay)

Momentum

$$\begin{cases} x_{i+1} = x_i - \alpha z_i \\ z_i = \nabla f(x_i) + \beta z_{i-1} \end{cases} \quad (\approx \text{memory term})$$

Similar to second order differential equations

→ damping factors appear.

Try writing it with two first-order equations.

We try to minimize $f(x) = \frac{1}{2}x^T S x$. $\rightarrow \nabla f = Sx$

$$\begin{cases} x_{k+1} = x_k - \alpha z_k \\ z_{k+1} - S x_{k+1} = \beta z_k \end{cases}$$

↦ The problem becomes scalar

Let $Sq = \lambda q$ (eigen) we track eigenvectors.

$x_k = c_k q$, $z_k = d_k q$. Then $S x_{k+1} = \lambda c_{k+1} q$

$$\Rightarrow \begin{bmatrix} 1 & 0 \\ -\lambda & 1 \end{bmatrix} \begin{bmatrix} c_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -\alpha \\ 0 & \beta \end{bmatrix} \begin{bmatrix} c_k \\ d_k \end{bmatrix}$$

$$\therefore \begin{bmatrix} c_{k+1} \\ d_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \lambda & 1 \end{bmatrix} \begin{bmatrix} 1 & -\alpha \\ 0 & \beta \end{bmatrix} \begin{bmatrix} c_k \\ d_k \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} 1 & -\alpha \\ \lambda & \beta - \lambda\alpha \end{bmatrix}}_{R} \begin{bmatrix} c_k \\ d_k \end{bmatrix}$$

Now we want to choose R to be as small as possible.

Suppose we know that $0 < m \le \lambda \le M$ (eigenvalues of S)

The condition number $K = \frac{M}{m} = \frac{\lambda_{max}}{\lambda_{min}}$,

if the condition number is large, the problem is harder

Suppose R has eigenvalues e_1, e_2. they depend on α and β. So we want to minimize the largest e_i, for all possible λ of S. $(0 < m \le \lambda \le M)$

$$\alpha_{optimal} = \left(\frac{2}{\sqrt{M} + \sqrt{m}}\right)^2 \qquad \beta_{optimal} = \left(\frac{\sqrt{M} - \sqrt{m}}{\sqrt{M} + \sqrt{m}}\right)^2$$

So what ... ?

Nesterov Accelerated Gradient (NAG)

$$x_{i+1} = x_i + \beta(x_i - x_{i-1}) - \alpha \nabla f(x_i + \gamma(x_i - x_{i-1}))$$

Evaluate it somewhere on a line

24. Linear Programming and Two-Person Games

Linear Programming

Want to minimize $c^T x = c_1 x_1 + \cdots + c_n x_n$.

Constraint on x. $Ax = b$, $A \in M_{m \times n}(\mathbb{R})$, $m < n$

and all $x_i \ge 0$. "feasible set"

① Simplex Method. (Dantzig)

Start from 1 corner → next corner → ... that lowers the cost

② Karmarkar's Method. "interior"

Dual LP : maximum of $b^T y = b_1 y_1 + \cdots + b_m y_m$

Constraint : $A^T y \le c$

Weak duality for any feasible x, y, $b^T y \le c^T x$.

Proof. $b^T y = x^T A^T y \le x^T c = c^T x$

($x \ge 0$ keeps the direction of the inequality)

Strong duality $b^T y^* = c^T x^*$ $\begin{cases} y^* \text{ maximizes } b^T y \\ x^* \text{ minimizes } c^T x \end{cases}$

Max flow – min cut

For a graph with each edge having a capacity, what is the maximum flow from the source to sink?

Maximum flow is equal to the min cut.

A cut partitions the vertices into two sets. A and B. the value of a cut is computed by the sum of all edges $e = (s.t)$ such that $s \in A$ and $t \in B$.

(its ends are in different sets)

26. Structure of Neural Nets for Deep Learning

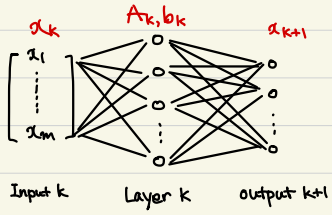
We have training data = feature vectors and the answer

$F: x = (x_1, \cdots, x_m) \rightarrow$ correct answer

We want to create a function $F(x)$ that is correct on almost all of the data (don't want overfitting)

Also we have activation functions (usually nonlinear).

$\text{ReLU}(x) = \max(0, x)$



In each layer,

$y_k = A_k x_k + b_k$

$x_{k+1} = f(y_k)$ (activation)

$r(N, m)$: Number of flat pieces in m dimensions with N folds

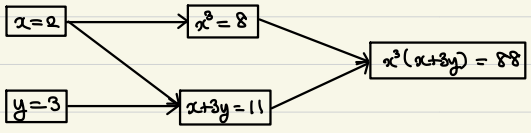
$$r(N, m) = \sum_{r=0}^{m} \binom{N}{r}$$

$r(N, m) = r(N-1, m) + r(N-1, m-1)$

27. Backpropagation

$F(x, y) = x^3(x+3y)$

Computational Graph of F.



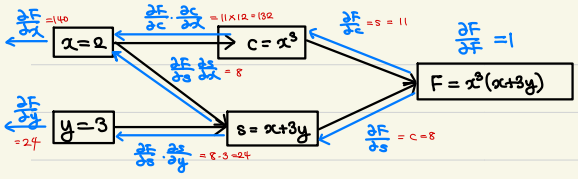
Forward (not recommended)

x → $\frac{\partial x^3}{\partial x} = 3x^2$, $\frac{\partial x^3}{\partial y} = 0$ ...?

$\frac{\partial}{\partial x}(x^3(x+3y)) = 3x^2(x+3y) + x^3 = 4x^3 + 3x^2 y$ $= 140$

y → $\frac{\partial (x+3y)}{\partial x} = 1$, $\frac{\partial (x+3y)}{\partial y} = 3$

$\frac{\partial}{\partial y}(x^3(x+3y)) = 0 + 3x^3 = 3x^3$ $= 24$



30. Completing a Rank 1 Matrix, Circulants

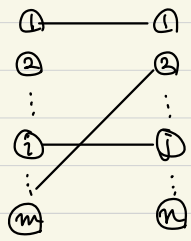
$A = uv^T$ ($m \times n$ matrix). $a_{ij} = u_i v_j$

We are given $m+n-1$ non-zero numbers (with position)

Can we complete the rank 1 matrix A?

For which positions, is the problem solvable?

If element (i,j) is given, add an edge



→ bipartite graph.

If such bipartite graph has a cycle, the problem will be unsolvable.

Circulants (cyclic convolution matrix)

diagonals circle around. and each diagonal is constant.

$$C = \begin{bmatrix} 2 & 5 & 1 & 0 \\ 0 & 2 & 5 & 1 \\ 1 & 0 & 2 & 5 \\ 5 & 1 & 0 & 2 \end{bmatrix}$$

For two circulant matrices C, D.

① CD is also circulant.

② $C = c_0 I + c_1 P + c_2 P^2 + \cdots$ (polynomial in P)

where P is a permutation matrix of $\sigma = (1\ 2\ 3\ \cdots\ n)$

Note that $P^n = I$.

For $v = (v_1, \cdots, v_n)$, $w = (w_1, \cdots, w_n)$, convolution $v * w$

$$v * w = (c_1, c_2, \cdots, c_{2n-1}) \text{ where}$$

$$c_k = \sum_{i=1}^{k} v_i \cdot w_{k+1-i} \quad (k = 1, \cdots, 2n-1)$$

A cyclic convolution $v \circledast w$ gives

$$v \circledast w = (d_1, d_2, \cdots, d_n) \text{ where}$$

$$d_k = c_k + c_{k+n} \quad (c_{2n} = 0 \text{ for convenience})$$

Eigenvalues, Eigenvectors of Circulants

Eigenvectors of the permutation matrix P are also eigenvectors of a circulant.

If $C = f(P)$, and λ is an eigenvalue of P, $f(\lambda)$ is an eigenvalue of C.

31. Eigenvectors of Circulants. Fourier Matrix

But note that the eigenvalues of P are roots of unity!

Because $P^n = I$, characteristic polynomial is $t^n - 1$.

∴ eigenvalues are ω^i for $i = 0, 1, \cdots, n-1$, $\omega = e^{\frac{2\pi}{n}}$

Fourier matrix is the eigenvector matrix of P.

$$F_n = \frac{1}{\sqrt{n}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \omega & \cdots & \omega^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & & \omega^{(n-1)^2} \end{bmatrix}$$

(i,j)-component : $(\omega^{i-1})^{j-1}$

32. Convolution Rule

Convolution of functions $f(t)$, $g(t)$. $h(t) = (f * g)(t)$

$$h(t) = \int_0^t f(u)\, g(t-u)\, du$$

Convolution Rule

Eigenvalues of P are $\omega^0, \omega^1, \cdots, \omega^{n-1}$ $\left(\omega = e^{\frac{2\pi}{n}}\right)$

thus eigenvalues of $C = c_0 I + c_1 P + \cdots + c_{n-1} P^{n-1}$

should be $f(\omega^0), \cdots, f(\omega^{n-1})$ where

$$f(t) = c_0 + c_1 t + \cdots + c_{n-1} t^{n-1}.$$

Since the i-th row of F is

$$[F]_i = (1, \omega^{i-1}, \omega^{2(i-1)}, \cdots, \omega^{(n-1)(i-1)}),$$

$$[F]_i \cdot (c_0, c_1, \cdots, c_{n-1})^T = c_0 + c_1 \omega^{i-1} + \cdots + c_{n-1} \omega^{(n-1)(i-1)}$$

$$= f(\omega^{i-1})$$

Thus $F \cdot (c_0, \cdots, c_{n-1})^T$ gives the eigenvalues of C

Eigenvalues of CD = (Eigenvalues of C) · (Eigenvalues of D)

$$F(c \circledast d) = F(c) \circ F(d) \quad (\text{pointwise multiplication})$$

→ By FFT, this computation is fast.

* Kronecker product : 2D matrix convolution.

33. Neural Nets and the Learning Function

Learning function $F(x, v_0)$

- x : weights in the neural network. A_k, b_k,
- v_0 : sample feature vectors.
- Optimize! using SGD etc

Each layer : $v_{k+1} = \text{ReLU}(A_k v_k + b_k)$ $k = 0, 1, \cdots, L$ (layers)

(No activation function for the last layer)

Often, weights are underdetermined because $\#x \gg \#v$

Loss Function.

Choose x to minimize L (loss function)

$$L = \frac{1}{N}\left[\sum_{i=1}^{N} F(x, a_i) - \text{true}_i\right]$$

$F(x, a_i)$: Output of a_i — true_i: True answer of a_i

Popular Loss Functions

① Square loss (error is squared)

② L^1 loss (LASSO)

③ Hinge loss (-1, 1 classification, regression)

④ Cross-Entropy loss

In SGD, we only use part of the data to calculate the loss and perform backpropagation

Distance Matrices

Given distances squared $\|x_i - x_j\|^2 = d_{ij}$ as matrix D

find positions $x_i \in \mathbb{R}^n$

$$d_{ij} = \|x_i - x_j\|^2 = \underbrace{\|x_i\|^2}_{\text{rank 1 matrix, cols repeated}} - 2x_i^T x_j + \underbrace{\|x_j\|^2}_{\text{rank 1 matrix, rows repeated}}$$

Define $G = X^T X$ and find X from G. (of course, X is not unique)

Also, let $d = (d_1, \cdots, d_n)^T$ and $d_i = \|x_i\|^2$ ← How do we know this?

$D = (d_{ij})$ and $d_{ij} = x_i^T x_j$

From $D = \mathbf{1}d^T + d\mathbf{1}^T - 2X^T X$, we see that

$$X^T X = G = -\frac{1}{2}D + \frac{1}{2}\mathbf{1}d^T + \frac{1}{2}d\mathbf{1}^T$$

Now, how to find X from $X^T X$?

$X^T X$ is positive semi-definite. !

① Eigenvalues of $X^T X = Q \Lambda Q^T$ (by Spectral Thm)

then $X = Q\sqrt{\Lambda}\, Q^T$.

② Elimination on $X^T X \rightarrow X^T X = LDU$

but since $X^T X$ is symmetric, $L^T = U$.

Then $X = \sqrt{D}\, L^T$ (Cholesky) (Much faster)

34. Procrustes Problem

Given two sets of vectors $X = [x_1 \cdots x_m]$ $Y = [y_1 \cdots y_n]$

Find $\underset{Q}{\operatorname{argmin}} \|YQ - X\|_F^2$ subject to $Q^T Q = I$

Closest orthogonal transformation.

Note that $\|A\|_F^2 = \operatorname{tr}(A^T A) = \operatorname{tr}(AA^T) \overset{\text{Trace is sum of eigenvalues}}{=} \sum \sigma_i^2$

$Y^T X = U\Sigma V^T$ then, best $Q = UV^T$

Proof. $\underset{Q}{\operatorname{argmin}} \|YQ - X\|_F^2$

$= \underset{Q}{\operatorname{argmin}} \operatorname{tr}((YQ - X)^T (YQ - X))$

$= \underset{Q}{\operatorname{argmin}} \operatorname{tr}(Q^T Y^T YQ - X^T YQ - Q^T Y^T X + X^T X)$

$= \underset{Q}{\operatorname{argmin}} \left(\|QY\|_F^2 + \|X\|_F^2 - 2\operatorname{tr}(Q^T Y^T X)\right)$

$= \underset{Q}{\operatorname{argmin}} \left(\underbrace{\|X\|_F^2}_{\text{const.}} + \|Y\|_F^2 - 2\operatorname{tr}(Q^T Y^T X)\right)$

$= \underset{Q}{\operatorname{argmax}} \operatorname{tr}(Q^T Y^T X)$

$= \underset{Q}{\operatorname{argmax}} \operatorname{tr}(Q^T U \Sigma V^T) = \underset{Q}{\operatorname{argmax}} \operatorname{tr}(\underbrace{Q^T U V^T}_{\text{orthogonal}} \underbrace{\Sigma}_{\text{diagonal}})$

Since $Q^T UV^T$ is orthogonal, $\operatorname{tr}(Q^T UV^T \Sigma)$ is maximized when $I = Q^T UV^T$ $\therefore Q = UV^T$ ∎

Each column of $Q^T UV^T$ will have norm 1. When multiplied by Σ, the columns are stretched. To maximize the trace, the i-th component of the i-th column should be 1. Thus $Q^T UV^T = I$.

35. Finding Clusters in Graphs

For partition A, B of X, find positions x, y such that

minimizes : $\sum_{a \in A} \|a - x\|^2 + \sum_{b \in B} \|b - y\|^2$

Given $a_1, \cdots, a_m$, the best x that minimizes $\sum_1 \|a_i - x\|^2$

is the centroid of a_i s. $x = \frac{1}{n}\sum a_i$

k-means (here, k = 2)

① Given A, B (clusters), find centroid x, y.

② Given x, y, form best clusters A, B.

- Each node (data) goes with the closer one of x, y

Repeat! (until clusters are same)

Solution Method (spectral clustering)

"Spectrum" of a matrix → Eigenvalues.

Start with graph Laplacian matrix

$$L = A^T A = D - G$$

Laplace's finite difference equation. connects with $u_{xx} + u_{yy} = 0$.

size : $|E| \times |V|$

A: incidence matrix. If $e_i = (u, v)$. $a_{iu} = -1$, $a_{iv} = 1$.

$|V| \times |V|$

D: degree matrix. Diagonal, $d_{ii} = \deg(v_i)$

$|V| \times |V|$

G: adjacency matrix. $g_{ij} = 1$ if $(i, j) \in E$, 0 otherwise.

L is PSD. and $\lambda_1 = 0$ with eigenvector $v_1 = c\mathbf{1}$

Fiedler vector v_2: eigenvector of smallest $\lambda > 0$

The cluster for data i is decided by the sign of the i-th component of v_2.

Also, since L is PSD. $v_1 \perp v_2$, sum of components of v_2 is 0.

36. Alan Edelman and Julia Language.