$U = A$ $\downarrow$ $L = I$ $\downarrow$ $U(2) = L \times U(1)$ $P = I$ $\downarrow$

$Lc = Pb = [\ ]$

$c = [\ ] = Ux$ $x = [\ ]$

useful: many RHS's inexpensive $O(n^2)$ even tho $LU = O(n^3)$. nonsingular

not: sparse bc doubles % per row add. op.

$\%_i \times 2^n = \%_f$

General iterative useful: sparse, large, approx sol ok

Project onto each eqn/line perpendicularly & iteratively get closer to intersection

$$x_{k+1} = x_k + \omega \frac{b_i - a_i \cdot x_k}{|a_i|^2} a_i^T \qquad \measuredangle = \cos^{-1}\left(\frac{a_1 \cdot a_2}{|a_1||a_2|}\right)$$ closer to 90°, converge faster

plot eqns: $\#x_1 + \#y = \#$ @ $x=0$ $y=$ @ $y=0$ $x=$

---

osc if opp signs, diff initial guess, diff ans

$e_1 = A^k \hat{v}_0$

$e_{min} = A^{-k} \hat{v}_0$

preferably sparse if big for efficiency

$(Q - \mu I) e_i = (\lambda_i - \mu I) v_0$ inverse

$U_0 = [v_0 \ w_0 \ \dots]$

$U_{k+1} = A \hat{U}_k$

$\hat{U}_{k+1} = GS(U_{k+1})$

CAN converge but test $p = p+1$

Hotelling if normal: $B = A - \lambda_1 e_1 e_1^T$

Wielandt: $B = A - \frac{e_1 a_p}{e_{1p}}$ $e_2 = e_B + \frac{a_p e_B}{\lambda_2 - \lambda_1} \frac{e_1}{e_{1p}}$

p = largest elem in e_1

$$\hat{\hat{Q}} = \alpha \underbrace{\left(Q + \frac{1}{n} \overset{\text{empty}}{\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}} \begin{bmatrix} \vdots \end{bmatrix}^T\right)}_{\hat{Q}} + (1-\alpha)\frac{1}{n}\begin{bmatrix} 1 \\ \vdots \end{bmatrix}\begin{bmatrix} 1 \\ \vdots \end{bmatrix}^T$$

$\boxed{e_1 \text{ of } \hat{\hat{Q}}^T}$

Perr. Frob. $\mathbb{R}^{\oplus}$, sq $\to$ 1 largest λ_1

$\uparrow \alpha \to$ smaller λ_2, converge faster, but can affect ranking

---

$e^{At} = S \begin{bmatrix} e^{\lambda_1 t} & & \\ & e^{\lambda_2 t} & \\ & & \ddots \end{bmatrix} S^{-1}$ $\frac{dx}{dt} = Ax$ $x(t) = \Sigma c_i e_i e^{\lambda_i t}$

$x(t) = e^{At} x_0$ $x_0 = \Sigma c_i e_i (1)$

$e^{A(t)} = I + A + \frac{A^2}{2!} + \dots$ $x_0 = (e^{At})^{-1} x(t)$ $C = S^{-1} x_0$

$x(t) = (S^{-1} x_0) \begin{bmatrix} e^{\lambda t} & \\ & \ddots \end{bmatrix} S$

let $x = e_i e^{\lambda_i t}$

$\frac{dx}{dt} = \frac{d}{dt} e_i e^{\lambda_i t} = \lambda_i e_i e^{\lambda_i t} = A e_i e^{\lambda_i t}$

phase space: check $\frac{dx}{dt}$ @ $(\pm 1, \pm 1)$

$\lambda_I \neq 0$, osc $\quad \lambda_R > 0, e^{\lambda t} \to \infty$

$\lambda_I = f \quad \lambda_R < 0, e^{\lambda t} \to 0$

λ_f = growth speed

+sol traj.

not normal (e_i not $\perp$)? non-normality $= \frac{|A^T A - AA^T|}{|A|^2}$

$B = e^{At}$ $B^T B x_0 = \lambda_B x_0$ $\lambda_2 << \lambda_1 < 0$

Amp factor $= \frac{|x(t)|^2}{|x_0|^2} = \max \lambda_B$

Jordan: $n \times n$ mat don't have n lin. ind. e_i's, no S^{-1} for $S^{-1}AS = \begin{bmatrix} \lambda_1 & \\ & \lambda_n \ddots \end{bmatrix}$

① λ of A

② $(A - \lambda I)^k e_i = 0$

③ $v_2 = (A - \lambda I) e_1$ $J = M^{-1} A M$

④ $v_3 = (A - \lambda I) e_2$

3×2 more eqns than unkn. = overdet = prob no sol bc contradictory = $\min |Ax - b|^2$ $x = (A^T A)^{-1} A^T b = (A^T A)^+ A^T b = A^+ \cancel{(A^T)^+} \cancel{A^T} b = A^+ b$

2×3 less eqns than unkn. = underdet = ∞ sols = $\min |x|$ $x = A^+ b$

ill und = multiple $\min |Ax - b|^2$ so also $\min |x|$ $x = (A^T A)^+ A^T b = A^+ b$

or QR decomp $A = QR$

+same adv. as LU $A^T A x = A^T b = R^T R = R^T Q^T b$

∇ $Rx = Q^T b$

$A = [a_1 \ a_2]$

$\hat{q}_1 = \frac{a_1}{|a_1|}$

$q_2 = a_1 - (a_2^T \hat{q}_1) \hat{q}_1$

$\hat{q}_2 = \frac{q_2}{|q_2|}$

$A = U \Sigma V^T$

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

smaller dim $A^T A$ or AA^T

λ_i & e_i of smaller

$\sigma_i = \sqrt{\lambda_i} \to$ into Σ

if $A^T A$, $[e_i \dots] = V$

if AA^T, $[e_i \dots] = U$

$U = \frac{A v_i}{\sigma_i}$ $V = \frac{A^T u_i}{\sigma_i}$

remaining cols: random guess & GS to find ($\perp$ to previous cols)

$A = u_1 \sigma_1 v_1^T + \dots u_k \sigma_k v_k^T$ keep only k terms

img compression quality $= \sum_{ij} (A'_{ij} - A_{ij})^2$

for $m \times n$, storage $= \frac{\overset{U}{km} + \overset{V}{kn} + \overset{\Sigma}{k}}{m \times n}$

$\overset{T}{\begin{bmatrix} 1 & 0 & u_1 \\ 0 & 1 & u_2 \\ 0 & 0 & 1 \end{bmatrix}}$ $\overset{R_1}{\begin{bmatrix} UV^T & 0 \\ & 0 \\ 0 \ 0 & 1 \end{bmatrix}}$ $\overset{S_1}{\begin{bmatrix} V \Sigma V^T & 0 \\ & 0 \\ 0 \ 0 & 1 \end{bmatrix}}$ $x = Mx$

$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $RS = A$

C = sensitivity of $\overset{\text{rel error}}{\frac{|\delta x|}{|x|}}$ at max given some $\frac{|\delta b|}{|b|}$

$= \frac{\sigma_1}{\sigma_{min}} = |A||A^{-1}|$ $\frac{|\delta x|}{|x|} \leq C \frac{|\delta b|}{|b|}$

$b_{max\ err.} = v_1$ $\delta b = \frac{b}{|\delta b|/|b|}$

$b_{min\ err} = v_4$

$\frac{\sigma_k}{\sigma_1} > \frac{|\delta b|}{|b|}$ k = eff rank

max stretch $\frac{|Ax|}{|x|} \leq |A| = \sigma_1$ @ v_1 or u_1 $|A^{-1}| = \frac{1}{\sigma_{min}}$

⓪ remove row means from each elem

PCA / multivariate $F = \begin{bmatrix} X \\ Y \end{bmatrix}$

U vecs = PC's (e_i's of C) V vecs = nondim time series coeffs σ_i = how well PC_i explains

$C = \frac{1}{N} FF^T =$ non diag C_{ij} = cov between rows i & j; diag C_{ii} = row i var

$\frac{\lambda_{ci}}{\Sigma \lambda_{cj}} = \frac{\sigma_i}{\Sigma \sigma_j}$ = % var explained

$C e_i = \lambda_{ci} e_i$ trace(C)

$T = U^T F = \Sigma V^T$ = amplitudes of PC's in data @ $t = n$

(proj of f_n onto PC_i) $f_n \cdot u_i = t_{in}$

$F_{reconst} = \underset{(:,k)}{U} \underset{(k,:)}{T} = U \Sigma V^T$ linear comb of PC_i's w weights t_{in}

MCA

U vecs = struct. of 1st dataset (X in $\frac{1}{N} XY^T$) most correlated w the V vec Y struct.

$\frac{\sigma_i}{\Sigma \sigma_i}$ = % of total cov by that U_i vec & V_j vec (X struct. & Y struct.)

→ If cov. PC (for inter-datasets) is smaller/not as strong as intra-dataset var. explained by other PC's. Bc after 1st PC, all rest have to be orthogonal, then if cov. isn't orthogonal, multivariate will miss it. BUT MCA ignores/loses intra-dataset variability to focus on inter-dataset cov.

Total var of dataset: sum of sqs of its elems divided by M = sum of each datavec's var
Represents variability in [amount of students taking courses]. If vanishes, datavecs are constant over time.

Total cov of 2 datasets: sum of sqs of the elems of C
Represents overall covariability of [dataset X] to [dataset Y]. If vanishes, X & Y are ind.

$N =$

$\frac{1}{N} \underset{M \times N}{F} \underset{N \times M}{F^T} = \underset{M \times M}{C}$

$\frac{1}{N}[\quad][\quad] = \frac{1}{N}[\quad] = [\quad]$ a b c d

---

$L_n = \left[\sum_{i=1}^{dim} (x_i - y_i)^n\right]^{\frac{1}{n}}$

$L_2 = \sqrt{\Sigma (x_i - y_i)^2} = |x - y|$

$L_1 = \sum_i^{dim} |x_i - y_i|$

$L_\infty = \max |x_i - y_i|$

$\cos\ dist = \theta = \cos^{-1}\left(\frac{x \cdot y}{|x||y|}\right) = \cos^{-1}\left(\frac{\sum^{dim} x_i y_i}{\sqrt{\Sigma x_i} \sqrt{\Sigma y_i}}\right) = \cos^{-1}\left(\text{corr. of } x \ \& \ y\right)$ var.

Jaccard-sets $= 1 - \frac{|x \cap y|}{|x \cup y|}$

Edit-ordered groups $= \cancel{\varepsilon} \cancel{d} \cancel{x} \cancel{d} = 5$

Hamming-same size ordered groups = # same elems

Curse of dim:

$L_2^2 = (x_i - y_i)^2 \to$ avg dist across all/many dims

$\theta = \cos^{-1}(\text{corr.}) \to \cos^{-1}(0)$ for random long seqs $\to 90°$

problem bc clustering needs diff dists

Multivariate:
"U1 (PC1) indicates as students taking A61, A62, C72 increase, C71 decreases. PC2 shows anticorr between A61 and rest, but sigma small so negligible."
"V1 row, given signs in PC1, above or below avg at time. + in PC1, + in V1 row elem n mean above avg at time n."
"ind cov bc PC1 only 1 dataset and PC2 only other dataset"

MCA:
"U1 shows opening hours of both pools vary together with salary (V1,1) but not covid (V2,1)=0"

λ_i poly: $\lambda^2 - (a+d)\lambda + (ad - bc) = 0$

e_i ref: $[A - \lambda_i I | 0]$ $\text{diag}^{-1} = \begin{bmatrix} \frac{1}{\#} & \frac{1}{\#} & \dots \end{bmatrix}$

$|a|^2 = a^T a = \Sigma a_i^2$ $\det(A) = |A|$

normal: $A^T A = AA^T$, $e_i \perp$

symm: normal, $A = A^T$, e_i are $\mathbb{R}$

orthonormal: symm, $A^T A = AA^T = I$

nonsingular: has A^{-1}, sq & $\det(A) \neq 0$

$\min \ x^T A x$ st $x^T x = 1$ & $x^T \begin{bmatrix} 1 \\ \vdots \end{bmatrix} = 0$

$x^T A x + \lambda(1 - x^T x) = 0$

$Ax = \lambda x$

$\min \lambda$

$\max (Ax)^T (Ax)$ st $x^T x = 1$

$x^T A^T A x + \lambda(1 - x^T x) = 0$

$A^T A x = \lambda x$

$\max \lambda$

$\min (Ax - b)^T (Ax - b)$ st $x^T x = 1$

$x^T A^T A x - b^T A x - x^T A^T b + b^T b + \lambda(1 - x^T x) = 0$

$A^T A x - b^T A = \lambda x$

$x = (A^T A - \lambda I)^{-1} b^T A$

$$x_c = \frac{1}{n}\sum_j^n x_j$$ ↗ strings don't have

$$var_c = \sum_j^n \sum_i^{dim} (x_{ij} - x_{ci})^2$$

radius = $\max |x_j - x_c|$

diameter = $\max |x_j - x_k|$

density = $\frac{n}{radius}$

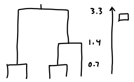
Ward $\Delta var = \frac{N_A N_B}{N_A + N_B} |x_B - x_A|^2$

$= var_{AB} - (var_A + var_B)$

single = $\min |x_A - x_B|$

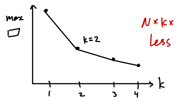
1) plot pts
2) manually pick & calc
3) calc & compare unsure ones

		$\square$ of just merged cluster
1 2 3 4 5		0
(1 2) (3 4) 5		0.7
(1 2) (3 4 5)		1.4
(1 2 3 4 5)		3.3



dist between all pts = $n(n-1) = O(n^2)$
from each new centroid = $n-1, n-2$
& find which pair to merge = $O(n^2 \log n)$

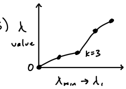
k pts ↗ replicates
1^{st} random
m+1 as far as possible
m+2 smallest dist as far as possible
n-k points
add pt to clustroid
adjust clustroid (or) adjust @ end
new clustroids as k pts
↳ iter. assign til converge



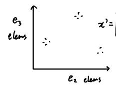
$N \times k \times (rep) \times (iter) \times dim$
less expensive, less powerful

SOM for each x datapt
→ nearest $m_j = m_j + \eta \left(\substack{ker \\ nearest}\right)(x_i - m_j)$
→ others $m_\ell = m_\ell + \eta \left(\substack{ker \\ others}\right)(x_i - m_\ell)$

1) calc $dist_{ij}$
2) $W_{ij} = e^{\left(\frac{-dist_{ij}}{dist_{avg}}\right)}$
3) D = sum over W cols (diag)
4) normalize $L = D - W$
$L' = D^{-1}L = I - D^{-1}W$ → better spectral gap
5) λ



6) k-means in $[e_2\ e_3 \dots e_k]$



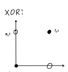
$x' = \begin{bmatrix} e_2^T \\ e_3^T \end{bmatrix}$
- lose orig. dim
+ smaller dim e_i?
+ exclude noise e_i
- calc k λ's tho (Power method) bc L symm

turns out $x^T L x = \frac{1}{2}\sum\sum W_{ij}(x_i - x_j)^2$
so min $x^T L x$ $Lx = \lambda x$
but $x = [\vdots]$ $\lambda = 0$ so
2^{nd} smallest λ

$w' = \begin{bmatrix} w \\ -b \end{bmatrix}$ $x' = \begin{bmatrix} x \\ 1 \end{bmatrix}$

pt 1
$w^T x = 0.6$ $y = 1$ ok
pt 2
$w^T x = 6$ $y = -1$ no
$w = w + \eta(-1)x = [\ ]$
iter. thru pts
plot: $w_1 x_1 + w_2 x_2 + w_3^{(-b)} = 0$
$x_2 = \frac{-w_1}{w_2}x_1 - \frac{w_3}{w_2}$

fails XOR:



$\underset{O \times N}{w'} = \begin{bmatrix} w \\ +b \end{bmatrix}$ $x' = \begin{bmatrix} x \\ 1 \end{bmatrix} = a'$

$z^2 = w^2 a^1$ $a^2 = \sigma(z^2)$
$z^3 = w^3 a^2$ $a^3 = \sigma(z^3)$

① get δ's backprop gives grad

$$\delta^L = \frac{\partial C}{\partial z^L} = \frac{\partial C}{\partial a^L}\cdot\frac{\partial a^L}{\partial z^L}$$

$$\frac{\partial C}{\partial a^L} = \frac{\partial}{\partial a^L}\left[\frac{1}{2}(y - a^L)^2\right] = a^L - y = \nabla_{a^L} C$$

$$\frac{\partial a^L}{\partial z^L} = \frac{\partial}{\partial z^L}(1 + e^{-z^L})^{-1} = e^{-z^L}(1 + e^{-z^L})^{-2} = \sigma'(z^L)$$

$$\delta^L = \nabla_{a^L} C \cdot \sigma'(z^L)$$

$$\delta^\ell = (w^{\ell+1})^T \delta^{\ell+1} \cdot \sigma'(z^\ell)$$

② adj w steepest descent

$\nabla_{w^\ell} C = \delta^\ell (a^{\ell-1})^T$ ⟶ calc avg gradient for minibatch, go through all minibatches
↖ from training x

$w^\ell = w^\ell - \eta \nabla_{w^\ell} C$

③ each time thru all pts = epoch

		$\sigma'(z)$
Relu	$\begin{cases} 0, & z<0 \\ z, & z \geq 0 \end{cases}$	$\begin{cases} 0, & z<0 \\ z, & z \geq 0 \end{cases}$
tanh	$\frac{e^z - e^{-z}}{e^z - e^{-z}}$	$1 - \left(\frac{e^z - e^{-z}}{e^z - e^{-z}}\right)^2$
sigmoid 0.5	$(1 + e^{-z})^{-1}$	$e^{-z}(1 + e^{-z})^{-2}$
softmax pick 1	$\max \frac{e^{z_i}}{\sum_j e^{z_j}}$	

if lots of data:

parameters	training	testing

Overfit? 1) more data 2) smaller network bc small can't perfectly fit training data 3) add dummy data 4) terminate optimization early, 5) add regularization term bc w's and b's go up when overfit

$$C = C_0 + \lambda\left[\sum_{\ell ij}(w_{ij}^\ell)^2 + b_i^2\right]$$

w's & b's ↑ when overfit

perturb each wt

$C_1 = C(w_{ij}^\ell)$
$C_2 = C(w_{ij}^\ell + \delta w_{ij}^\ell)$
⇓
2x feedforward

$\frac{\partial C}{\partial w_{ij}^\ell} \approx \frac{C_2 - C_1}{\delta w_{ij}^\ell}$

BUT need to repeat feedforward for each w_{ij}^ℓ inefficient

$$\frac{d^2x}{dt^2} = a\frac{dx}{dt} + bx$$

define $y = \frac{dx}{dt}$

$\frac{dy}{dt} = ay + bx$

$\frac{dx}{dt} = y$

$\frac{d\begin{bmatrix} x \\ y \end{bmatrix}}{dt} = A\begin{bmatrix} x \\ y \end{bmatrix}$

$A = \begin{bmatrix} 0 & 1 \\ a & b \end{bmatrix}$

$\frac{dx}{dt} = ax + by$ (1)

$\frac{dy}{dt} = cx + dy$ (2)

solve for y (1)

$y = \frac{dx}{dt}\frac{1}{b} - \frac{ax}{b}$ (3)

sub (2)

$\frac{dy}{dt} = cx + \frac{dx}{dt}\frac{d}{b} - \frac{ad}{b}x$

$\frac{d^2x}{dt^2} = a\left(\frac{dx}{dt}\right) + b\left(\frac{dy}{dt}\right)$

$= a^2x + aby + bcx + bdy$

plug in (3) for y

$\frac{dx^2}{dt^2} = (a+d)\frac{dx}{dt} + (bc - ad)x$

Decision trees

Most informative near top branches

Overfit? limit growth, pruning

takes into acc both yes's & no's

$$I_G(\text{interesting}) = \frac{N_{yes}}{N} \sum \text{impurity for yes} + \frac{N_{no}}{N} \sum \text{impurity for no}$$

$$P(\checkmark | yes) = \frac{9\ \checkmark/yes}{12_{yes}}$$

max @ 50%, min @ 0 or 100% "impurity" $= P(1-P)$

$$I_G = \frac{N_{yes}}{N}\left[P(\checkmark|yes)(1-P(\checkmark|yes)) + P(\times|yes)(1-P(\times|yes))\right]$$

$$+ \frac{N_{no}}{N}\left[P(\checkmark|no)(1-P(\checkmark|no)) + P(\times|no)(1-P(\times|no))\right]$$

multilabels/endings

$$I_G(A) = \frac{N_{yes}}{N}\sum_{i=1}^{\ell} P(\ell|yes)(1-P(\ell|yes)) + \frac{N_{no}}{N}\sum_{i=1}^{\ell} P(\ell|no)(1-P(\ell|no))$$

discrete

$N_{yes} = 32$ $\quad P(\checkmark|yes) = \frac{16}{32}$ $\quad P(\times|yes) = \frac{16}{32}$

$N_{no} = 26$ $\quad P(\checkmark|no) = \frac{11}{26}$ $\quad P(\times|no) = \frac{15}{26}$

$N = \quad + \quad =$

$I_G = -(\quad + \quad) + -(\quad + \quad)$ lowest I_G top

________ ignore top attrib's yes rows

________ ignore top attrib's no rows

determine endings based on majority

cont & discrete labels

I_G for each threshold split

I_G's for subsequent subset thresholds

cont & cont labels

min var on each leaf around each leaf's mean

"impurity" ("I_G") $= \frac{N_<}{N} Var_< + \frac{N_>}{N} var_>$

$$Var(y_<) = \frac{1}{N_<}\sum_{i}^{N_<}(y_i - y_{avg<})^2$$

$$\Rightarrow \text{actual split} = \frac{x_{A\,last} + x_{B\,first}}{2}$$

With 2 continuous attrib., calc var.'s for all possible splits based on attrib. 1 and then for all possible splits based on attrib. 2 and then pick lowest split & attrib.