Low-Rank Decompositions of Matrices

Objectives

- Define low-rank decomposition (matrix factorization)

- Explore applications

Matrices represent many types of information 2

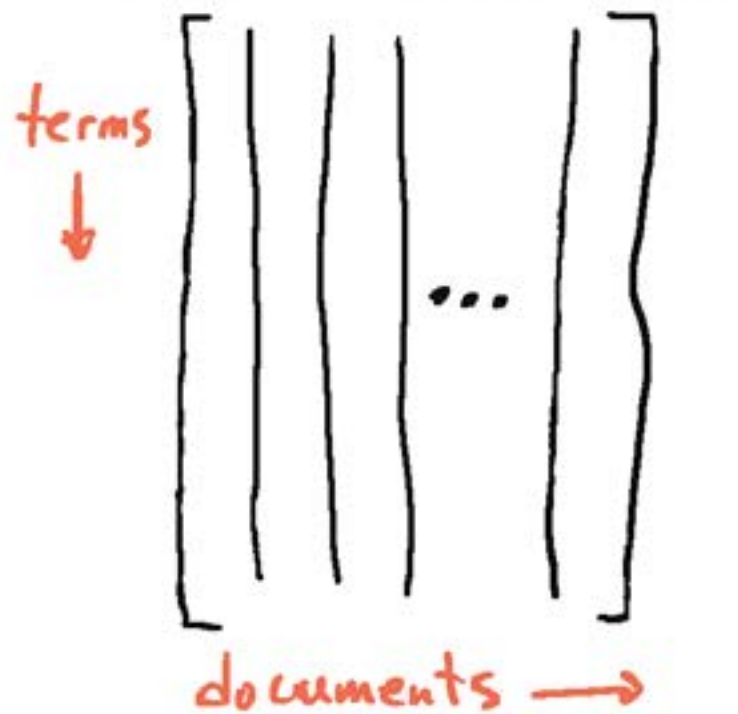
1) Features in classification or modeling
2) User ratings
3) Collections of documents

Bag of words model

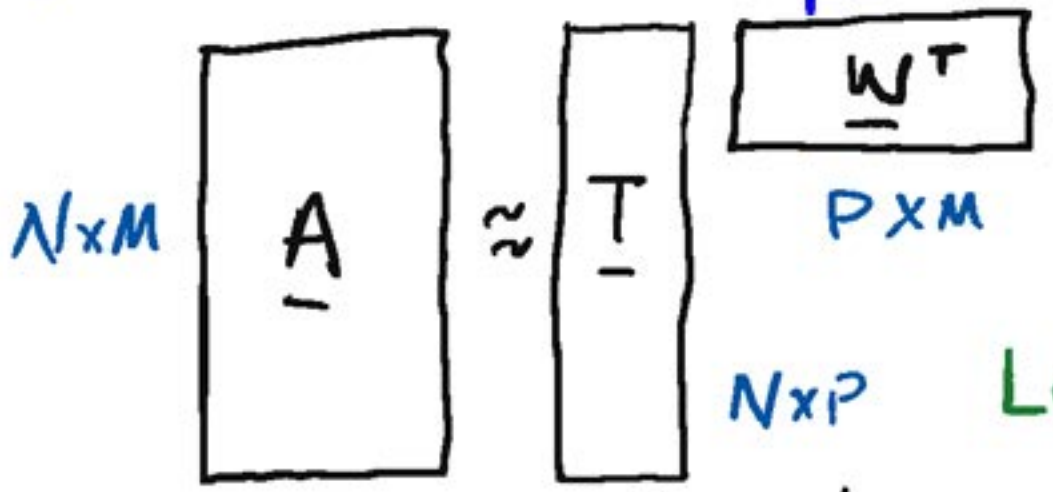
"Apple trees blossom in May."

word	frequency
agency	0
apple	1
blossom	1
car	0
currency	0
happy	0
May	1
politics	0
road	0
tree	1

Term-document matrix



Low-rank decompositions emphasize patterns



$P < N,\ P < M \quad \text{rank}\,\underline{T} = \text{rank}\,\underline{W} = P$

$\Rightarrow \text{rank}(\underline{T}\,\underline{W}^T) = P$

Let $\underline{T} = [\underline{t}_1\ \underline{t}_2 \cdots \underline{t}_P]$, $\underline{W} = [\underline{w}_1 \cdots \underline{w}_P]$

$$\underline{T}\,\underline{W}^T = [\underline{t}_1\ \underline{t}_2 \cdots \underline{t}_P] \begin{bmatrix} -\underline{w}_1^T- \\ -\underline{w}_2^T- \\ \vdots \\ -\underline{w}_P^T- \end{bmatrix} = \sum_{i=1}^{P} \underline{t}_i \underline{w}_i^T = \sum_{i=1}^{P} \Big|_N^M = \sum_{i=1}^{P}$$

column patterns

row patterns

rank-one patterns

Finding patterns —

1) $\min_{\underline{T},\underline{W}} \|\underline{A} - \underline{T}\underline{W}^T\|$ — singular value decomposition

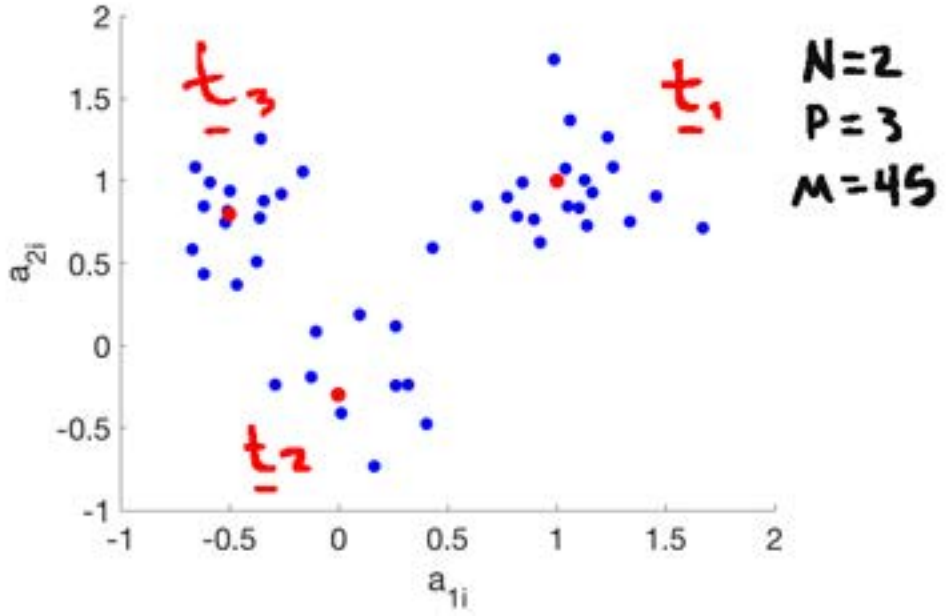
2) $\underline{A} \approx \underline{T}\underline{W}^T,\ \underline{T}, \underline{W} \geq 0$ — nonnegative matrix factorization

3) $\underline{A} \approx \underline{T}\underline{W}^T$ each col $\underline{W}^T$ all 0 w. single 1 — clustering

Clustering groups similar columns

$$\begin{bmatrix} \underline{a}_1 \, \underline{a}_2 \, \underline{a}_3 \, \underline{a}_4 \cdots \underline{a}_m \end{bmatrix} \approx \begin{bmatrix} \underline{t}_1 \, \underline{t}_2 \, \underline{t}_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 0 & & 0 \\ 1 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & 1 & & 0 \end{bmatrix}$$

$$\Rightarrow \underline{a}_1 \approx \underline{t}_2, \; \underline{a}_m \approx \underline{t}_2, \; \underline{a}_2 \approx \underline{t}_1, \; \underline{a}_3 \approx \underline{t}_1, \; \underline{a}_4 \approx \underline{t}_3 \cdots$$



Group similar documents, customers, products, etc

Many algorithms –

k-means

Low rank models "complete" missing data

Jill

Star Trek	8
Pride + Prejudice	3
The Martian	7
Sense + Sensibility	4
Empire Strikes Back	?

Suppose $\underline{a} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} w_1 + \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} w_2$

Use known ratings to solve w_1, w_2

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 8 \\ 3 \\ 7 \\ 4 \end{bmatrix} \Rightarrow \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 5.5 \\ 2 \end{bmatrix}$$

Predict ratings using w_1, w_2

$$\hat{\underline{a}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} 5.5 + \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{bmatrix} 2 = \begin{bmatrix} 7.5 \\ 3.5 \\ 7.5 \\ 3.5 \\ 7.5 \end{bmatrix}$$

Use of "patterns" can mitigate noise

Noisy data $\underline{A}_m = \underline{A}_t + \underline{E}$

$\underline{A}_t$: strong patterns; $\underline{E}$: no dominant patterns

Low-rank model $\hat{\underline{A}}_m = \underline{T}\,\underline{W}^T$ can be closer to $\underline{A}_t$ than $\underline{A}_m$

Low rank classifier / model fit

$\hat{\underline{A}}_m \underline{w} = \underline{d}$

$\underline{T}\,\underline{w}' = \underline{d}$

$\underline{t}$ transformed features $\underline{x}'$

$\underline{x}'^T = \underline{x}^T \underline{W}(\underline{W}^T\underline{W})^{-1}$

New feature $\underline{x}^T$:

1) $\underline{x}'^T = \underline{x}^T \underline{W}(\underline{W}^T\underline{W})^{-1}$ 2) $\hat{d} = \text{sign}(\underline{x}'^T \underline{w}')$

Clustering Data with the K-means Algorithm

Objectives-

- Introduce the K-means algorithm
- Illustrate K-means with an example
- Identify considerations

Clustering: Organizing data in groups

given $\underline{a}_i \in \mathbb{R}^N$, $i = 1, 2, \cdots, M$, find centroids $\underline{\mu}_j$, $j = 1, \cdots K$ and clusters $S_j = \{ i \mid \underline{a}_i \text{ belongs to cluster } j \}$

Example:

$$\underline{A} = [\underline{a}_1\ \underline{a}_2\ \underline{a}_3\ \underline{a}_4\ \underline{a}_5] = \begin{bmatrix} 1 & 3 & 4 & 2 & 5 \\ 2 & 4 & 5 & 1 & 3 \end{bmatrix}$$

$S_1 = \{2, 3, 5\}$, $\underline{\mu}_1 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$; $S_2 = \{1, 4\}$, $\underline{\mu}_2 = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$

Unsupervised learning: data w/o labels, # clusters unknown

Matrix factorization: $\underline{A} \approx \underline{T}\,\underline{W}^T$, $\underline{T} = [\underline{\mu}_1\ \underline{\mu}_2 \cdots \underline{\mu}_K]$

$$\underline{T} = [\underline{\mu}_1\ \underline{\mu}_2], \quad \underline{W}^T = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$[\underline{W}^T]_{\ell m} = \begin{cases} 1, & m \in S_\ell \\ 0, & m \notin S_\ell \end{cases}$$

The K-Means Algorithm (k clusters)

Clusters: $S_j = \{ i \mid \underline{a}_i \in \text{cluster } j \}$, $|S_j| = \#\,\underline{a}_i \text{ in } S_j$

centroids: $\underline{\mu}_j = \frac{1}{|S_j|} \sum_{i \in S_j} \underline{a}_i$ coherence: $c_j = \sum_{i \in S_j} \|\underline{a}_i - \underline{\mu}_j\|_2^2$

Overall coherence $C = \sum_{j=1}^{k} c_j = \sum_{j=1}^{k} \sum_{i \in S_j} \|\underline{a}_i - \underline{\mu}_j\|_2^2 = \|\underline{A} - \underline{T}\,\underline{W}^T\|_F^2$

1) Initialize: choose $\underline{\mu}_j^0$, $j = 1, 2, \ldots, K$ randomly from $\underline{a}_i$; set $\ell = 0$

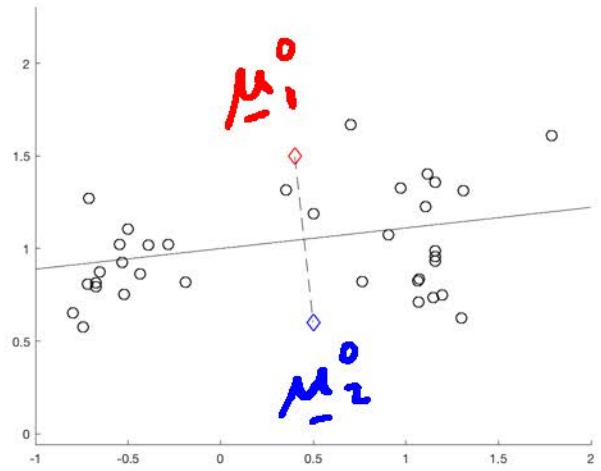
2) Assignment: put $\underline{a}_i \in S_j^\ell$ if $\underline{a}_i$ is closest to $\underline{\mu}_j^\ell$

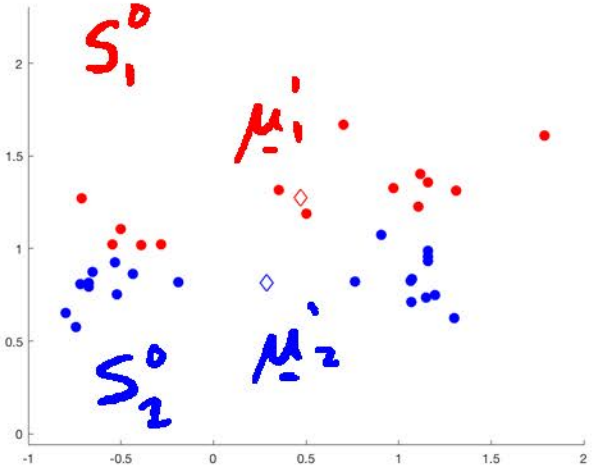
3) Update Centroids: $\underline{\mu}_j^{\ell+1} = \frac{1}{|S_j^\ell|} \sum_{i \in S_j^\ell} \underline{a}_i$

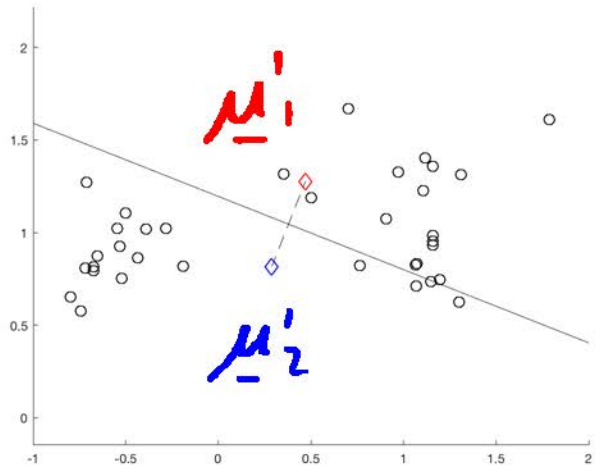
4) If converged $\longrightarrow$ stop

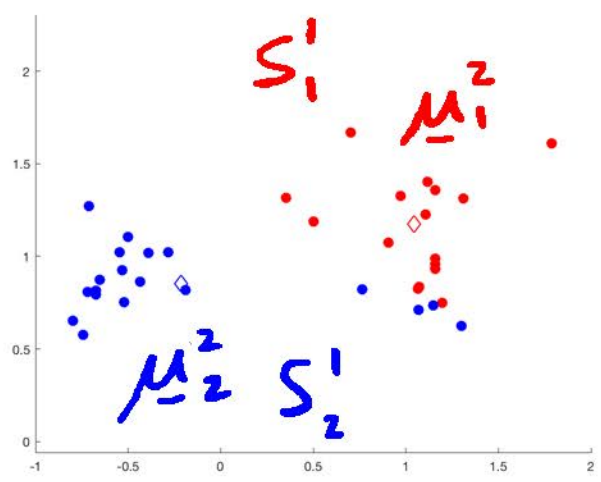
else $\longrightarrow$ $\ell = \ell + 1$, go to 2)

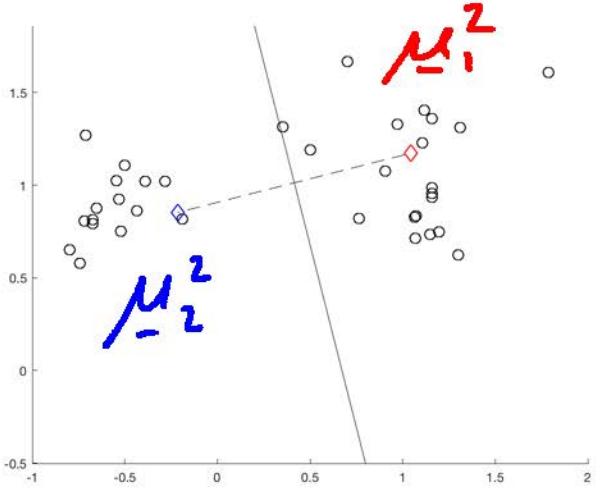
K-Means Algorithm Example

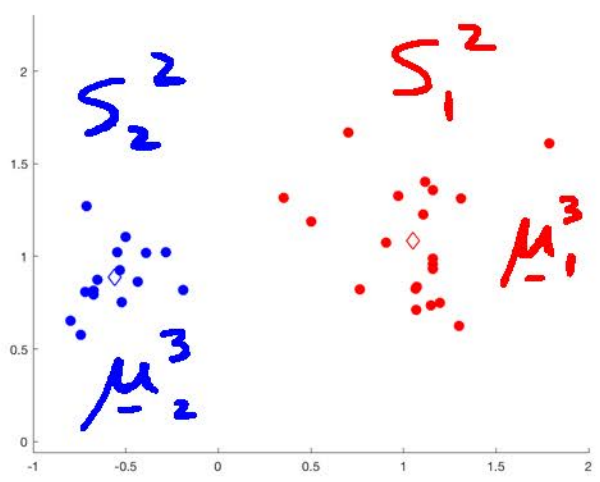












K-Means Algorithm Options

- Initialization: many variations
- Termination: change in clusters or overall coherence or fix iterations
- Use different norms to assign clusters

Challenges

- Convergence to local minima
  repeat for multiple initializations
- Unknown K
  try multiple values

The Singular Value Decomposition (SVD)

Objectives

- Define singular value decomposition (SVD)
- Express skinny SVD
- Write SVD as sum of outer products
- Use SVD to find best low-rank approximation
- Interpret matrix as an operator

SVD

- matrix decomposition that leads to good low-rank approximations
- vast range of applications

Definition:

Any $N \times M$ matrix $\underline{A}$ can be written as

$$\underline{A} = \underline{U}\,\underline{\Sigma}\,\underline{V}^T$$

- $\underline{U}$: $N \times N$, orthonormal columns
- $\underline{V}$: $M \times M$, orthonormal columns
- $\underline{\Sigma}$: $N \times M$, diagonal, $\Sigma_{ii} \geq 0$

$N > M$

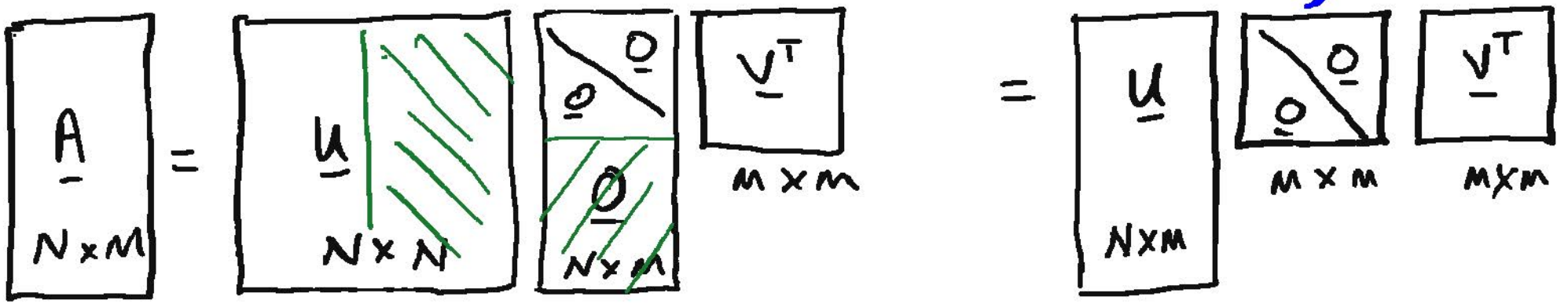
$$\begin{bmatrix} \sigma_1 & & \underline{0} \\ & \sigma_2 & \\ & \ddots & \\ \underline{0} & & \sigma_M \\ & \underline{0} & \end{bmatrix}$$

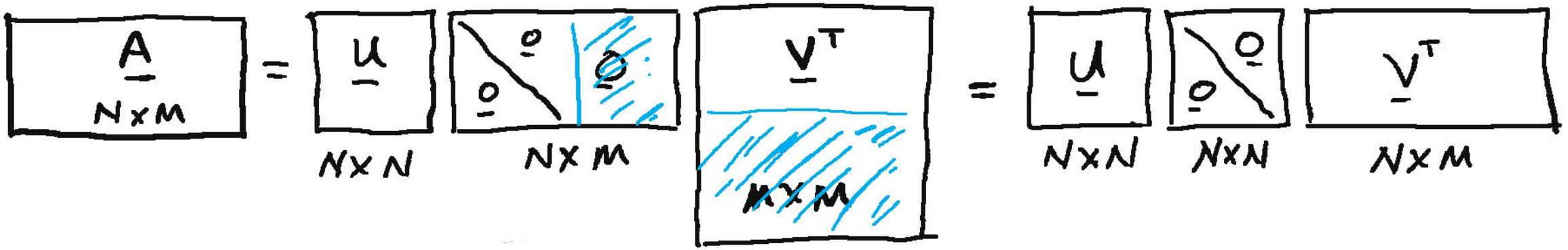
$M > N$

$$\begin{bmatrix} \sigma_1 & & & \underline{0} \\ & \sigma_2 & & \\ & & \ddots & \\ \underline{0} & & & \sigma_N \end{bmatrix}$$

$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{N/M} \geq 0$

SVD Dimensions





Skinny SVD

3

Sum of Outer Products Form:

"rank 1"

$$\underline{A} = \begin{bmatrix} | & | & & | \\ \underline{u}_1 & \underline{u}_2 & \cdots & \underline{u}_M \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \underline{0} \\ & \sigma_2 & & \\ & & \ddots & \\ \underline{0} & & & \sigma_M \end{bmatrix} \begin{bmatrix} - & \underline{v}_1^T & - \\ - & \underline{v}_2^T & - \\ & \vdots & \\ - & \underline{v}_M^T & - \end{bmatrix} = \sum_{i=1}^{M} \sigma_i \underline{u}_i \underline{v}_i^T = \sum_{i=1}^{M} \sigma_i \underline{u}_i \underline{v}_i^T$$

$\sigma_i \underline{u}_i \underline{v}_i^T$ is $N \times M$

SVD gives the "best" low-rank approximation

Frobenius norm $\|\underline{A}\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{M}\left([\underline{A}]_{ij}\right)^2 = \|\mathrm{vec}(\underline{A})\|_2^2$

Eckart-Young Theorem (1936) Let $\mathrm{rank}(\underline{A}) = r$ and $k < r$:

$$\min_{\mathrm{rank}(\underline{B}) \le k} \|\underline{A} - \underline{B}\|_F^2 = \sum_{i=k+1}^{r} \sigma_i^2 \quad \text{for} \quad \underline{B} = \sum_{i=1}^{k} \sigma_i \underline{u}_i \underline{v}_i^T$$

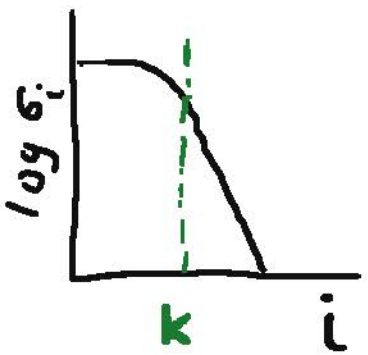
where $\underline{A} = \underline{U}\underline{\Sigma}\underline{V}^T$ is the SVD.

cols: scaled $\underline{u}_i$

rows: scaled $\underline{v}_i^T$

$$\underline{A} \approx \sigma_1 \underline{u}_1\underline{v}_1^T + \sigma_2 \underline{u}_2\underline{v}_2^T + \cdots + \sigma_k \underline{u}_k\underline{v}_k^T$$

patterns: most important, 2nd most, k^{th} most

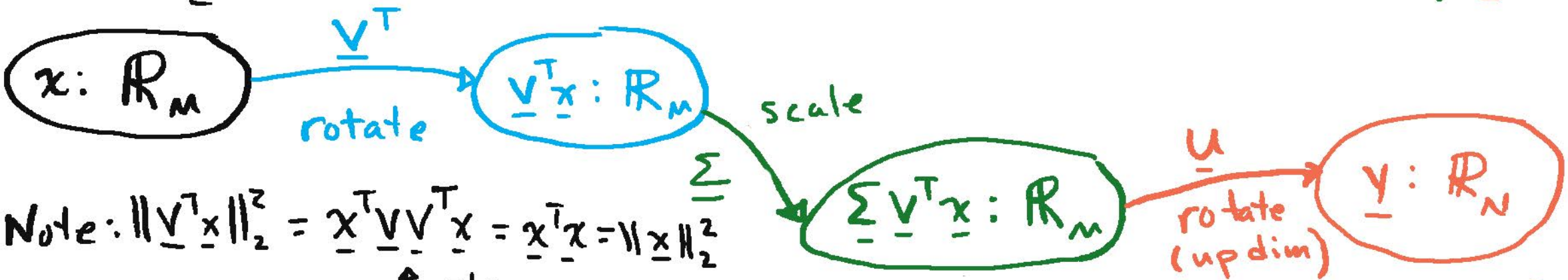


σ_i provide ordered ranking of components

SVD describes matrix as an operator

$A: N \times M$, $x: M \times 1$, $y: N \times 1$

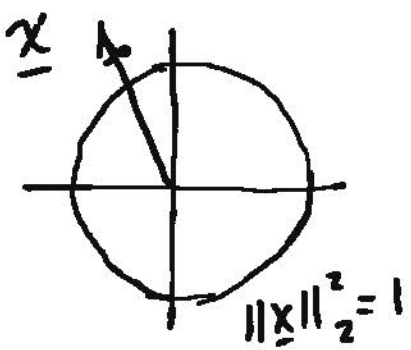
$$y = Ax = U \Sigma V^T x = U\left[\Sigma\left(V^T x\right)\right]$$

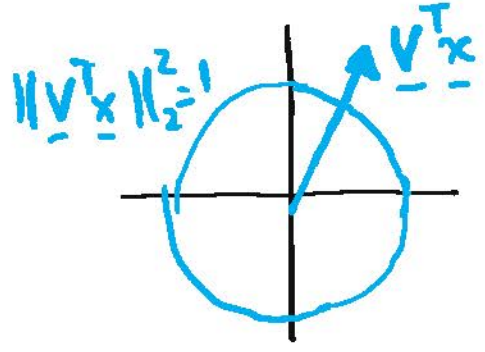


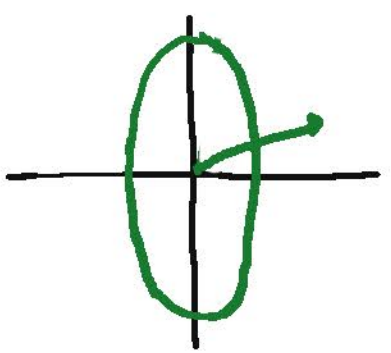
Note: $\|V^T x\|_2^2 = x^T V V^T x = x^T x = \|x\|_2^2$ (V o/n)

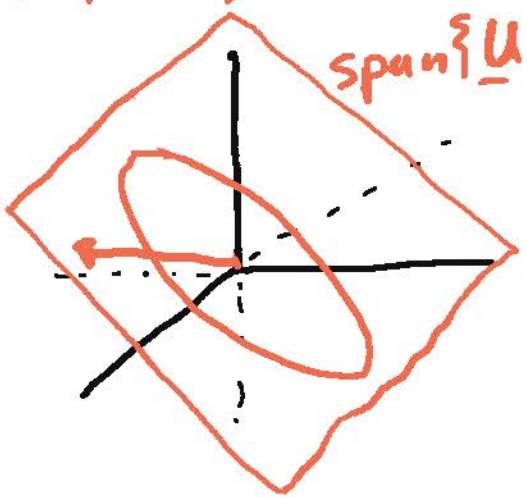
$N = 3$

$M = 2$









Operator Norm

$$\|A\|_2 = \|A\|_{op} := \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sigma_1$$

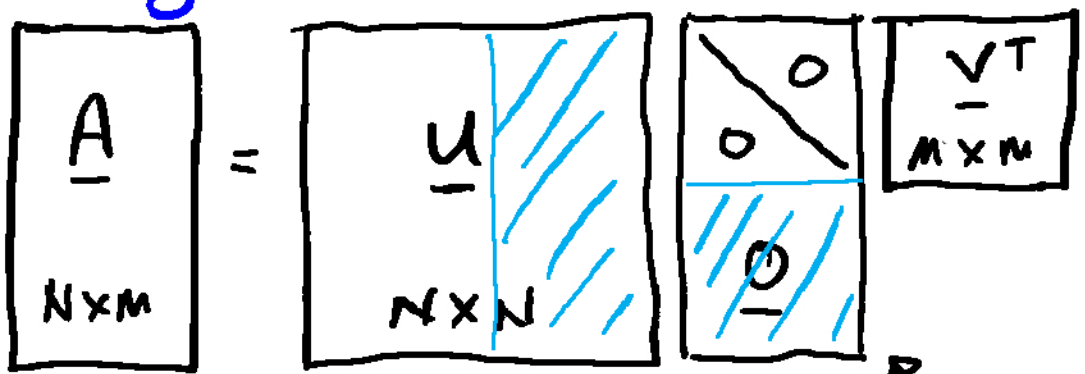
(proof: notes)

Properties of Singular Value Decomposition

Objectives

- review orthormality of singular vectors
- review rank and singular values
- explore singular vectors as bases
- Connect SVD and matrix inversion

Singular Value Decomposition



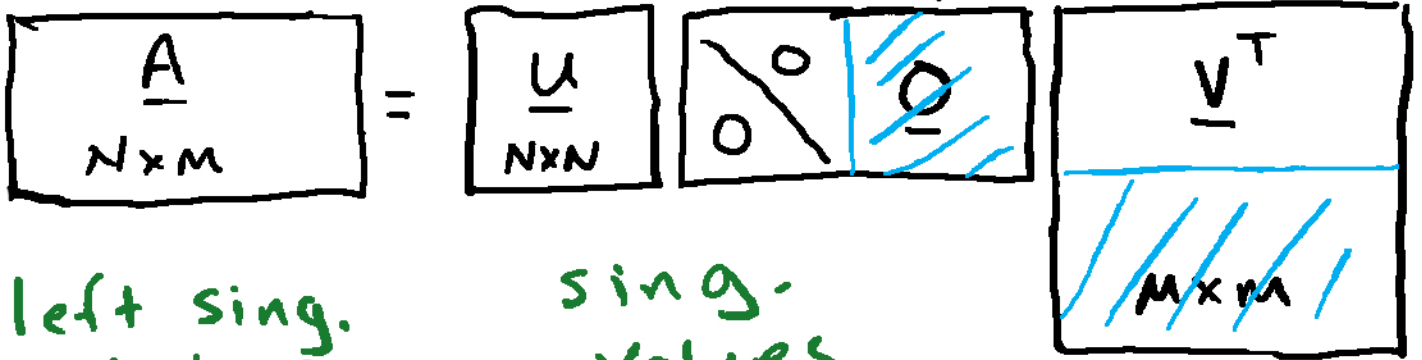
$\Sigma : N \times m$

Orthonormality

$\underline{U}^T\underline{U} = \underline{I}$; $\underline{V}^T\underline{V} = \underline{I}$ full econ

$\underline{U}\,\underline{U}^T = \underline{I}_N$; $\underline{V}\,\underline{V}^T = \underline{I}_m$ full only

Rank



left sing. vectors

$$\underline{U} = \begin{bmatrix} | & | & & | \\ \underline{u}_1 & \underline{u}_2 & \cdots & \underline{u}_N \\ | & | & & | \end{bmatrix}$$

sing. values

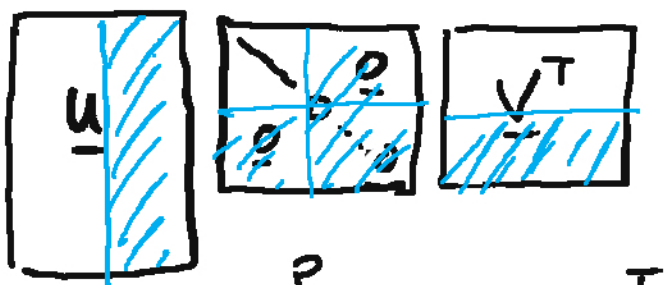
$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{N,m\}} \geq 0$

right sing. vectors

$$\underline{V} = \begin{bmatrix} | & | & & | \\ \underline{v}_1 & \underline{v}_2 & \cdots & \underline{v}_m \\ | & | & & | \end{bmatrix}$$

$\text{rank}(\underline{A}) = p \Leftrightarrow$

$\sigma_1 \geq \cdots \geq \sigma_p > \sigma_{p+1} = \cdots = \sigma_{\min(N,m)} = 0$



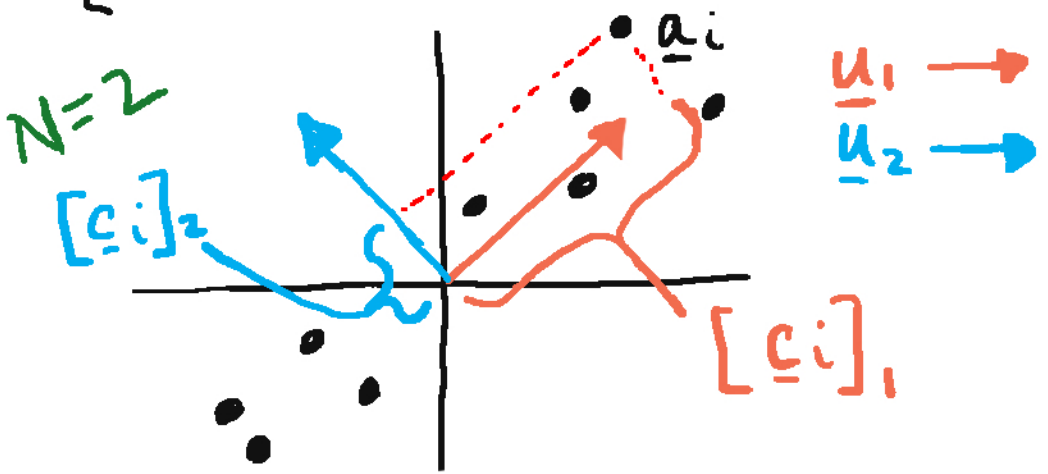
$$\underline{A} = \sum_{i=1}^{p} \sigma_i \underline{u}_i \underline{v}_i^T$$

Singular vectors are o/n bases for rows/columns

$$\begin{bmatrix} | & | & & | \\ \underline{a}_1 & \underline{a}_2 & \cdots & \underline{a}_M \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \underline{u}_1 & \underline{u}_2 & \cdots & \underline{u}_P \\ | & | & & | \end{bmatrix} \underbrace{\begin{bmatrix} | & | & & | \\ \underline{c}_1 & \underline{c}_2 & \cdots & \underline{c}_M \\ | & | & & | \end{bmatrix}}_{\underline{C} = \underline{\Sigma}\,\underline{V}^T}$$

$$\Rightarrow \underline{a}_i = \sum_{j=1}^{P} \underline{u}_j [\underline{c}_i]_j$$

coords of $\underline{a}_i$ in basis $\underline{U}$



left sing. vec. $\underline{U}$: o/n basis cols $\underline{A}$

j^{th} coord $\sim \sigma_j$ $\quad [\underline{c}_i]_j = \sigma_j \underbrace{[\underline{V}^T]_{j,i}}_{\max = 1}$

$$\begin{bmatrix} - & \underline{x}_1^T & - \\ - & \underline{x}_2^T & - \\ & \vdots & \\ - & \underline{x}_N^T & - \end{bmatrix} = \underbrace{\begin{bmatrix} - & \underline{d}_1^T & - \\ - & \underline{d}_2^T & - \\ & \vdots & \\ - & \underline{d}_N^T & - \end{bmatrix}}_{\underline{D} = \underline{U}\,\underline{\Sigma}} \begin{bmatrix} - & \underline{V}_1^T & - \\ - & \underline{V}_2^T & - \\ & \vdots & \\ - & \underline{V}_P^T & - \end{bmatrix}$$

$$\Rightarrow \underline{x}_i^T = \sum_{j=1}^{P} \underline{V}_j^T [\underline{d}_i]_j$$

right sing. vec. $\underline{V}$: o/n basis for rows of $\underline{A}$

j^{th} coord $\sim \sigma_j$ $\quad [\underline{d}_i]_j = \sigma_j [\underline{U}]_{i,j}$

SVD gives inverse of square matrices

$N=M$ $\quad \underline{A} = \underline{U}\,\underline{\Sigma}\,\underline{V}^T \quad \underline{U}, \underline{\Sigma}, \underline{V} : N \times N$

Noninvertible (Singular): $\operatorname{rank}(\underline{A}) < N$

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > \sigma_{p+1} = \cdots = \sigma_N = 0$$

Invertible: $\operatorname{rank}(\underline{A}) = N \quad \underline{A}^{-1} = \underline{V}\,\underline{\Sigma}^{-T}\,\underline{U}^T$

$$\underline{A}\cdot\underline{A}^{-1} = \underline{U}\,\underline{\Sigma}\,\underline{V}^T\,\underline{V}\,\underline{\Sigma}^{-T}\,\underline{U}^T = \underline{U}\,\underline{\Sigma}\,\underline{\Sigma}^{-T}\,\underline{U}^T = \underline{U}\,\underline{\Sigma}\,\underline{\Sigma}^{-1}\,\underline{U}^T$$

$$= \underline{U}\,\underline{U}^T = \underline{I} \quad \text{(no econ SVD for full rank square)}$$

$$\underline{A} = \sum_{i=1}^{N} \sigma_i \underline{u}_i \underline{v}_i^T, \quad \underline{A}^{-1} = \sum_{i=1}^{N} \frac{1}{\sigma_i} \underline{v}_i \underline{u}_i^T$$

SVD of $\underline{A}$ gives SVD of $\underline{A}^{-1}$

The SVD and Least-Squares Problems

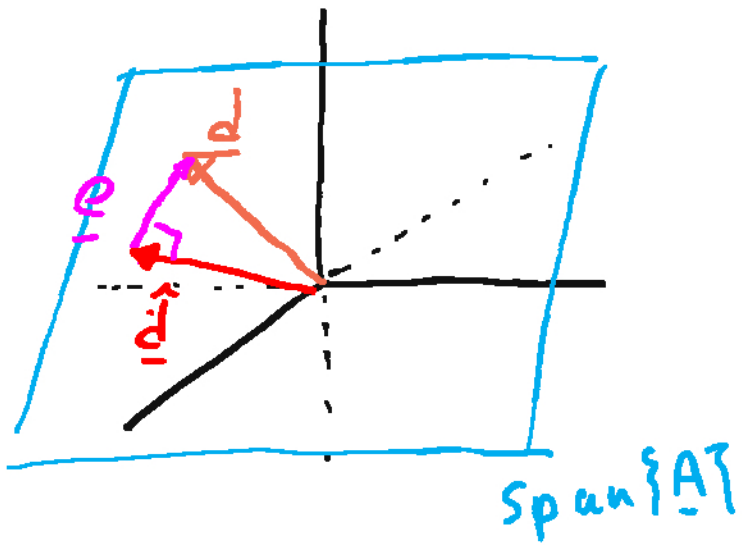
Objectives

- Express least-squares solution in terms of SVD

- Express least-squares error in terms of SVD

- Use SVD to solve the orthobases classification problem

SVD gives insight into the least-squares problem



$$\min_{\underline{w}} \|\underline{d} - \underline{A}\underline{w}\|_2^2 \qquad \underline{e} = \underline{d} - \underline{A}\underline{w}$$

$\underline{A}$: $N \times p$ $\quad \text{rank}(\underline{A}) = p$

$\underline{d}$: $N \times 1$

$$\underline{w} = (\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{d}$$

$$\hat{\underline{d}} = \underline{A}\underline{w} = \underline{A}(\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{d} = \underline{P}_A\underline{d}$$

$$\underline{e} = \underline{d} - \hat{\underline{d}} = (\underline{I} - \underline{P}_A)\underline{d} = \underline{P}_{A^\perp}\underline{d}$$

SVD -

$$\underline{A} = \begin{bmatrix} \tilde{\underline{U}} & \underline{U}_\perp \end{bmatrix} \begin{bmatrix} \underline{\Sigma} & \underline{0} \\ \underline{0} & \underline{0} \end{bmatrix} \underline{V}^T$$

$$\underline{U} = \underset{N \times N}{\left[\tilde{\underline{U}} : \underline{U}_\perp\right]}$$

$$\underline{A} = \tilde{\underline{U}}\,\underline{\Sigma}\,\underline{V}^T$$

Least-squares solution

$$\underline{w} = (\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{d} = (\underline{V}\underline{\Sigma}^T\tilde{\underline{U}}^T\tilde{\underline{U}}\underline{\Sigma}\underline{V}^T)^{-1}\underline{V}\underline{\Sigma}^T\tilde{\underline{U}}^T\underline{d}$$

$$= (\underline{V}\underline{\Sigma}^2\underline{V}^T)^{-1}\underline{V}\underline{\Sigma}\tilde{\underline{U}}^T\underline{d}$$

recall $(\underline{E}\underline{F}\underline{G})^{-1} = \underline{G}^{-1}\underline{F}^{-1}\underline{E}^{-1}$

$\underline{V}^{-1} = \underline{V}^T$

$$= \underline{V}\underline{\Sigma}^{-2}\underline{V}^T\underline{V}\underline{\Sigma}\tilde{\underline{U}}^T\underline{d}$$

$$= \underline{V}\underline{\Sigma}^{-1}\tilde{\underline{U}}^T\underline{d} = \sum_{i=1}^{p}\frac{1}{\sigma_i}\underline{v}_i(\tilde{\underline{u}}_i^T\underline{d})$$

$\underline{V}\underline{\Sigma}^{-1}\tilde{\underline{U}}^T$ = psuedo-inverse of $\underline{A}$

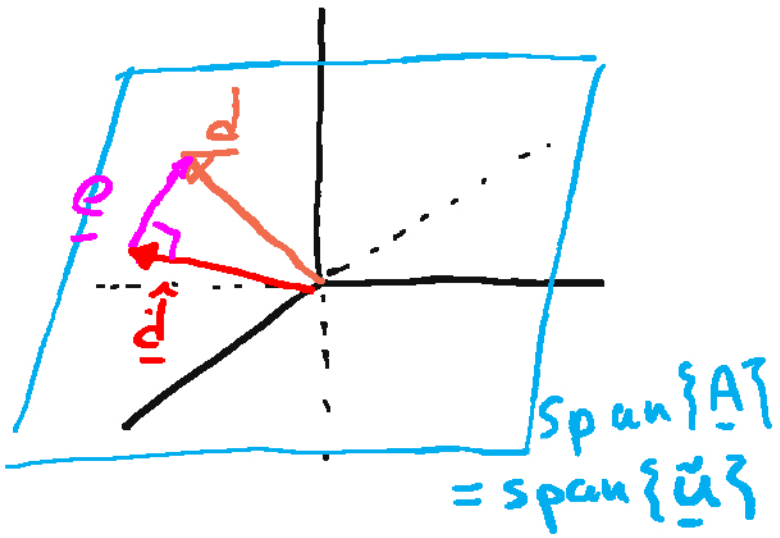
$$\underline{\Sigma}^{-2}\underline{\Sigma} = \begin{bmatrix} 1/\sigma_1^2 & & \underline{0} \\ & 1/\sigma_2^2 & \\ \underline{0} & & \ddots \\ & & & 1/\sigma_p^2 \end{bmatrix}\begin{bmatrix} \sigma_1 & & \underline{0} \\ & \sigma_2 & \\ \underline{0} & & \ddots \\ & & & \sigma_p \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sigma_1 & & \underline{0} \\ & 1/\sigma_2 & \\ \underline{0} & & \ddots \\ & & & 1/\sigma_p \end{bmatrix} = \underline{\Sigma}^{-1}$$

$$\left[(\underline{A}^T\underline{A})^{-1}\underline{A}^T\right]\underline{A} = (\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{A} = \underline{I}$$

$$\underline{V}\underline{\Sigma}^{-1}\tilde{\underline{U}}^T\tilde{\underline{U}}\underline{\Sigma}\underline{V}^T = \underline{V}\underline{\Sigma}^{-1}\underline{\Sigma}\underline{V}^T = \underline{V}\underline{V}^T = \underline{I}$$

Least-squares error and projections



$$\hat{d} = A\left[(A^TA)^{-1}A^T\right]d = P_A d$$

$$= \tilde{U}\Sigma V^T\left[V\Sigma^{-1}\tilde{U}^T\right]d = \tilde{U}\Sigma\Sigma^{-1}\tilde{U}^T d$$

$$= \tilde{U}\tilde{U}^T d \Rightarrow P_A = \tilde{U}\tilde{U}^T$$

$$e = d - \hat{d} = (I - \tilde{U}\tilde{U}^T)d = P_{A\perp}d \Rightarrow P_{A\perp} = I - \tilde{U}\tilde{U}^T$$

Recall $U = [\tilde{U} \mid U_\perp]$ $(N \times N)$ $U^TU = UU^T = I = [\tilde{U} \mid U_\perp]\begin{bmatrix}\tilde{U}^T \\ U_\perp^T\end{bmatrix}$

So $I = \tilde{U}\tilde{U}^T + U_\perp U_\perp^T \Rightarrow P_{A\perp} = U_\perp U_\perp^T$

Classification using SVD o/n bases

Training

$$\begin{bmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_N^T \end{bmatrix} \underline{w} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \Rightarrow \underline{A}\underline{w} = \underline{d}$$

features labels

Prediction

$$\tilde{y} = \text{sign}(\tilde{\underline{x}}^T \underline{w})$$

Orthobases

$$\underline{A} = \tilde{\underline{U}} \underline{\Sigma} \underline{V}^T$$

$$\tilde{\underline{U}} \underline{w}' = \underline{d}$$

$$\underline{w}' = \tilde{\underline{U}}^T \underline{d}$$

$$\underline{w} = \underline{V} \underline{\Sigma}^{-1} \tilde{\underline{U}}^T \underline{d} = \underline{V} \underline{\Sigma}^{-1} \underline{w}'$$

$|w_i'|$ indicates importance of i^{th} ortho feature

Orthobases Prediction

$$\tilde{y} = \text{sign}(\tilde{\underline{x}}^T \underline{w}) = \text{sign}(\tilde{\underline{x}}^T \underline{V} \underline{\Sigma}^{-1} \underline{w}')$$

$$\tilde{\underline{x}}'^T = \tilde{\underline{x}}^T \underline{V} \underline{\Sigma}^{-1} \Rightarrow \tilde{y} = \text{sign}(\tilde{\underline{x}}'^T \underline{w}')$$

transformed feature

orthobasis classifier

SVD and Regularization of Least-Squares Problems

Objectives

- Analyze impact of errors in least-squares problems using SVD
- Introduce truncated SVD regularization
- Analyze ridge regression using SVD

Ill-conditioned least-squares problems

have small singular values

$$\min_{\underline{w}} \|\underline{A}\underline{w} - \underline{d}\|_2^2 \Rightarrow \underline{w} = (\underline{A}^T\underline{A})^{-1}\underline{A}^T\underline{d}$$

SVD: $\underline{A} = \underline{U}\underline{\Sigma}\underline{V}^T \Rightarrow$

$N \times P$, rank P

$$\underline{w} = \underline{V}\underline{\Sigma}^{-1}\underline{U}^T\underline{d} = \sum_{i=1}^{P} \frac{1}{\sigma_i} \underline{v}_i(\underline{u}_i^T\underline{d})$$

$$\Rightarrow \|\underline{w}\|_2^2 = \sum_{i=1}^{P} \left(\frac{1}{\sigma_i}\right)^2 (\underline{u}_i^T\underline{d})^2$$

Small $\sigma_i \Rightarrow$ large $\|\underline{w}\|_2$

Prediction with errors: $\tilde{y} = (\tilde{\underline{x}} + \underline{\varepsilon})^T\underline{w} = \underline{x}^T\underline{w} + \underline{\varepsilon}^T\underline{w}$

$$|\underline{\varepsilon}^T\underline{w}|^2 = \|\underline{w}\|_2^2 \|\underline{\varepsilon}\|_2^2 \cos^2\theta$$

large $\|\underline{w}\|_2^2 \Rightarrow$ sensitive to errors

What if rank$(\underline{A}) < P$?

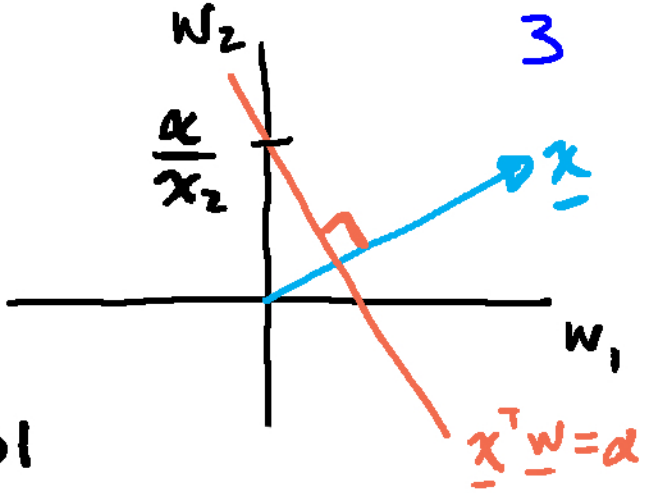
$\sigma_P = 0$

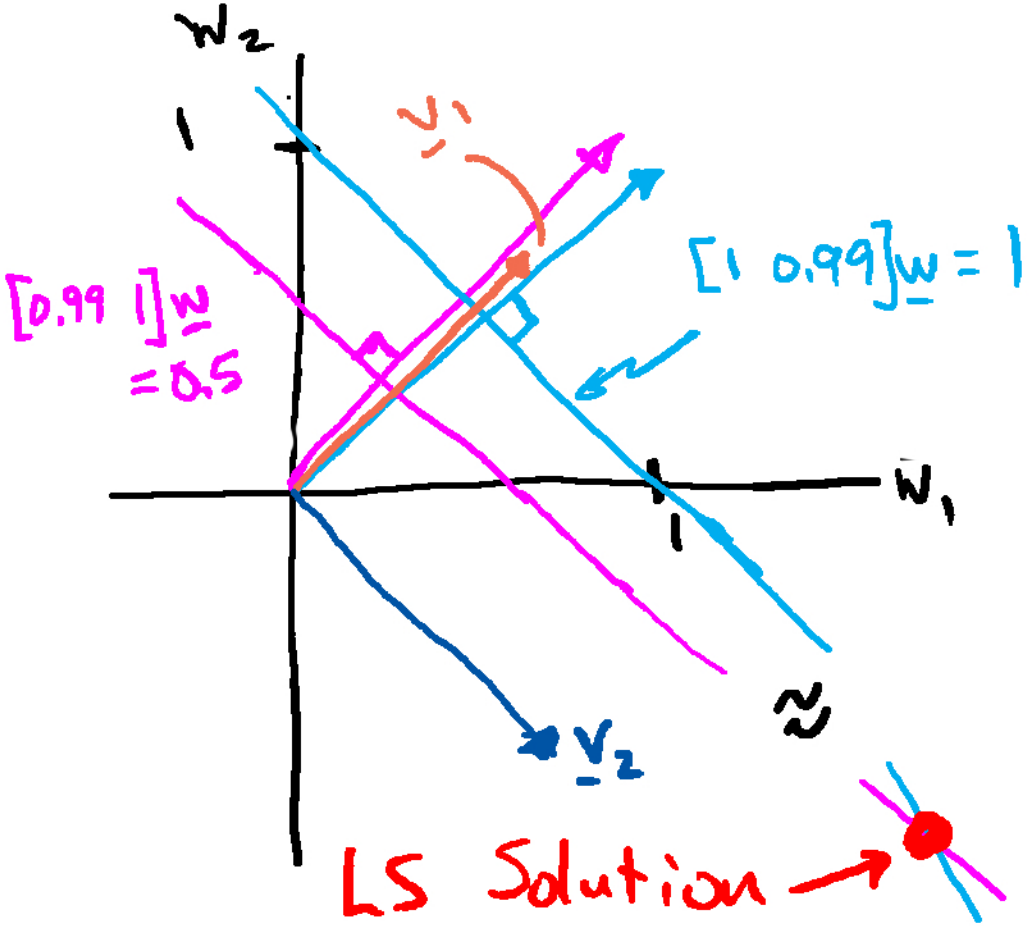
no unique solution

Example: ill-conditioned $\underline{A}$

$$\underline{A} = \begin{bmatrix} 1 & 0.99 \\ 0.99 & 1 \end{bmatrix}, \quad \underline{d} = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$$

Geometry $\underline{x}^T \underline{w} = \alpha$





$\sigma_1 = 1.99 \quad \sigma_2 = 0.01$

$1/\sigma_1 \simeq 0.5, \quad 1/\sigma_2 = 100$

$\underline{v}_1 = \frac{1}{\sqrt{2}} [1 \quad 1]^T \quad \underline{v}_2 = \frac{1}{\sqrt{2}} [1 \quad -1]^T$

$\underline{u}_1^T \underline{d} = \frac{1.5}{\sqrt{2}}, \quad \underline{u}_2^T \underline{d} = \frac{0.5}{\sqrt{2}}$

$$\underline{w} = \underline{v}_1 \frac{\underline{u}_1^T \underline{d}}{\sigma_1} + \underline{v}_2 \frac{\underline{u}_2^T \underline{d}}{\sigma_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} 3/8 + \begin{bmatrix} 1 \\ -1 \end{bmatrix} 25$$

$\|w\|_2 \uparrow$ as $\sigma_2 \downarrow$ ($\nearrow \rightarrow \nearrow$)

Regularized LS via truncated SVD

Replace $\sum_{i=1}^{p} \frac{1}{\sigma_i} \underline{v}_i (\underline{u}_i^T \underline{d})$ with $\sum_{i=1}^{r} \frac{1}{\sigma_i} \underline{v}_i (\underline{u}_i^T \underline{d})$

where $r < p$.

- Avoid inverting small/zero singular values
- Equivalent to replacing $\underline{A} = \sum_{i=1}^{p} \sigma_i \underline{u}_i \underline{v}_i^T$ with the rank-r approximation $\underline{A}_r = \sum_{i=1}^{r} \sigma_i \underline{u}_i \underline{v}_i^T$
- Increases $\min_{\underline{w}} \|\underline{A}\underline{w} - \underline{d}\|_2^2$
- Can choose r using intuition or cross-validation

Regularized LS via ridge regression

$$\min_{\underline{w}} \|\underline{A}\underline{w}-\underline{d}\|_2^2 + \lambda\|\underline{w}\|_2^2 \Rightarrow \underline{w} = (\underline{A}^T\underline{A}+\lambda\underline{I})^{-1}\underline{A}^T\underline{d}$$

controls norm!

Use SVD: $\underline{A}^T\underline{A} = \underline{V}\underline{\Sigma}^2\underline{V}^T$, $\lambda\underline{I} = \underline{V}\lambda\underline{I}\underline{V}^T$

$$\underline{w} = \left(\underline{V}(\underline{\Sigma}^2+\lambda\underline{I})\underline{V}^T\right)^{-1}\underline{V}\underline{\Sigma}\underline{U}^T\underline{d} = \underline{V}\underbrace{(\underline{\Sigma}^2+\lambda\underline{I})^{-1}\underline{\Sigma}}_{\underline{D}}\underline{U}^T\underline{d}$$

$$\underline{D} = \begin{bmatrix} \frac{1}{\sigma_1^2+\lambda} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_p^2+\lambda} \end{bmatrix}\begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \end{bmatrix} = \begin{bmatrix} \sigma_1/\sigma_1^2+\lambda & & 0 \\ & \ddots & \\ 0 & & \frac{\sigma_p}{\sigma_p^2+\lambda} \end{bmatrix}$$

Controlled!

$$\underline{w} = \sum_{i=1}^{p} \frac{\sigma_i}{\sigma_i^2+\lambda}\underline{v}_i(\underline{u}_i^T\underline{d})$$

- as $\sigma_i \to 0$, $\frac{\sigma_i}{\sigma_i^2+\lambda} \to \sigma_i/\lambda$
- increased value $\|\underline{A}\underline{w}-\underline{d}\|_2^2$

Principal Component Analysis

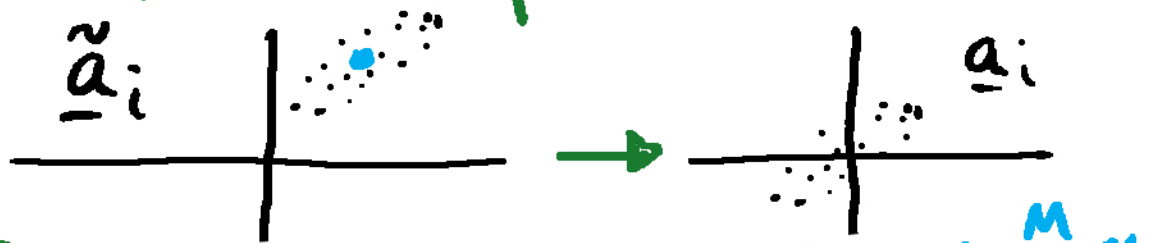
Objectives

- Define principal components
- Relate principal components to singular vectors
- Relate geometry of data matrix to singular vectors and singular values

PCA represents maximum "variance"

Data: $\underline{a}_i$, $i=1,2,\cdots M$ $(N\times 1)$ vectors, $\underline{A}=[\underline{a}_1\ \underline{a}_2\cdots\underline{a}_M]$

- PCA assumes zero mean $\Rightarrow$ 1st step: center data



$$\underline{a}_i=\tilde{\underline{a}}_i-\frac{1}{M}\sum_{j=1}^{M}\tilde{\underline{a}}_j$$

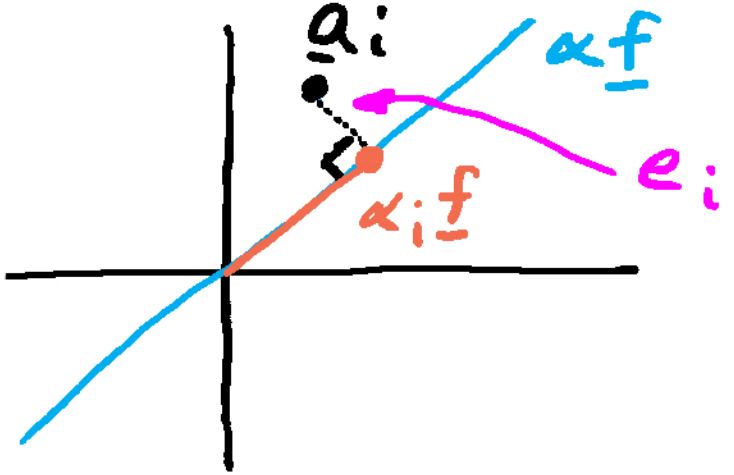
- First principal component: direction $\underline{f}$ accounting for maximum variance in data, $\|\underline{f}\|_2^2=1$

$$\max_{\|\underline{f}\|_2^2=1}\left\{\frac{1}{M}\sum_{i=1}^{M}\|\alpha_i\underline{f}\|_2^2\right\}$$ best line

$$\min_{\alpha_i}\|\underline{a}_i-\alpha_i\underline{f}\|_2^2 \Rightarrow \alpha_i=\underline{f}^T\underline{a}_i$$

$$\max_{\|\underline{f}\|_2^2=1}\left\{\frac{1}{M}\sum_{i=1}^{M}|\underline{f}^T\underline{a}_i|^2\right\}$$

$$\|\underline{f}^T\underline{A}\|_2^2=\|\underline{A}^T\underline{f}\|_2^2$$



Principal Components are singular vectors

$$\max_{\|\underline{f}\|_2^2=1} \frac{1}{M} \underline{f}^T \underline{A}\underline{A}^T \underline{f} \Rightarrow \underline{A} = \underline{U}\underline{\Sigma}\underline{V}^T \qquad \underline{A}\underline{A}^T = \underline{U}\underline{\Sigma}\underline{V}^T\underline{V}\underline{\Sigma}\underline{U}^T = \underline{U}\underline{\Sigma}^2\underline{U}^T$$

$$\max_{\|\underline{f}\|_2^2=1} \frac{1}{M} \underline{f}^T \underline{U}\underline{\Sigma}^2\underline{U}^T \underline{f} \Rightarrow \underline{f} = \underline{u}_1 \quad \text{(notes)} \quad \text{"best line"}$$

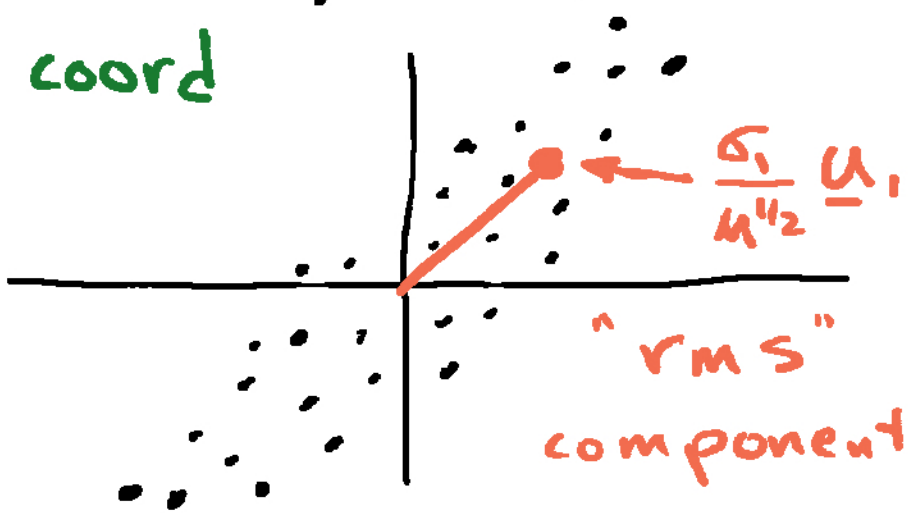
Variance associated w. 1st PC $\frac{1}{M}\underline{u}_1^T \underline{A}\underline{A}^T \underline{u}_1 = \frac{\sigma_1^2}{M}$

Coordinates of data: $\alpha_i = \underline{u}_1^T \underline{a}_i$, $\underline{\alpha}^T = [\alpha_1 \alpha_2 \cdots \alpha_M]$

$$\underline{\alpha}^T = \underline{u}_1^T \underline{A} = \underline{u}_1^T \underline{U}\underline{\Sigma}\underline{V}^T = \sigma_1 \underline{V}_1^T$$

root mean square coord

$$\left(\frac{1}{M}\sum_{i=1}^{M} |\alpha_i|^2\right)^{1/2} = \frac{1}{M^{1/2}}\|\underline{\alpha}\|_2 = \frac{\sigma_1}{M^{1/2}}$$



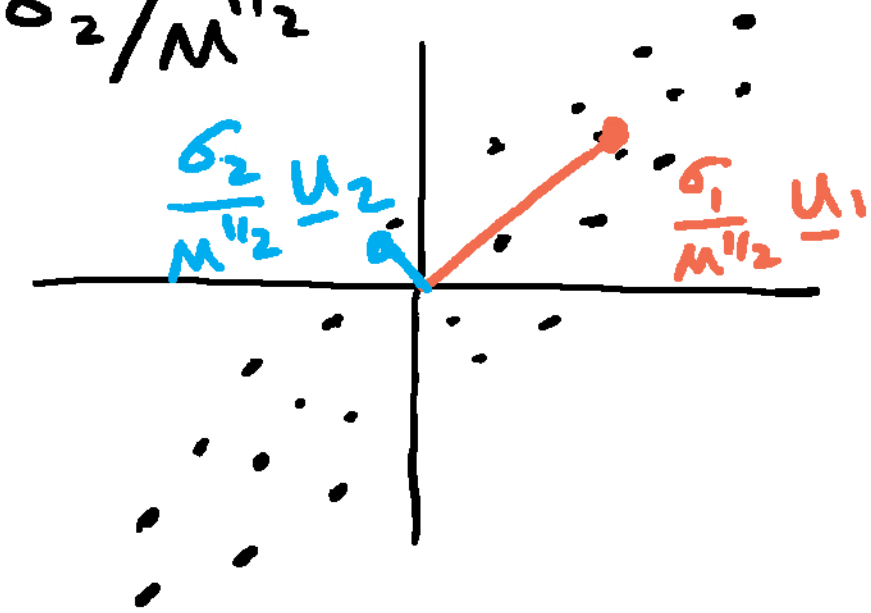
PC are singular vectors

2nd PC: $$\max_{\|g\|_2^2=1,\ g^T u_1=0} \frac{1}{M}\sum_{i=1}^{M} |g^T a_i|^2$$

$\Rightarrow g = u_2$ (2nd left singular vector)

Variance associated w. 2nd PC: $\frac{1}{M} u_2^T A A^T u_2 = \frac{\sigma_2^2}{M}$

RMS value of 2nd PC coord: $\sigma_2/M^{1/2}$



k^{th} PC: u_k

k^{th} PC Variance: $\frac{\sigma_k^2}{M}$

k^{th} PC coord RMS: $\frac{\sigma_k}{M^{1/2}}$

Summary $\underline{A} = \underline{U}\underline{\Sigma}\underline{V}^T$

- Left sing. vectors $\longleftrightarrow$ PC for columns of $\underline{A}$
- Sing. values $\longleftrightarrow$ ~ RMS value PC coords
- PC for rows of $\underline{A}$
  - use columns of $\underline{A}^T = \underline{V}\underline{\Sigma}\underline{U}^T$
  - Right sing. vectors $\underline{v}_i$ are PC
  - Sing. values ~ RMS value PC coords
- Eckhart-Young: SVD gives best rank r approximation to $\underline{A}$ $\quad \underline{A} \approx \sum_{k=1}^{r} \sigma_k \underline{u}_k \underline{v}_k^T$

$$r=1 \quad \underline{A} \approx \underline{u}_1 (\sigma_1 \underline{v}_1^T)$$

Bias-Variance Tradeoff in Low-Rank Approximations

Objectives

- Introduce concept of noisy data
- Consider impact of noise on SVD
- Define bias and variance
- Use low-rank models to trade bias for variance

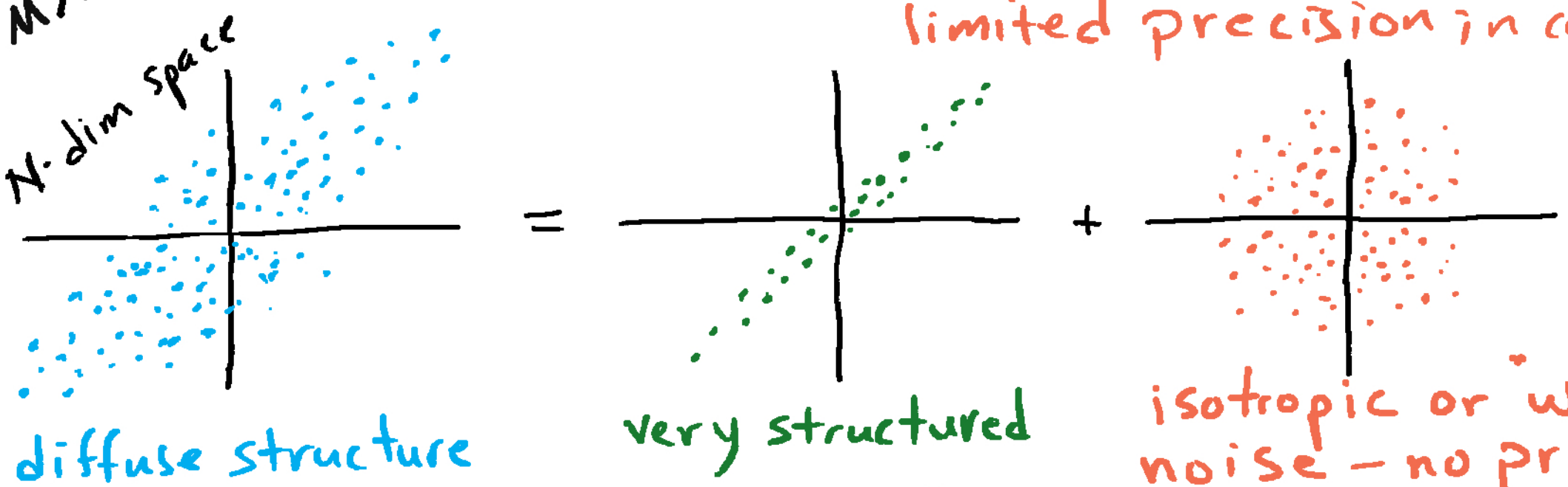
Data is often contaminated by noise

$N \times M$
$M > N$

$\underline{A} = \underline{S} + \underline{G}$

measured = clean + noise

electronics in sensing systems
enviromental static
limited precision in computers



diffuse structure

very structured

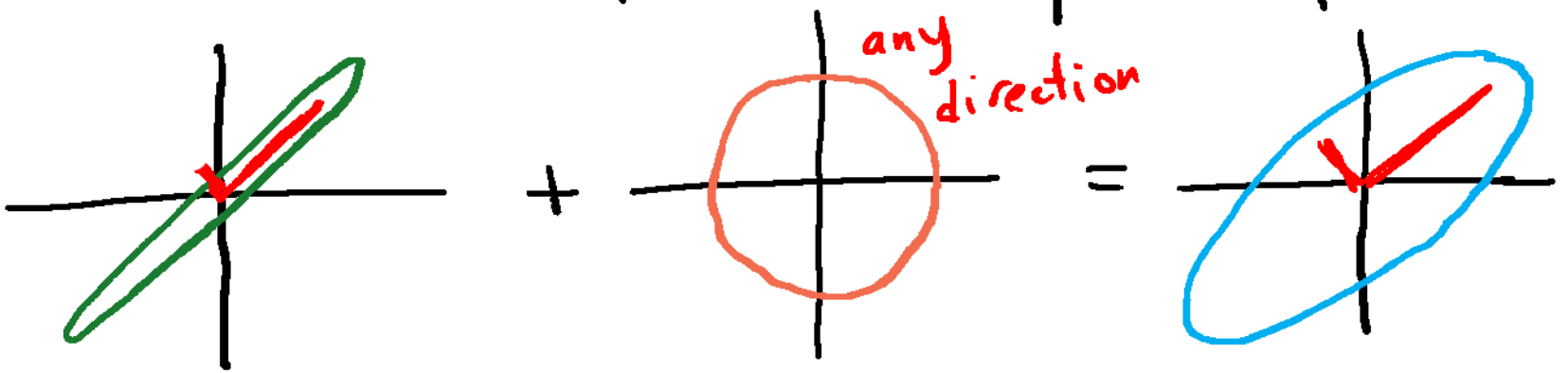
isotropic or "white" noise - no preferred direction

Sum of squared errors: $\|\underline{G}\|_F^2$

$$\|\underline{G}\|_F^2 = \sum_{i=1}^{N} M\left(\frac{1}{M}\sum_{j=1}^{M} g_{ij}^2\right) = M\sum_{i=1}^{N} \text{var}_i \overset{\text{isotropic}}{\approx} MN\sigma_g^2$$

Singular vectors are invariant (approx) to isotropic noise

- Proof uses probability concepts



principal comp directions are unchanged ($M \to \infty$)

variance along each component (singvals) changes

$$\underline{U}\,\underline{\Sigma}_A\,\underline{V}^T \approx \underline{U}\,\underline{\Sigma}_S\,\underline{V}^T + \underline{U}\,\underline{\Sigma}_G\,\underline{V}^T \qquad \sigma_{A_i} \approx \sigma_{S_i} + M^{1/2}\sigma_g$$

$$\underline{\Sigma}_A = \begin{bmatrix} \sigma_{A_1} & & 0 \\ & \sigma_{A_2} & \\ 0 & & \ddots \\ & & & \sigma_{A_N} \end{bmatrix}, \quad \underline{\Sigma}_S = \begin{bmatrix} \sigma_{S_1} & & 0 \\ & \sigma_{S_2} & \\ 0 & & \ddots \\ & & & \sigma_{S_N} \end{bmatrix}, \quad \Sigma_G \approx \begin{bmatrix} M^{1/2}\sigma_g & & 0 \\ & M^{1/2}\sigma_g & \\ 0 & & \ddots \\ & & & M^{1/2}\sigma_g \end{bmatrix}$$

(isotropic)

$\sigma_{g_i} = M^{1/2} \cdot \mathrm{RMS}$

Low-rank models trade bias for variance

Original: $\underline{A} = \underline{S} + \underline{G}$ Error: $\underline{A} - \underline{S}$, $\|\underline{G}\|_F^2 \approx NM\sigma_g^2$

Low rank: $\hat{\underline{A}}_r = \sum_{i=1}^{r} \sigma_{A_i} \underline{u}_i \underline{v}_i^T \approx \hat{\underline{S}}_r + \hat{\underline{G}}_r$

$$\hat{\underline{S}}_r = \sum_{i=1}^{r} \sigma_{S_i} \underline{u}_i \underline{v}_i^T \qquad \hat{\underline{G}}_r \approx \sum_{i=1}^{r} M^{1/2} \sigma_g \underline{u}_i \underline{v}_i^T$$

Bias^2: $b^2(r) = \|\underline{S} - \hat{\underline{S}}_r\|_F^2$

$$b^2(r) = \left\| \sum_{i=r+1}^{N} \sigma_{S_i} \underline{u}_i \underline{v}_i^T \right\|_F^2$$

$$= \sum_{i=r+1}^{N} \sigma_{S_i}^2 \quad \text{(notes)}$$

sum of squared "tail" singular values

Variance: $V(r) = \|\hat{\underline{G}}_r\|_F^2$

$$V(r) = \left\| \sum_{i=1}^{r} M^{1/2} \sigma_g \underline{u}_i \underline{v}_i^T \right\|_F^2$$

$$= rM\sigma_g^2$$

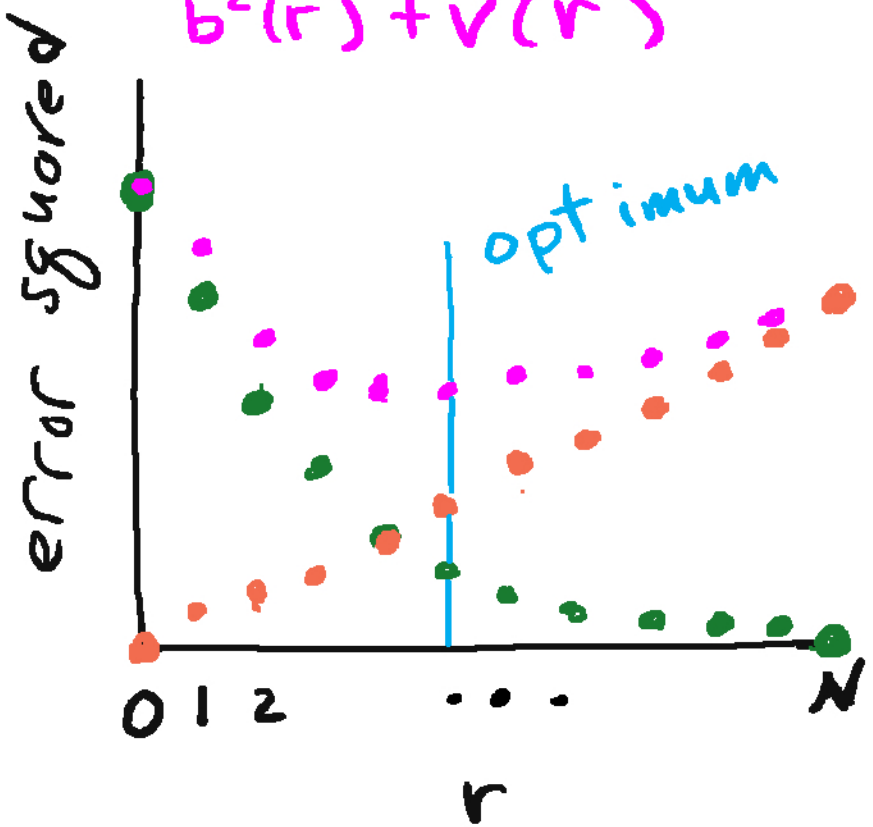
dimensions × $\frac{\text{variance}}{\text{dimension}}$

Trading bias for variance

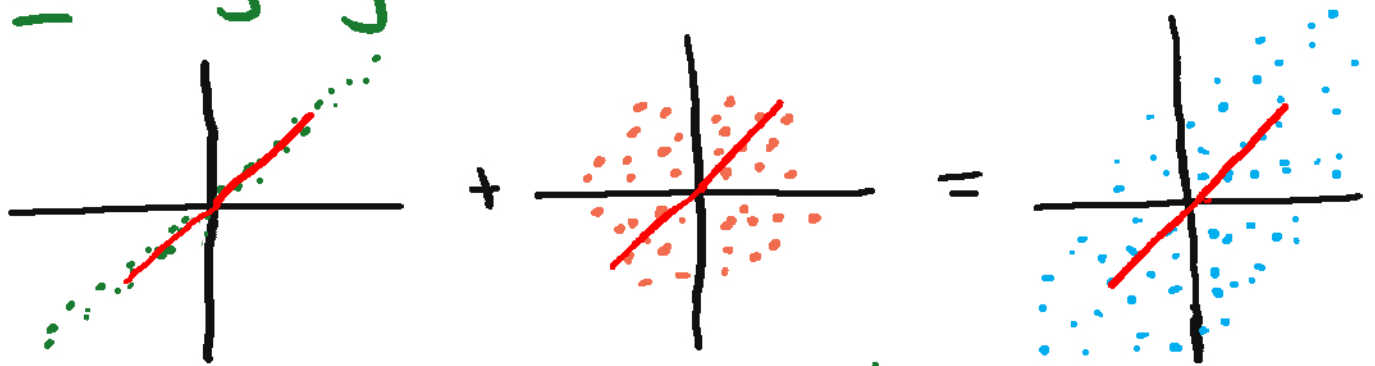
$$b^2(r) = \sum_{i=r+1}^{N} \sigma_{s_i}^2$$ decreases as r increases

$$v(r) = r M \sigma_g^2$$ increases as r increases

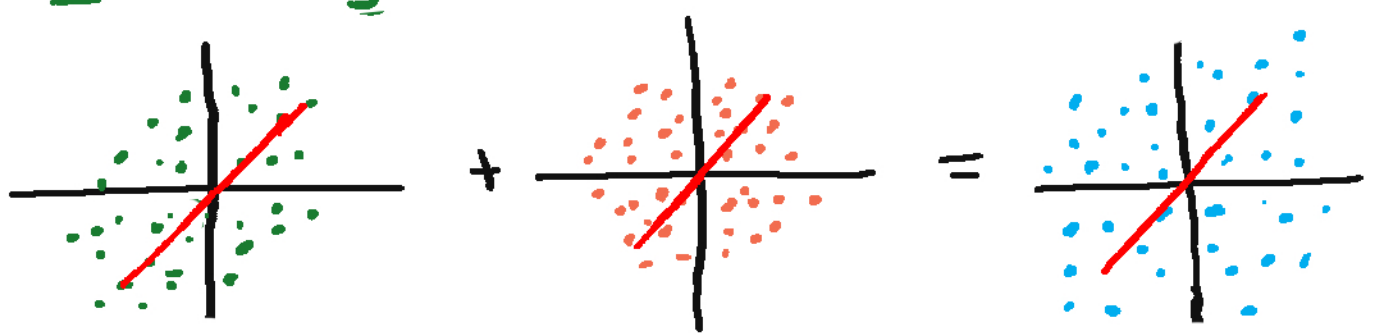
$b^2(r) + v(r)$



≤ highly structured $\sigma_1 \gg \sigma_2$



≤ weakly structured $\sigma_1 \approx \sigma_2$



Eigendecomposition, SVD, and Power Iterations

Objectives

- Define eigenvectors and eigenvalues
- Relate the eigendecomposition to SVD
- Power iterations for computing eigenvector with largest eigenvalue

Eigendecomposition applies to square matrices

Eigenvector $\underline{e}_i$, eigenvalue λ_i, $\underline{B}$ $(k \times k)$

$$\underline{B}\,\underline{e}_i = \lambda_i\,\underline{e}_i$$

matrix mult $\Longleftrightarrow$ scalar mult

$$\underline{e}_i \longrightarrow \boxed{\underline{B}} \longrightarrow \lambda_i\,\underline{e}_i \qquad i = 1, 2, \cdots, k$$

- K eigenvalues, possibly complex valued
- Distinct $\lambda_i \Rightarrow$ linearly independent $\underline{e}_i$
- Symmetric $\underline{B} \Rightarrow$ K orthonormal $\underline{e}_i$ $\quad \underline{E}\,\underline{E}^T = \underline{E}^T\underline{E} = \underline{I}$

$$\underline{B}\,\underline{e}_i = \lambda_i \underline{e}_i \Rightarrow \underline{B}\left[\underline{e}_1\ \underline{e}_2 \cdots \underline{e}_k\right] = \left[\underline{e}_1\ \underline{e}_2 \cdots \underline{e}_k\right]\begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & 0 \\ & 0 & \ddots \\ & & & \lambda_k \end{bmatrix}$$

$$\underline{B}\,\underline{E} = \underline{E}\,\underline{\Lambda} \Rightarrow \underline{B} = \underline{E}\,\underline{\Lambda}\,\underline{E}^T = \sum_{i=1}^{k} \lambda_i \underline{e}_i \underline{e}_i^T$$

Symmetric PSD matrices and SVD

$$\underline{A} = [\underline{a}_1\ \underline{a}_2 \cdots \underline{a}_M] = \begin{bmatrix} \underline{x}_1^T \\ \underline{x}_2^T \\ \vdots \\ \underline{x}_N^T \end{bmatrix} = \underline{U}\,\underline{\Sigma}\,\underline{V}^T \quad (N \times M,\ N > M)$$

full SVD

1) $\underline{B} = \underline{A}\,\underline{A}^T = \sum_{i=1}^{M} \underline{a}_i \underline{a}_i^T$

$$\underline{B} = \underline{U}\underline{\Sigma}\underline{V}^T\underline{V}\,\underline{\Sigma}^T\underline{U}^T = \underline{U}\,\underline{\Sigma}\,\underline{\Sigma}^T\underline{U}^T = \underline{U}\begin{bmatrix} \sigma_1^2 & & & \underline{0} \\ & \ddots & & \\ & & \sigma_M^2 & \\ \underline{0} & & & 0 \cdots 0 \end{bmatrix}\underline{U}^T$$

left SV of $\underline{A}$ $\Leftrightarrow$ eigenvectors $\underline{B}$

$$\lambda_i = \begin{cases} \sigma_i^2 & i = 1, 2, \cdots, M \\ 0 & i = M+1, \cdots, N \end{cases}$$

2) $\underline{B} = \underline{A}^T\underline{A} = \sum_{i=1}^{N} \underline{x}_i \underline{x}_i^T$

$$= \underline{V}\,\underline{\Sigma}^T\underline{U}^T\underline{U}\,\underline{\Sigma}\,\underline{V}^T = \underline{V}\,\underline{\Sigma}^T\underline{\Sigma}\,\underline{V}^T = \underline{V}\begin{bmatrix} \sigma_1^2 & & \underline{0} \\ & \ddots & \\ \underline{0} & & \sigma_M^2 \end{bmatrix}\underline{V}^T$$

right SV of $\underline{A}$ $\Leftrightarrow$ eigenvectors $\underline{B}$, $\lambda_i = \sigma_i^2$, $i = 1, 2, \cdots M$

Power iteration for computing 1st principal component

$\underline{A}$: $N \times M$, $N \gg M$ want $\underline{v}_1$ 1st principal component

right SV of $\underline{A}$, eigenvector of $\underline{B} = \underline{A}^T\underline{A}$ $(M \times M)$

Power Iteration

pick $\underline{c}_0$ (random)

for $k = 1, 2, \ldots$ to converge

$$\underline{c}_k = \underline{B}\underline{c}_{k-1} / \|\underline{B}\underline{c}_{k-1}\|_2$$

end

$\underline{v}_1 = \underline{c}_{end}$

$$\underline{B}\underline{c}_{k-1} = \underbrace{\underline{B}\cdot\underline{B}\cdots\underline{B}}_{k \text{ times}}\underline{c}_0 = \underline{B}^k\underline{c}_0$$

$$\underline{B}^k = \underbrace{\underline{V}\underline{\Lambda}\underline{V}^T\underline{V}\underline{\Lambda}\underline{V}^T\cdots\underline{V}\underline{\Lambda}\underline{V}^T}_{k \text{ times}}$$

$$= \underline{V}\underline{\Lambda}^k\underline{V}^T$$

Let $\underline{c}_0 = \underline{V}\underline{g} = \underline{V}\begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_M \end{bmatrix}$

$$\underline{B}^k\underline{c}_0 = \underline{V}\underline{\Lambda}^k\underline{V}^T\underline{V}\,\underline{g}$$

$$= \underline{V}\underline{\Lambda}^k\underline{g}$$

Power iteration...

$$\underline{c}_k = \underline{B}\underline{c}_{k-1} / \|\underline{B}\underline{c}_{k-1}\|_2 = \underline{V}\underline{\Lambda}^k \underline{g} / \|\underline{V}\underline{\Lambda}^k \underline{g}\|_2$$

$$\underline{V}\underline{\Lambda}^k \underline{g} = [\underline{v}_1\ \underline{v}_2 \cdots \underline{v}_m] \begin{bmatrix} \lambda_1^k & & \underline{0} \\ & \lambda_2^k & \\ \underline{0} & & \ddots\ \lambda_m^k \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix} \quad \text{but } \frac{\lambda_i}{\lambda_1} < 1$$

$$= \lambda_1^k g_1\, \underline{V} \begin{bmatrix} 1 & & \underline{0} \\ & \left(\frac{\lambda_2}{\lambda_1}\right)^k & \\ \underline{0} & & \ddots\ \left(\frac{\lambda_m}{\lambda_1}\right)^k \end{bmatrix} \begin{bmatrix} 1 \\ g_2/g_1 \\ \vdots \\ g_m/g_1 \end{bmatrix} \quad \text{so } \left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0$$

$$\underline{V}\underline{\Lambda}^k \underline{g} \to \lambda_1^k g_1 \underline{V} \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \end{bmatrix} = \lambda_1^k g_1 \underline{v}_1$$

$$c_k \to \frac{\lambda_1^k g_1 \underline{v}_1}{\|\lambda_1^k g_1 \underline{v}_1\|_2} = \frac{\underline{v}_1}{\|\underline{v}_1\|_2} = \underline{v}_1$$

Network Graphs and the PageRank Algorithm

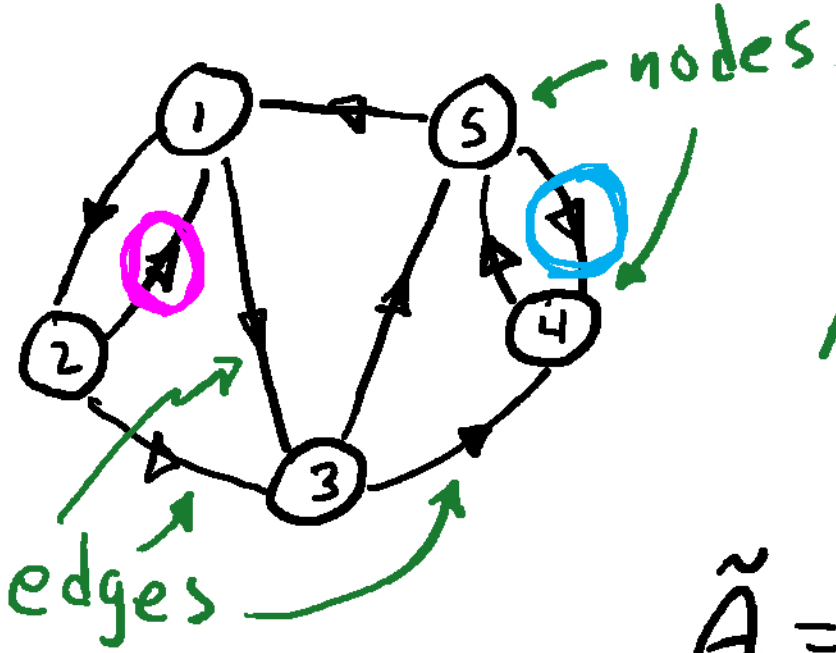
Objectives

- Introduce matrix representations for network graphs

- Define transition probability matrix and paths on graph

- Illustrate PageRank algorithm concepts

Matrices represent network graphs



Examples: webpages/links, cities/roads, routers/wires

Adjacency matrix; connection topology

$$\underline{\tilde{A}} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

from

to

edge from node 2 to node 1

edge from node 5 to node 4

$$\underline{A} = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 1/2 \\ 1/4 & 0 & 0 & 0 & 0 \\ 3/4 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/2 \\ 0 & 0 & 1/3 & 1 & 0 \end{bmatrix}$$

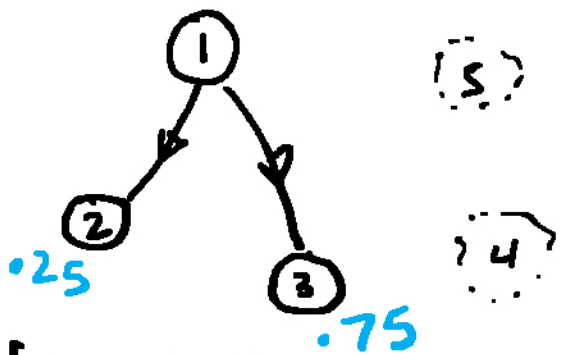
transition probability matrix - columns sum to 1

Transition probability matrix predicts "paths"

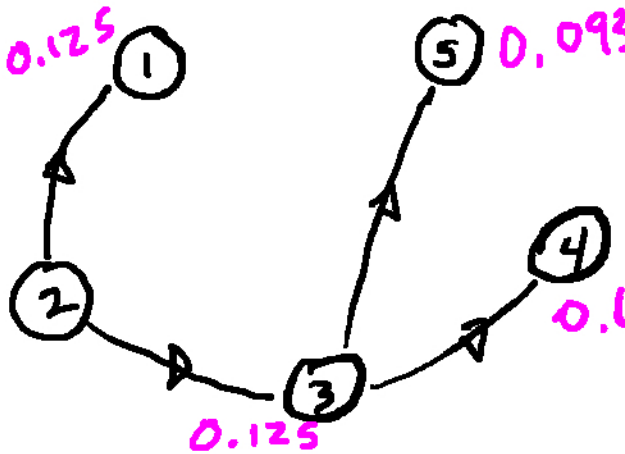
$$\underline{A} = \begin{bmatrix} 0 & 1/2 & 0 & 0 & 1/2 \\ 1/4 & 0 & 0 & 0 & 0 \\ 3/4 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/8 & 0 & 1/2 \\ 0 & 0 & 1/8 & 1 & 0 \end{bmatrix}$$

Start at node 1: $\underline{p}_0 = [1\ 0\ 0\ 0\ 0]^T$

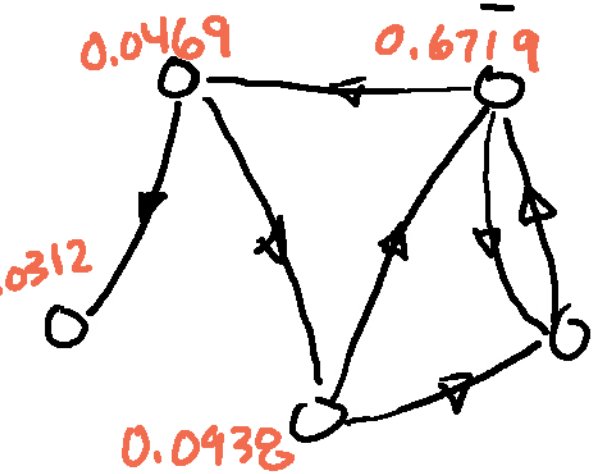
After 1 move: $\underline{p}_1 = \underline{A}\,\underline{p}_0 = \begin{bmatrix} 0 \\ 0.25 \\ 0.75 \\ 0 \\ 0 \end{bmatrix}$ 25% #2, 75% #3



After 2 moves: $\underline{p}_2 = \underline{A}\,\underline{p}_1 = \begin{bmatrix} 0.125 \\ 0 \\ 0.125 \\ 0.6562 \\ 0.0938 \end{bmatrix}$



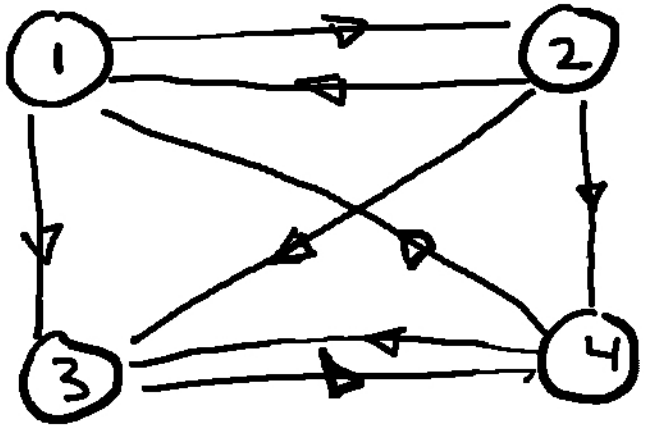
After 3 moves: $\underline{p}_3 = \underline{A}\,\underline{p}_2 = \begin{bmatrix} 0.0469 \\ 0.0312 \\ 0.0938 \\ 0.1562 \\ 0.6719 \end{bmatrix}$



Markov chain: next state depends only on current state

PageRank algorithm ranks web pages

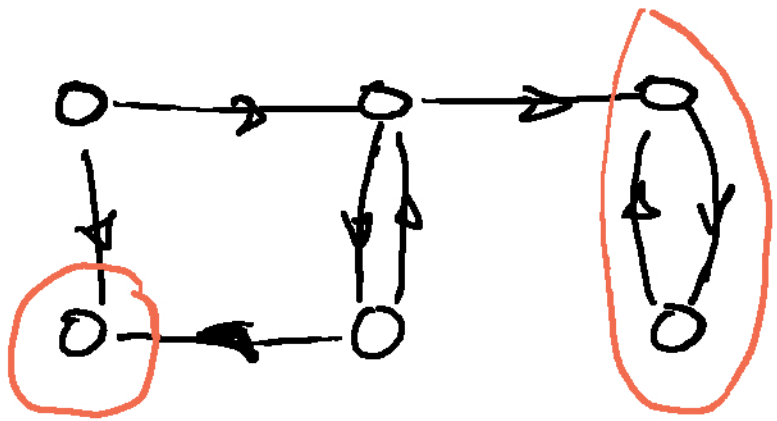
where will I visit most?



1) Adjacency matrix $\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$

2) Normalize columns (equal prob. outlinks) $A = \begin{bmatrix} 0 & 1/3 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 1/3 & 0 & 1/2 \\ 0 & 1/3 & 1 & 0 \end{bmatrix}$

3) Eliminate traps



introduce small probability to go from any node to any other node

Transition matrix: $Q = (1-\alpha)A + \frac{\alpha}{N}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$

$\frac{\alpha}{N}$: random jump probability

Eigenvector of Q ranks pages

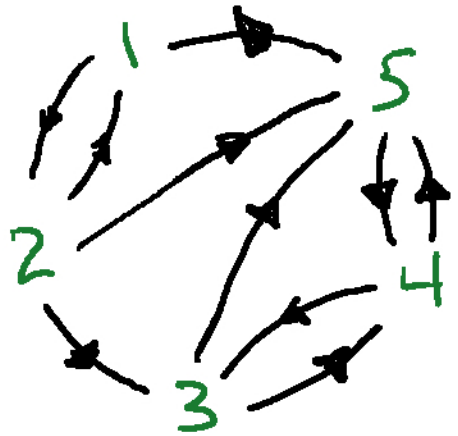
Q is irreducible (no traps) column stochastic (cols sum to 1), with nonnegative entries $\Rightarrow$ (Perron-Frobenius) Largest eval is 1, evect $P = [p_1 \cdots p_N]^T$ satisfies $p_i > 0$, $\sum_i p_i = 1$

$$u = \frac{1}{N}[1\ 1\ \cdots\ 1]^T \qquad \lim_{k\to\infty} Q^k u = P$$

Steady-state Distribution $\quad QP = P$

P ranks importance of pages

Example:



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

$$Q = \begin{bmatrix} .002 & .332 & .002 & .002 & .002 \\ .497 & .002 & .002 & .002 & .002 \\ .002 & .332 & .002 & .497 & .002 \\ .002 & .002 & .497 & .002 & .992 \\ .497 & .332 & .497 & .497 & .002 \end{bmatrix} \quad (\alpha = 0.01)$$

$$u = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix}, \quad Qu = \begin{bmatrix} .07 \\ .10 \\ .17 \\ .30 \\ .36 \end{bmatrix}, \quad Q^2u = \begin{bmatrix} .003 \\ .004 \\ .18 \\ .45 \\ .30 \end{bmatrix}, \cdots \quad Q^{10}u = \begin{bmatrix} .003 \\ .004 \\ .22 \\ .44 \\ .33 \end{bmatrix}$$

$$P = \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix}$$
$$\begin{bmatrix} 0.0032 & 0.0036 & 0.2211 & 0.4401 & 0.3320 \end{bmatrix}$$

Matrix Completion

Objectives

- define the matrix completion problem
- approach missing data using low-rank models
- introduce iterative singular value thresholding

Use "patterns" to fill in missing entries

Ratings matrix

$\underline{X} \in \mathbb{R}^{N \times M}$

5	4	9	1	×
9	6	×	×	7
×	10	×	2	4
3	7	3	×	×
8	×	×	6	2

N movies ↓

M users →

Can we predict the missing entries?

Model: assume $\underline{X}$ is well approximated with a small number of patterns

$$\underline{X} \approx \sum_{i=1}^{r} \underline{t}_i \underline{s}_i^T = \underline{T}\,\underline{S}$$

genres, actors, director...

hobbies, age, address...

Matrix completion: use known data to find patterns and predict missing entries

$\Omega = \{ i,j : \underline{X}_{ij} \text{ given} \}$ indices of known values

1) Rank minimization $\underline{X} = \underset{\underline{M}}{\operatorname{argmin}} \operatorname{rank}(\underline{M})$ s.t. $\underline{M}_{ij} = \underline{X}_{ij}$ $\forall\, i,j \in \Omega$

minimum number of patterns matching given values

Intractable !

$\operatorname{rank} \underline{M} = \#\{ \ell : \sigma_\ell > 0 \}$

2) Nuclear norm minimization

$\underline{X} = \underset{\underline{M}}{\operatorname{argmin}} \|\underline{M}\|_* \text{ s.t. } \underline{M}_{ij} = \underline{X}_{ij} \;\forall\, i,j \in \Omega$

Nuclear norm / trace

$\|M\|_* = \sum_\ell \sigma_\ell$

Computationally tractable

Iterative Singular Value Thresholding is one possible algorithm

Initialize

$\underline{M}^{(0)} = \underline{O}$

Set threshold or r

Iterate

for $k = 1, 2, 3, \ldots$

$\underline{M}^{(k)} = \underline{M}^{(k-1)}$

$\underline{M}_{\Omega}^{(k)} = \underline{X}_{\Omega}$ (fill in known values)

$[\underline{U}, \underline{\Sigma}, \underline{V}] = \text{svd}(\underline{M}^{(k)})$

$\hat{\Sigma}_{ii} = \Sigma_{ii} \cdot \begin{cases} 1 & \Sigma_{ii} > \text{threshold} \\ 0 & \Sigma_{ii} \leq \text{threshold} \end{cases}$

— or —

$\hat{\Sigma}_{ii} = \begin{cases} \Sigma_{ii}, & i \leq r \\ 0, & i \geq r+1 \end{cases}$

$\underline{M}^{(k)} = \underline{U} \hat{\underline{\Sigma}} \underline{V}^T$

if $\| \underline{M}^{(k)} - \underline{M}^{(k-1)} \|_F < \varepsilon$

stop

else

next k

Matrix completion is an open problem

- choosing r or threshold in ISVT
- multiple algorithms:
  - convergence
  - complexity
  - noise
- results depend on distribution of missing entries
- applications include missing pixels in images, position from partial distance info, ...