CS 331, Fall 2025

Lecture 16 (10/22)

Today:
- Low-rank approx
- SVD/PCA
- Power method

Low-rank approximation (Part VI, Section 3.1)

Quiz: What are missing entries?

$$\begin{pmatrix} 7 & ? & ? & 14 & 21 \\ ? & 8 & 12 & ? & 6 \\ ? & ? & 6 & 2 & ? \end{pmatrix}$$

Obviously, an unfair question.

Nonetheless, we believe data "in the wild" is structured... maybe guessable?

Ans: $\begin{pmatrix} 7 & 28 & 42 & 14 & 21 \\ 2 & 8 & 12 & 4 & 6 \\ 1 & 4 & 6 & 2 & 3 \end{pmatrix}$

Highly-structured. Every col = multiple of $\begin{pmatrix} 7 \\ 2 \\ 1 \end{pmatrix}$

Rank-1 matrix: $A = \underbrace{u}_{n\times 1}\underbrace{v}_{d\times 1}{}^{T} \in \mathbb{R}^{n\times d}$

$$A = \begin{pmatrix} A_{:1} & A_{:2} & \cdots & A_{:d} \end{pmatrix} = \begin{pmatrix} (v)_1 u & (v)_2 u & \cdots & (v)_d u \end{pmatrix}$$

More generally, rank-r

$$A = \left(u_1 v_1^T\right) + \left(u_2 v_2^T\right) + \cdots + \left(u_r v_r^T\right)$$

Matrix factorization crash course

1) $$\underbrace{\Lambda}_{\mathrm{diag}(\lambda)} V^T = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_d \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_d^T \end{bmatrix}$$

$$= \begin{bmatrix} \lambda_1 v_1^T \\ \lambda_2 v_2^T \\ \vdots \\ \lambda_d v_d^T \end{bmatrix}$$

2) $$UV^T = \begin{array}{c c} & r \\ n & \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \end{array} \begin{array}{c c} r & \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix} \\ & d \end{array}$$

Intuition: breaking up r-dim. dot products

$$= u_1 v_1^T + \cdots + u_r v_r^T$$

Compactly, rank-r decomposition

$$A = UV^T = \sum_{k \in [r]} u_k v_k^T \in \mathbb{R}^{n \times d}$$

$$U = \begin{pmatrix} u_1 & u_2 & \cdots & u_r \end{pmatrix} \in \mathbb{R}^{n \times r}$$

$$V = \begin{pmatrix} v_1 & v_2 & \cdots & v_r \end{pmatrix} \in \mathbb{R}^{d \times r}$$

Most interesting when $r \ll \min(n, d)$

We can always achieve $r = \min(n, d)$
by choosing $U = I$ or $V = I$

$$I = \text{"identity matrix"} = \begin{pmatrix} 1 & & & 0 \\ & 1 & & \\ & & 1 & \\ 0 & & & \ddots \\ & & & & 1 \end{pmatrix}$$

Who cares? One motivation:

Netflix prize (matrix completion)

$$A = \begin{array}{c} n \\ \text{users} \end{array} \left\{ \underbrace{\begin{pmatrix} & & & & ? \\ ? & ? & ? & \vdots & \\ & ? & ? & & \\ ? & ? & ? & & ? \end{pmatrix}}_{d \text{ movies}} \right.$$

guess unknown movie ratings

Simplified model: everyone agrees

$$v^T = \begin{pmatrix} 9.3 & 9.2 & \cdots & 1.9 & 1.5 \end{pmatrix}$$

e.g. Shawshank Redemption, Godfather, Disaster movie, Superbabies

$$A_{i:} \approx v \cdot \underbrace{[u]_i}_{\text{how much does user } i \text{ appreciate movies?}} \qquad A \approx uv^T$$

More sophisticated model:

$$A \approx \underset{n\times r}{U}\;\underset{d\times r}{V}^T = \left(u_1 v_1^T \right) + \dots + \left(u_r v_r^T \right)$$

$$v_1^T = \Big(\cdots \underset{\text{Silence of the lambs}}{9.5} \cdots \underset{\text{Get out}}{9.3} \cdots \cdots \Big) \quad \text{"horror enjoyer"}$$

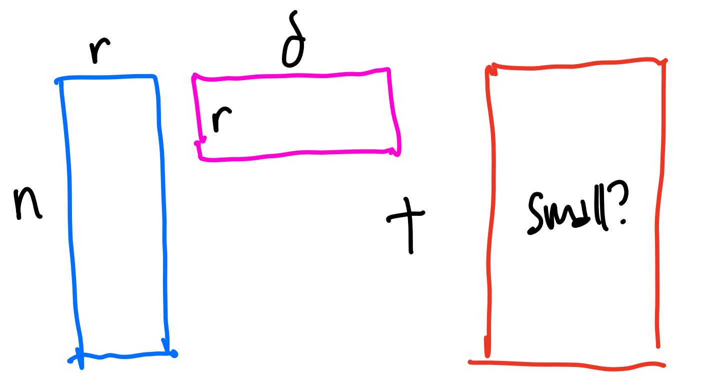
$\vdots$

$$v_r^T = \Big(\cdots\cdots \underset{\text{John Wick}}{8.7} \cdots \underset{\text{Kill Bill}}{9.9} \cdots \Big) \quad \text{"action enjoyer"}$$

$$\underbrace{A_{i:}}_{\substack{\text{ratings of}\\ \text{user } i}} = \sum_{k\in[r]} \underbrace{[u_k]_i}_{\substack{\text{explained}\\ \text{by topic } k}} v_k \in \mathbb{R}^d$$

In real life, not <u>exactly</u> low-rank

$$A = \underbrace{UV^T}_{r \text{ "explanatory factors"}} + \underbrace{N}_{\text{noise}} =$$



"simple" explanation

Sampling error, rounding, etc.

Goal in low-rank approximation (LRA):

Make "unexplained" $A - UV^T$ as small as possible

Other motivations:

- topic modeling

word freqs. in doctype 1   word freqs. in doctype r

$$\begin{array}{c} d \text{ words} \\ n \text{ docs} \begin{pmatrix} \\ \boxed{\text{word counts / doc}} \\ \\ \end{pmatrix} \end{array} \approx \begin{pmatrix} u_1 v_1^T \end{pmatrix} + \dots + \begin{pmatrix} u_r v_r^T \end{pmatrix}$$

- word embeddings (see HW!)
- save on computation, e.g. if $r = O(1)$

$$\underbrace{Ax}_{\text{takes } O(nd) \text{ time}} \approx U \underbrace{V^T x}_{\text{takes } O(d) \text{ time}} = \underbrace{U(V^T x)}_{\text{takes } O(n) \text{ time}}$$

Singular value decomposition (Part VI, Section 4.2)

We have a complete characterization of

$$\underset{\substack{U \in \mathbb{R}^{n \times r} \\ V \in \mathbb{R}^{d \times r}}}{\operatorname{argmin}} \underbrace{\| A - UV^T \|}_{\substack{\text{"unitarily invariant"} \\ \text{norm. incl. basically all} \\ \text{common size bounds...} \\ \text{Frobenius, operator, nuclear norms}}}$$

Detour to SVD needed.

Say that $U \in \mathbb{R}^{d \times d}$ is orthonormal if

$$U^T U = I = \begin{pmatrix} 1 & & 0 \\ & 1 & \\ & & \ddots \\ 0 & & 1 \end{pmatrix}$$

Interpretation: Let $U = \begin{pmatrix} u_1 & u_2 & \cdots & u_d \end{pmatrix}$

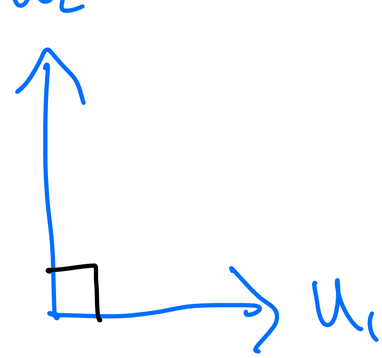
Then, $\forall \, (i,j) \in [d] \times [d]$

$$[U^T U]_{ij} = \underbrace{u_i^T u_j}_{\cos(\theta_{ij})} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

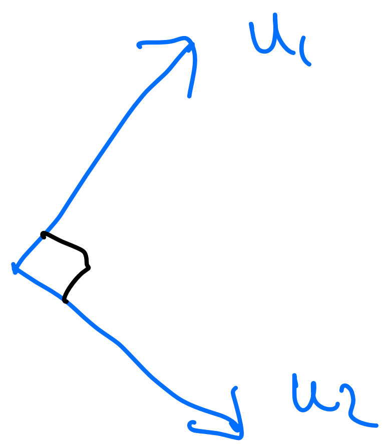
Recall $u^T u = \sum u_i^2 = \underbrace{\| u \|_2}_{\text{length of } u}{}^2$ (Pythagoras)

Hence columns of U
- unit length
- pairwise perpendicular

e.g.



"Standard basis" $U = I$



Orthonormal basis

Also possible for rectangular "tall" = orthonormal

$U \in \mathbb{R}^{n \times d}$, $n \geq d$ $\quad U^T U = I$

$$U = \begin{pmatrix} \underbrace{u_1 \quad u_2 \quad \cdots \quad u_d}_{\substack{\text{unit length,} \\ \text{pairwise perp}}} \end{pmatrix}$$

Subset of full basis $\{u_i\}_{i \in [n]}$

e.g.

$$U = \begin{array}{c} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \\ \begin{array}{ccc} e_1 & e_2 & e_4 \end{array} \end{array}$$

is orthonormal, Span(U) is "subspace" of $\mathbb{R}^5$ with dimension 3

$$Ux = x_1 e_1 + x_2 e_2 + x_4 e_4$$

(no mass on e_3, e_5 allowed)

SVD: <u>All</u> matrices $A \in \mathbb{R}^{n \times d}$, $n \geq d$

can be decomposed as

$$A = U \Sigma V^T = \sum_{i \in [d]} \sigma_i u_i v_i^T$$

$$\left.\begin{array}{l} U = \begin{pmatrix} u_1 & \cdots & u_d \end{pmatrix} \in \mathbb{R}^{n \times d} \\[2em] V = \begin{pmatrix} v_1 & \cdots & v_d \end{pmatrix} \in \mathbb{R}^{d \times d} \end{array}\right\} \text{orthonormal matrices}$$

$$\Sigma = \underbrace{\begin{pmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_d \end{pmatrix}}_{\#\,\text{nonzero} = \operatorname{rank}(A) = \dim(\operatorname{span}(A))} \in \mathbb{R}^{d \times d}_{\geq 0}$$

nonneg. diagonal matrix

Interpretation: SVD sends $x \to Ax$

v_i = "input world" to $\sigma_i u_i$ = "output world"

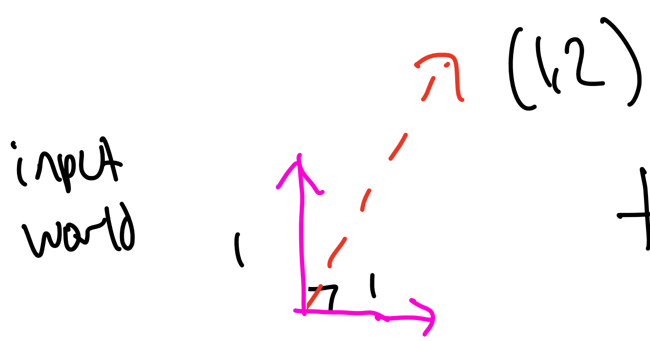
$v_i \in \mathbb{R}^d$, $u_i \in \mathbb{R}^n$

$$x = \sum_{i \in [d]} c_i v_i = V c \quad (\text{coeffs. } c \in \mathbb{R}^d)$$

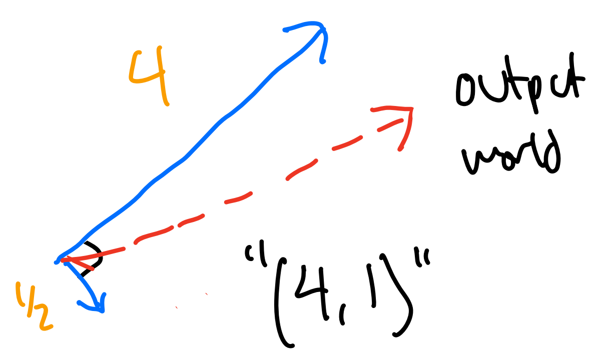
$$\to \quad Ax = U \Sigma \underbrace{V^T V}_{= I} c = \sum_{i \in [d]} \sigma_i c_i u_i$$

e.g.

$$A = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} 4 & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$



to



Eckart-Young-Mirsky:

For any $A \in \mathbb{R}^{n \times d}$, $n \geq d$, unitarily-invariant $\|\cdot\|$

$$\operatorname*{argmin}_{\text{rank-}r\ M} \| A - M \|$$

achieved by $M = \sum_{k \in [r]} \sigma_k u_k v_k^T$

where $A = U \Sigma V^T$ (SVD)

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d$$

AKA, optimal low-rank approx.

= do SVD, keep top r components,

all else $\Rightarrow 0$.

Special case: Symmetric matrix $M \in \mathbb{R}^{d \times d}$

$$M = \underbrace{U \Lambda U^T}_{\text{"eigendecomposition"}}, \text{ but } \Lambda \in \mathbb{R}^{d}$$

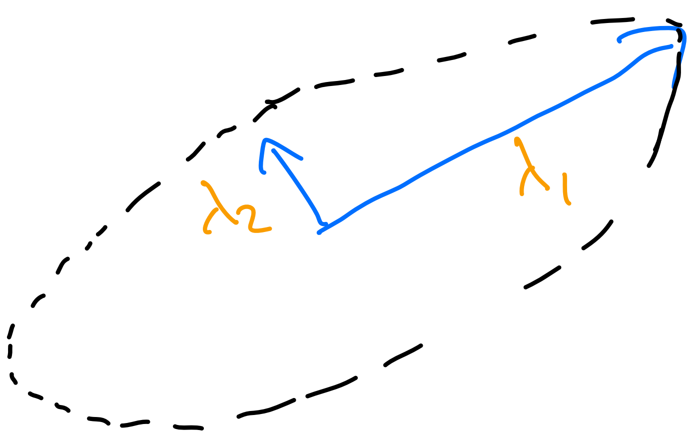
diagonal, could be neg.

Even more special: if $\Lambda \in \mathbb{R}^{d \times d}_{\geq 0}$ already

M is "positive semidefinite" (PSD)

(Definition: symmetric matrix, all eigvals $\lambda_i \geq 0$)

Input world = output world!



$$M u_i = \lambda_i u_i$$

geometric interpretation of eigvecs/eigvals

"every PSD matrix is an ellipse"

If $A = U \Sigma V^T$, we have

$$A^T A = V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T$$

$$A A^T = U \Sigma V^T V \Sigma U^T = U \Sigma^2 U^T$$

Both PSD, eigvectors give SVD of A.

Enough to give algos to recover:

top eigvec of PSD matrix M (PCA)

Power method (Part VI, Section 4.3)

Let PSD $M \in \mathbb{R}^{d \times d}$

$$M = U \underbrace{\Lambda}_{\mathrm{diag}(\lambda)} U^T = \sum_{i \in [d]} \lambda_i u_i u_i^T$$

$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$

Goal: recover u_1 to high accuracy.

Idea: suppose $U = I$, $\lambda_1 \geq 2000\, d^2 \lambda_2$

Then $\underset{\substack{\text{random} \\ \text{normal vector}}}{M g} \approx$ multiple of e_1

In fact, $Ug \sim g$ (still works for $U \neq I$)

But $\lambda_1 \geq 2000\, d^2 \lambda_2$ really strong...

What if only $\lambda_1 \geq 1.1 \lambda_2$?

No problem: for $p = O(\log d)$

$$M^p g = U \Lambda \cancel{U^T U} \Lambda \cancel{U^T U} \Lambda U^T \cdots$$

$$= U \underbrace{\Lambda^p}_{\text{gap now } 1.1^p \geq 2000\, d^2} U^T g$$

normalize $M^p g$

Runtime: $O(d^2 \log(d))$ gives $\cos\theta(u, u_1) \geq 0.99$ w.p. ≥ 0.99