LOW - RANK APPROXIMATION:

Given some matrix

$$X \qquad n\left\{ \overbrace{\boxed{\begin{matrix} \quad & \quad & \quad \\ \quad & \quad & \quad \\ \quad & \quad & \quad \end{matrix}}}^{d} \right. \cong$$

Approximate X in a "simpler" way

1. What does Simple mean?
2. What does "approximate" mean?

1. SIMPLE $\equiv$ "LOW" - RANK

$\text{RANK}(X) = \dim(\text{span of the columns}) = \dim(\text{span of the rows})$

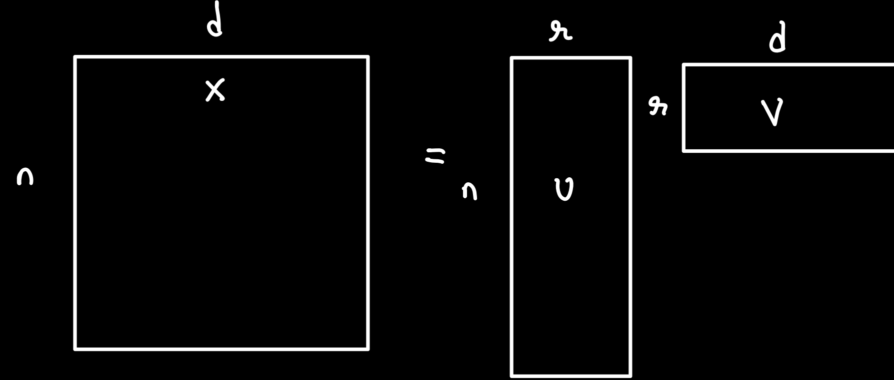
$$\text{rank}\left(\begin{bmatrix} u & u & \cdots & u \end{bmatrix} \right) = 1$$

$$\text{rank}\left(\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & 1 \end{bmatrix} \right) = n$$

→ Why rank as a notion of simplicity?

→ Probabilistic / Statistical models

→



$$X = U \cdot V$$

$$\underset{n \times r}{\downarrow} \quad \underset{r \times d}{\downarrow}$$

$$\#\text{ parameters needed to write } U, V \equiv n \cdot r + r \cdot d$$

$$= (n+d)r.$$

2. QUANTIFYING APPROXIMATION :

Squared Error (RMSE) :

$$\left(\sum_{i=1}^{n} \sum_{j=1}^{d} (x_{ij} - \tilde{x}_{ij})^2 \right)^{1/2}$$

$$= \|X - \tilde{X}\|_F$$

For any matrix $A \in \mathbb{R}^{n \times d}$

$$\|A\|_F = \left(\sum_{i,j} A_{ij}^2 \right)^{1/2}$$

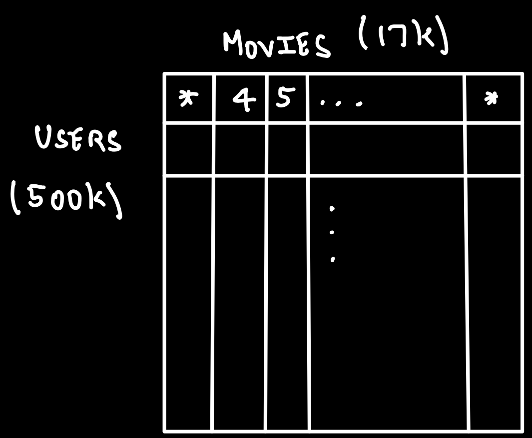
"FROBENIUS NORM"

LOW-RANK APPROXIMATION PROBLEM:

INPUT: Given $X \in \mathbb{R}^{n \times d}$, k

OUTPUT: $\underset{\operatorname{rank}(\tilde{X}) \le k}{\arg\min} \|X - \tilde{X}\|_F$

NETFLIX CHALLENGE:



$* \longrightarrow$ not rated

$\approx$ 10 Million entries seen

$\approx$ 1% of all possible entries.

GOAL: Guess missing entries

SCORE: Average RMSE on holdout entries.

CHALLENGE: First team to do 10% better than their baseline gets \$1 Million.

Step 1: Preprocess the data: center the scores / normalizing.

Step 2: Put 0 for the missing entries to get a matrix X.

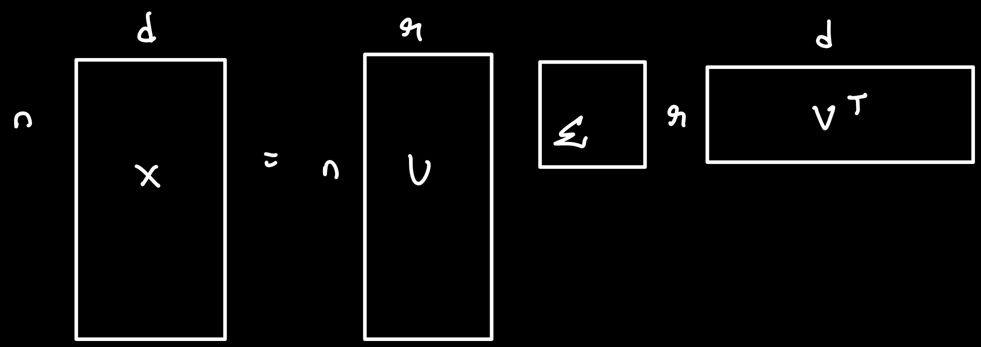
Step 3: Find the best low-rank approximation $\tilde{X}$ to X (for some k)

Step 4: Predict according to $\tilde{X}$.

Experiments: If you run this with $k = 30$, you beat their baseline by 4%.

SINGULAR VALUE DECOMPOSITION :

THEOREM : Any matrix X



1. Columns of U are orthonormal $(U^T U = I_r)$

2. Columns of V are orthonormal $(V^T V = I_r)$

3. Σ is a diagonal matrix with non-negative entries.

(Remark : Some take Σ to be a $d \times d$ matrix with zeroes)

Examples :

1.
$$\begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 4 & \\ & & & 5 \end{bmatrix} = \underset{U}{I} \; \underset{\Sigma}{\begin{bmatrix} 1 & & & \\ & 2 & & \\ & & 4 & \\ & & & 5 \end{bmatrix}} \; \underset{V^T}{I}$$

2. $$\begin{bmatrix} 1 & 0 \\ 0 & -2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{V^T}$$

3. Orthonormal matrix

x : has columns that are orthonormal

$$x = x \cdot I \cdot I$$

TWO MAGICAL PROPERTIES OF SVD :

$$\begin{bmatrix} \\ & X & \\ \\ \end{bmatrix} = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \Sigma_{11} & & & \\ & \Sigma_{22} & & \\ & & \ddots & \\ & & & \Sigma_{nn} \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix}}_{V^T}$$

$$\Sigma_{11} \geq \Sigma_{22} \geq \Sigma_{33} \geq \cdots \geq \Sigma_{nn}$$

THEOREM 1: v_i is the i^{TH} - Right Singular vector of x.

THEOREM 2: (ECKART - YOUNG - MINSKY THEOREM 1936)

$$X_k = \underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix}}_{n \times k} \underbrace{\begin{bmatrix} \Sigma_{11} & & & \\ & \Sigma_{22} & & \\ & & \ddots & \\ & & & \Sigma_{kk} \end{bmatrix}}_{k \times k} \underbrace{\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_k^T \end{bmatrix}}_{k \times d}$$

is a solution to best rank k approximation.

HOW DO YOU GET THE SVD ?

Some more properties of SVD

$$X = U \Sigma V^T$$

$$(\Sigma_{11} \geq \Sigma_{22} \geq \cdots \geq \Sigma_{rr})$$

1. $X = U \Sigma V^T$

$$X = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \begin{bmatrix} \Sigma \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix}$$

$$= \underbrace{\Sigma_{11}}_{\sigma_1} \begin{bmatrix} u_1 \end{bmatrix} \begin{bmatrix} v_1^T \end{bmatrix} + \underbrace{\Sigma_{22}}_{\sigma_2} \begin{bmatrix} u_2 \end{bmatrix} \begin{bmatrix} v_2^T \end{bmatrix}$$

$$+ \cdots + \underbrace{\Sigma_{rr}}_{\sigma_r} \begin{bmatrix} u_r \end{bmatrix} \begin{bmatrix} v_r^T \end{bmatrix}$$

$$X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

2. $X \cdot v_i = \sigma_i \cdot u_i$

PROOF:

$$X \cdot v_i = \left(\sum_{l=1}^{r} \sigma_l \cdot u_l \cdot v_l^T \right) \cdot v_i$$

$$= \sum_{l=1}^{r} \sigma_l u_l \underbrace{\langle v_l, v_i \rangle}_{\substack{0 \text{ if } l \neq i \\ 1 \text{ if } l = i}}$$

$$= \sigma_i u_i$$

3. $u_i^T X = \sigma_i v_i^T$

4. $\sigma_i = \| X \cdot v_i \|$ (as $X \cdot v_i = \sigma_i u_i$ and u_i is a unit vector)

5. $u_i = \dfrac{X \cdot v_i}{\| X \cdot v_i \|}$

if we know v's, we can infer σ and u's.

THEOREM: Can compute full SVD directly in time

$$O(n \cdot d^2).$$

$$\underset{500k}{n} \qquad \underset{17k}{d}$$

$n \cdot d \equiv 8 GB$ in memory!

Question: What if I want to compute just the top $k = 30$ singular vectors?

POWER ITERATION:

How can we compute the first right singular vector (v_1)?

Idea: $X = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_n u_n v_n^T$

$$= U \Sigma V^T$$

$$y = X^T X$$

$$= (V \Sigma U^T)(U \Sigma V^T)$$

$$= V \Sigma (U^T U) \Sigma V^T$$

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

$$y^2 = (V \Sigma^2 V^T)(V \Sigma^2 V^T)$$

$$= V \Sigma^4 V^T$$

$$y^3 = V \Sigma^6 V^T$$

$$\boxed{y^\ell = V \Sigma^{2\ell} V^T}$$

$$= \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \sigma_1^{2\ell} & & & 0 \\ & \sigma_2^{2\ell} & & \\ & & \ddots & \\ 0 & & & \sigma_n^{2\ell} \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ v_n^T \end{bmatrix}$$

$$y^\ell = \sigma_1^{2\ell} v_1 \cdot v_1^T + \sigma_2^{2\ell} v_2 \cdot v_2^T + \dots + \sigma_n^{2\ell} v_n \cdot v_n^T$$

EXAMPLE:

Suppose $\sigma_1 = 1$, $\sigma_2 = 1/2$, $\sigma_3 \leq 1/2$, ...

$$y^\ell = 1 \cdot v_1 \cdot v_1^T + \left(\frac{1}{2}\right)^{2\ell} v_2 \cdot v_2^T + \sigma_3^{2\ell} \cdot v_3 v_3^T + \dots$$

$$= v_1 v_1^T + \underbrace{\sum_{i=2}^{n} \sigma_i^{2\ell} v_i \cdot v_i^T}_{\to 0}$$

As $\ell \to \infty$

$$v_1 v_1^T = \begin{bmatrix} v_1 \cdot (v_{1,1}) & v_1 (v_{1,2}) \cdots \end{bmatrix}$$

↳ Look at a single column to get v_1!

IDEA:

1. Compute y^t for a large $t = 100$

2. Compute first column of y^t

$$\bar{v} = \boxed{y^t} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

3. Output $\dfrac{\bar{v}}{\|\bar{v}\|}$

$$\bar{v} = \underbrace{(y \cdot (y \cdot (y \cdots (y \cdot}_{100 \text{ times}} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}))))$$

POWER ITERATION :

- $\bar{v}_0 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

- for $l = 1, \dots t$

$$\bar{v}_l = X^T \cdot (X \cdot \bar{v}_{l-1})$$

- Output $\dfrac{\bar{v}_t}{\|\bar{v}_t\|}$ $\longrightarrow$ Gives first RSV.