LECTURE 9

(See end of transcript of lecture 8 for start of this lecture).

Last class
→ Analyzed MWM
→ Defined Boosting problem

Today
→ Finish boosting (see lecture 8 scribble)
→ Unsupervised learning
- identifying structure in data.

Assignment 2 is due on Wednesday.
I have office hours from 3-4 today.

→ We will view PCA as an optimization problem
"Best-Fit-Subspace" problem
→ "Maximizing Variance problem".
→ How to find principal components? Greedy works!

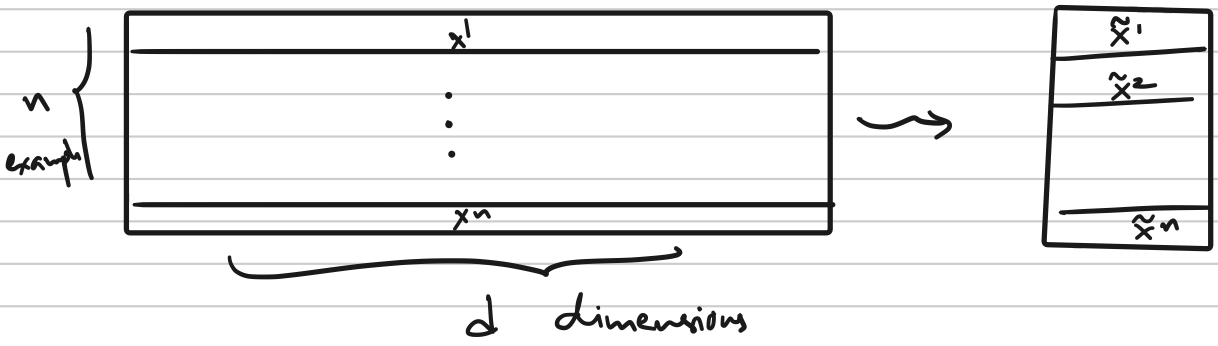
Low-Rank Matrix Approximation.

→ Equivalence to PCA
→ Computing PCA by Matrix algebra.

LECTURE 10

Last class	Today
– Boosting	– What is PCA?
– Why Principal Components?	– Toward computing PC's.
– Demo PCA	

ASSIGNMENT 2 IS DUE TODAY @ 9:59 PM.



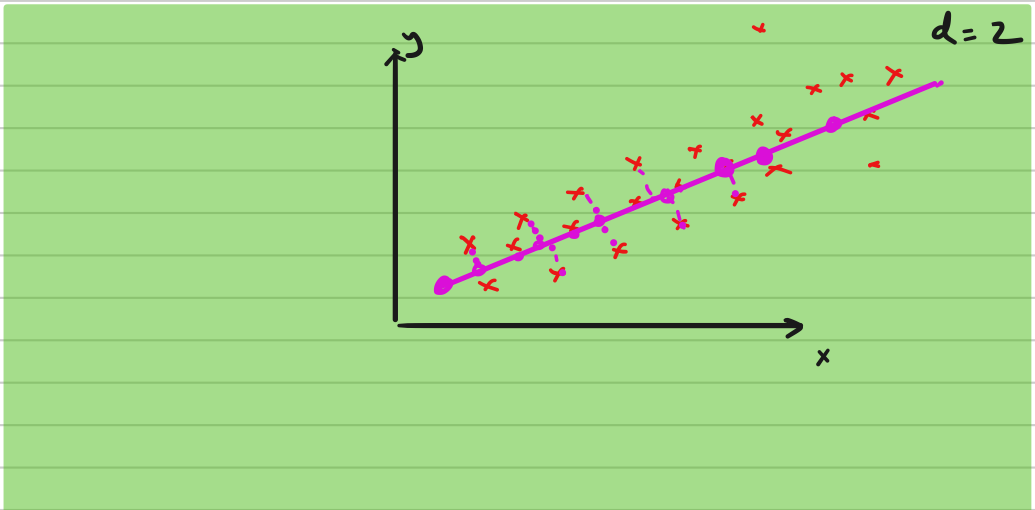
$n \ll d$.

Idea: We want to "represent"

$x \longrightarrow \tilde{x}$

in d dimensions — in low dimensions

is a "linear map"?



BEST-FIT LINE PROBLEM

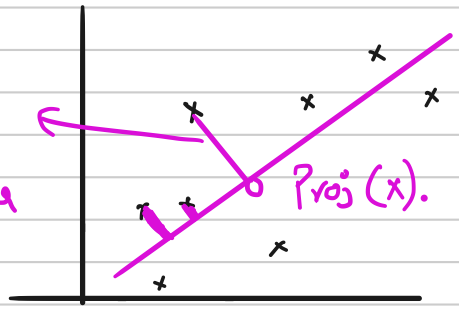
INPUT: $x^1, x^2, \ldots, x^n \in \mathbb{R}^d$.

OUTPUT: Find the line that is "closest" to the dataset.

→ Defining Closest:

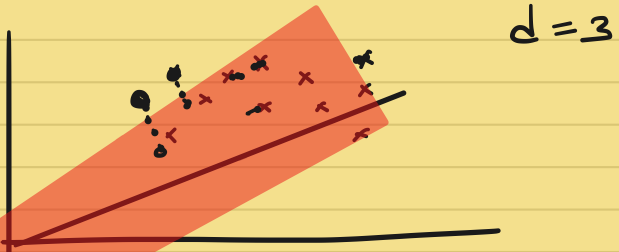
Minimize aggregate distance of points to the line.

Measuring distance of x to a line as length of the perpendicular to the line $\equiv \| x - \text{Proj}(x) \|$.



$$\text{Avgdistance}(L) = \sum_{i=1}^{n} \| x^i - \text{Proj}_L(x^i) \|_2^2$$

$\text{Proj}_L(x) \equiv$ closest point to x on the line L.



General Best-Fit-Subspace Problem

INPUT: $X = \{x^1, x^2, \ldots, x^n\}$; dimension of subspace k.

For any k-dimensional subspace $S \subseteq \mathbb{R}^d$,

$$\mathbf{ERR}(S; X) = \sum_{i=1}^{n} \| x^i - \text{Proj}_S(x^i) \|_2^2 .$$

OUTPUT: Find a k-dim. subspace that minimizes $\text{ERR}(S;X)$.

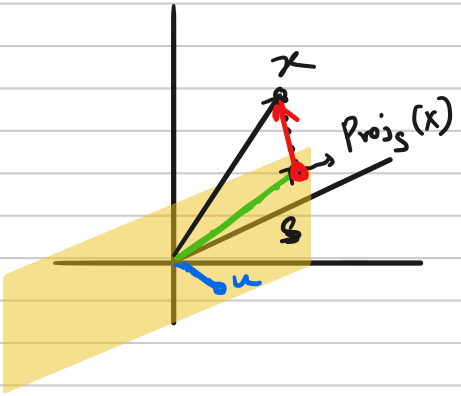
↓

– Return an orthonormal basis for the subspace.

Next Goal: How to solve $k=1$.

LINEAR ALGEBRA FACTS.

① $\text{Proj}_S(x) = \underset{y \in S}{\arg\min} \|x - y\|_2$.



② $\langle x - \text{Proj}_S(x), \text{Proj}_S(x) \rangle = 0$

(angle is 90°)

③ $\|x\|^2 = \|x - \text{Proj}_S(x)\|^2 + \|\text{Proj}_S(x)\|^2$.

④ For all points $u \in S$

$$\langle u, x - \text{Proj}_S(x) \rangle = 0.$$

INPUT $X = \{x^1, \ldots, x^n\} \subseteq \mathbb{R}^d$. $k = 1$.

Goal: Find 1-dimensional subspace S to minimize

$$\text{ERR}(S; X) = \sum_{i=1}^{n} \| x^i - \text{Proj}_S(x^i) \|_2^2.$$

$$\overset{\text{by } ③}{=} \sum_{i=1}^{n} \left(\|x^i\|^2 - \|\text{Proj}_S(x^i)\|_2^2 \right)$$

$$= \sum_{i=1}^{n} \|x^i\|^2 - \sum_{i=1}^{n} \|\text{Proj}_S(x^i)\|_2^2.$$

So looking for an S to minimize $\text{ERR}(S; X)$ is the same as looking for an S to maximize

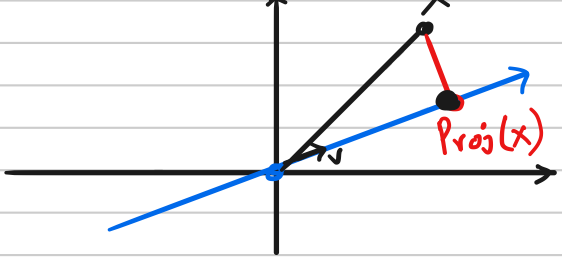
$$\text{Var}(S; X) = \sum_{i=1}^{n} \|\text{Proj}_S(x^i)\|_2^2.$$

Minimizing ERR(S) over k-dim. subspaces $\equiv$ Maximizing Var(S) over k-dim. subspaces

---

Focusing on $k=1$ Case:

$S \equiv \text{span}\{v\}$



v is a unit vector

$$\text{Proj}_S(x) = \langle v, x \rangle \cdot v .$$

$$\text{Proj}_S(x) = \frac{\langle v, x \rangle}{\|v\|^2} \cdot v .$$

So for $k=1$: if $S = \text{span}\{v\}$.

$$\text{Var}(S) = \sum_{i=1}^{n} \|\text{Proj}_S(x^i)\|^2$$

$$= \sum_{i=1}^{n} \left\| \frac{\langle v, x^i \rangle}{\|v\|^2} \cdot v \right\|^2$$

$$= \sum_{i=1}^{n} \frac{\langle v, x^i \rangle^2}{\|v\|^2} .$$

$$= \frac{1}{\|v\|^2} \cdot \sum_{i=1}^{n} \langle x^i, v \rangle^2 . \equiv \frac{\|X \cdot v\|^2}{\|v\|^2} .$$

$$X \left\{ \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ \vdots \end{bmatrix} \right. \quad v \quad = \quad \begin{bmatrix} \langle x^1, v \rangle \\ \langle x^2, v \rangle \\ \vdots \end{bmatrix}$$

$S = \text{span}\{v\}$.

$$\text{VAR}(S; X) = \frac{\|X \cdot v\|^2}{\|v\|^2}.$$

The best-fit-Line problem is equivalent to

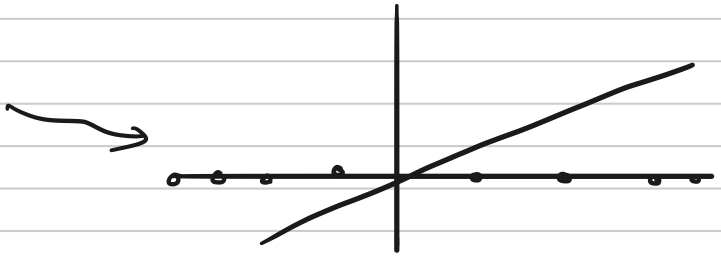
$$\arg\max_{v \neq 0} \frac{\|X \cdot v\|^2}{\|v\|^2}$$ unit norm → • FIRST RIGHT SINGULAR VECTOR OF X.
• FIRST PRINCIPAL COMPONENT OF X.

Example 1:

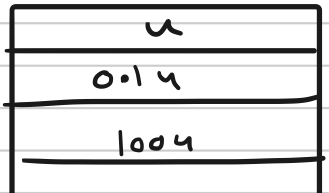
$X \equiv$

10	0 . . . 0
9	0 . 0
1000	0 . . 0
1	0 . . . 0

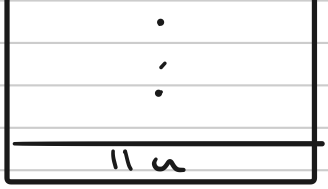
First PC of $X \equiv (1, 0, \ldots, 0)$



Example 2:



First PC of $X \equiv u$.



Summary:

Solving for Best-Fit-Line $\equiv$ Solving for max Var $\equiv$ Solving for $\arg\max_{v:\|v\|=1} \|X \cdot v\|^2$ $\rightsquigarrow$ (First PC).

How about $k = 2$?

Recall: $\arg\max_{S:\, \dim(S)=2} \sum_{i=1}^{n} \|\text{Proj}_S(x^i)\|^2$.

Idea: To use a "greedy" approach.

Find first PC and then find first PC of the "left-over".

Idea:

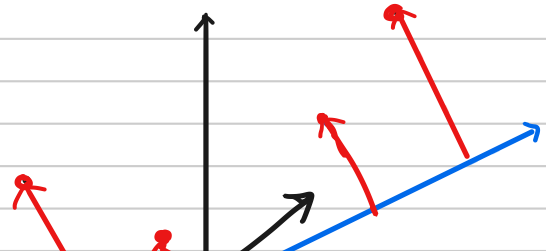
① Find First PC $\rightarrow v_1$

② Replace $\bar{x}^i = x^i - \text{Proj}_{\langle v_1 \rangle}(x^i)$

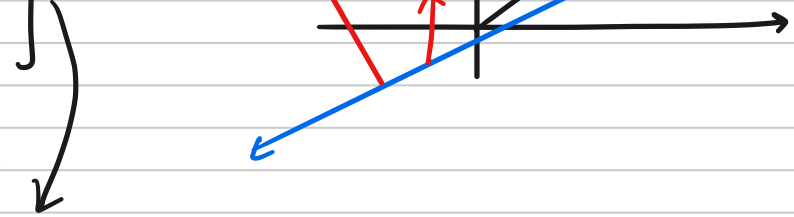
③ Find first PC of the "new" dataset.

Idea 2:

$\arg\max \|X \cdot v\|^2$



$v: \|v\| = 1$
and $v \perp v_1$
(v is perpendicular to first PC).



Second right singular vector of X
or Second principal component of X.

If $v_1, \ldots, v_{i-1}$ are the first, second third, ..., $(i-1)^{th}$ PCs of X, then

$$\arg\max_{v:\|v\|=1,\ v \perp v_1, v \perp v_2, \ldots, v \perp v_{i-1}} \|X \cdot v\|^2$$

$\}$ i^{th} PC of X or i^{th} Right singular vector of X.

Theorem: The Span of first k right singular vectors minimizes $ERR(S; X)$.
($\equiv$ maximizes $VAR(S; X)$).

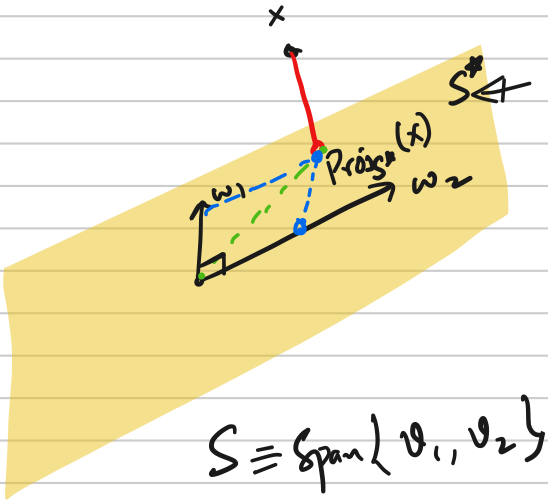
Remark: For $ERR_1(S; X) = \sum_{i=1}^{n} \|X^i - Proj_S(x^i)\|$,
finding best-fit-line, & then best-fit-line of left-overs does not work.

Thm: Span of first two right singular vectors

maximizes $\operatorname{Var}(S;X)$.
$\dim(S)=2$

Proof: $S^* = \arg\max_{\dim(S)=2} \operatorname{Var}(S;X)$

$S^* = \operatorname{Span}\{w_1, w_2\}$ → orthonormal basis for S^*.



$$\begin{cases} v_1 \equiv \arg\max_{\|v\|=1} \|X\cdot v\| \\ v_2 \equiv \arg\max_{\substack{\|v\|=1 \\ v\perp v_1}} \|X\cdot v\| \end{cases}$$

$S \equiv \operatorname{Span}\{v_1, v_2\}$ are $\perp$ to each other.

Claim:

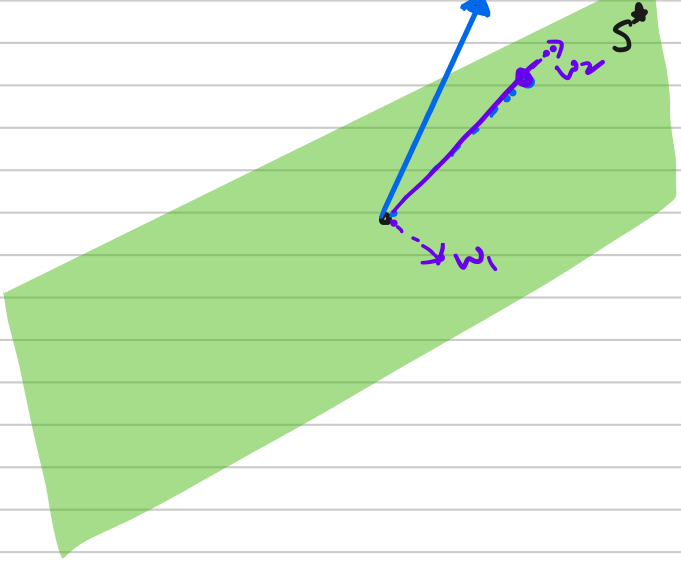
1. $\|\text{Proj}_{S^*}(x)\|^2 = \langle x, w_1\rangle^2 + \langle x, w_2\rangle^2$
2. $\|\text{Proj}_{S}(x)\|^2 = \langle x, v_1\rangle^2 + \langle x, v_2\rangle^2$

$$\Rightarrow \quad \operatorname{Var}(S^*;X) = \|X\cdot w_1\|^2 + \|X\cdot w_2\|^2$$
$$\operatorname{Var}(S;X) = \|X\cdot v_1\|^2 + \|X\cdot v_2\|^2.$$

At the very least, by definition

$$\|X\cdot w_1\|^2 \le \|X\cdot v_1\|^2.$$

Claim: I can always pick an orthonormal basis $\{w_1, w_2\}$ for S^* where $w_2 \perp v_1$.



$\Rightarrow$ For this basis: $\text{Var}(S^*; X) = \|X \cdot w_1\|^2 + \|X \cdot w_2\|^2$

$\text{Var}(S; X) = \|X \cdot v_1\|^2 + \|X \cdot v_2\|^2$

$\Rightarrow \text{Var}(S^*; X) \leq \text{Var}(S; X)$

$\Rightarrow$ S maximizes the variance.

LECTURE 11 05/02

LAST CLASS

→ Defining PCA

→ min Error ≡ max Variance

→ Singular vectors

→ Greedy strategy will work

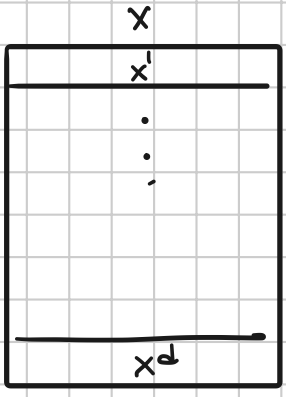
Today

→ Low-Rank Approximation

→ SVD: The Swiss-Army Knife of Linear Algebra

→ Computing top singular vector.

Recap



Rows ≡ individual points

$$ERR(S;X) = \sum_i \| x^i - Proj_S(x^i) \|_2^2$$

$$VAR(S;X) = \sum_i \| Proj_S(x^i) \|_2^2.$$

Best-Fit Subspace

$$\min_{S:\dim(S)=k} ERR(S) \equiv \max_{S:\dim(S)=k} VAR(S)$$

Right Singular Vector

First RSV: $\operatorname{argmax}_{\|v\|=1} \|Xv\|$

k^{th} RSV:

$$v_k = \operatorname*{argmax}_{v \perp v_1, v \perp v_2, \ldots, v \perp v_{k-1}} \|Xv\|$$

Thm: Span $\{v_1, \ldots, v_k\}$ (the top k right SV's) gives a solution to the Best-Fit-Subspace problem.

LOW-RANK APPROXIMATION

Given some matrix

X ($n \times d$)

$\approx$ Approximate X in a "simpler" way.

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

SIMPLE $\equiv$ "LOW" RANK

RANK(X) = dim(Span of the columns) = dim(Span of the rows).

$\text{rank}\begin{pmatrix} u & u & \cdots & u \\ | & | & & | \end{pmatrix} = 1$.

$\text{rank}\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \\ 0 & & & 1 \end{pmatrix} = n$.

→ Why rank as a notion of simplicity?

→ Probabilistic/statistical models

→ X ($n \times d$) $= U$ ($n \times$) Y ($\times d$)

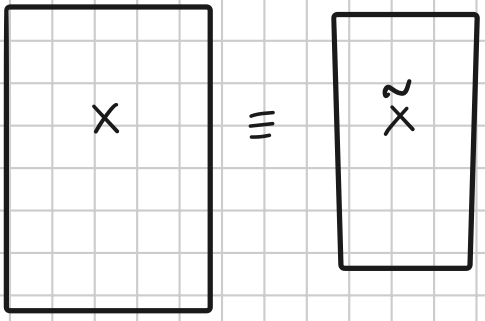
$X = U \cdot V$ (U: $n \times r$, V: $r \times d$)

parameters needed to work $U, V \equiv n \cdot r + r \cdot d$

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

---

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: $\underset{\text{rank}(\tilde{X}) \le k}{\text{argmin}} \| X - \tilde{X} \|_F$

NETFLIX CHALLENGE

$\approx$ 100 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./Normalize the data

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

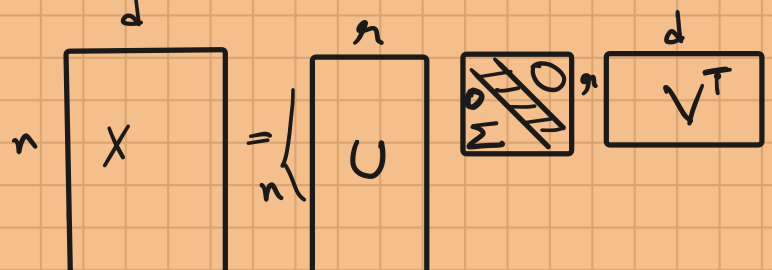
Step 3: Find the best low-rank approximation $\tilde{X}$ to X (for some k).

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

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

SINGULAR VALUE DECOMPOSITION.

Thm: Any matrix X



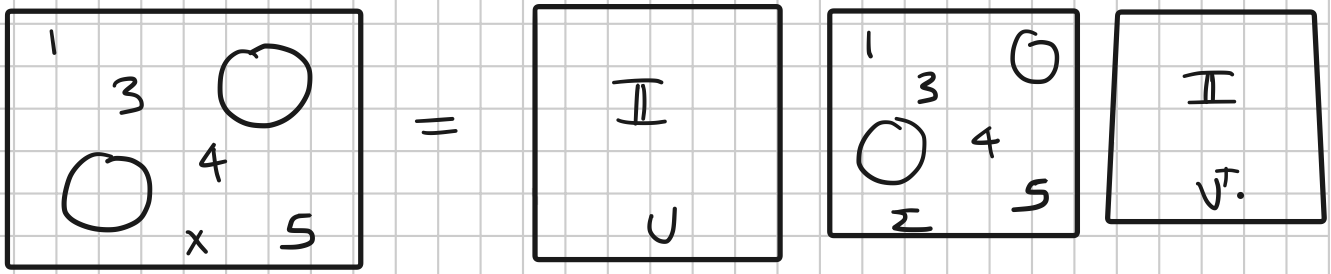
① Columns of U are orthonormal. $U^T U = I_n$

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

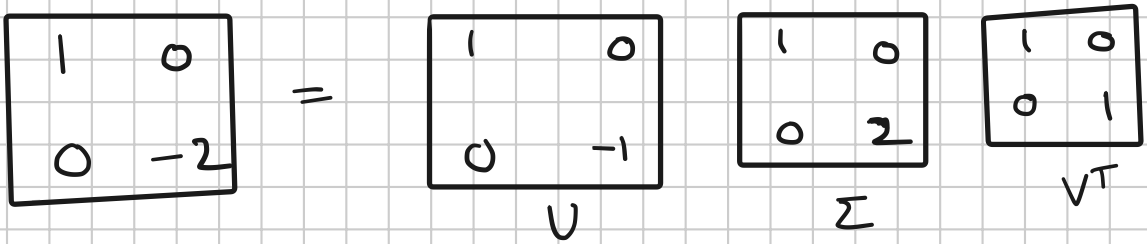
③ Σ is a diagonal matrix with non-negative entries.

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

Ex: ①



②

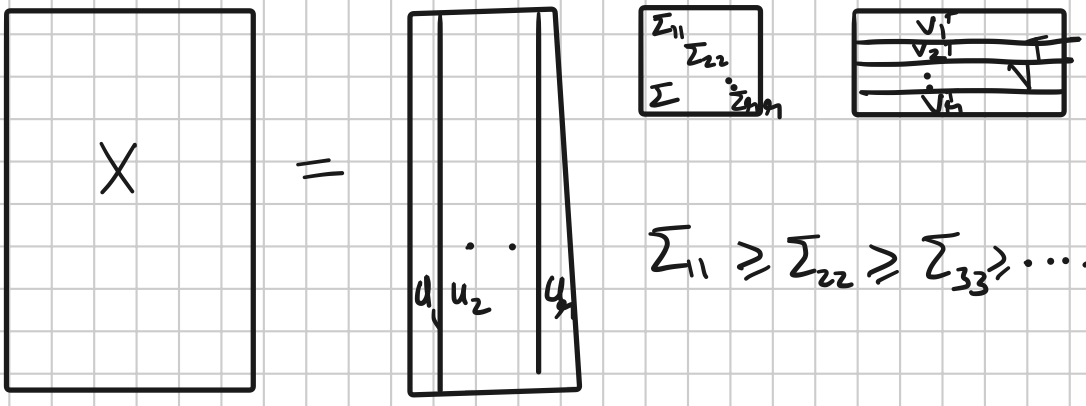


③ Orthonormal matrix

X: has columns that are orthonormal

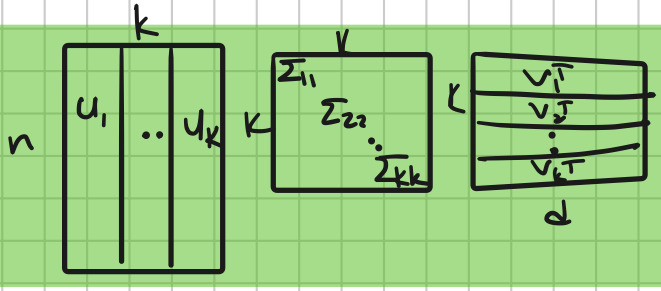
$X = X \cdot \text{I} \cdot \text{I} \cdot$

TWO MAGICAL PROPERTIES OF SVD



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

Thm: V_i is the i^{th} Right Singular vector of X.

Thm: $X_k \equiv$ 

is a solution to best rank k approximation!

(Eckart-Young-Mirsky Theorem 1936).

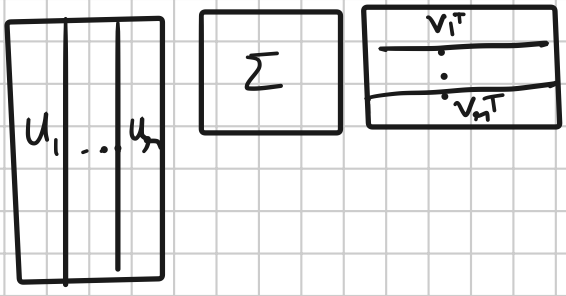
How do you get the SVD?

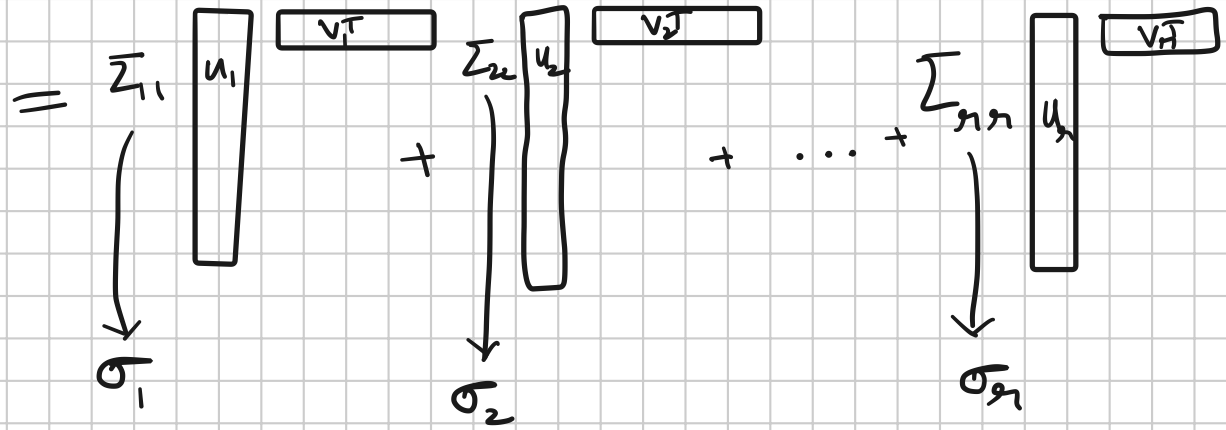
Some more properties of SVD.

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

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

① $X = U \Sigma V^T$

$X =$ 

$=$ 

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

② $X \cdot v_i = \sigma_i \cdot u_i$

$$\left(X \cdot v_i = \left(\sum_{\ell=1}^{n} \sigma_\ell \cdot u_\ell \cdot v_\ell^T \right) \cdot v_i \right.$$

$$= \sum_{\ell=1}^{n} \sigma_\ell \cdot u_\ell \cdot \underbrace{\langle v_\ell, v_i \rangle}_{\begin{array}{l} 0 \text{ if } \ell \neq i \\ 1 \text{ if } \ell = i \end{array}}$$

$$\left. = \sigma_i u_i \cdot \right)$$

③ $u_i^T X = \sigma_i v_i^T$.

④ $\sigma_i = \| X \cdot v_i \|$.

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

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

$\Downarrow$

If we know v's, we can infer σ_i and u's.

Thm: Can compute full SVD directly in time $O(n \cdot d^2)$.

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

500k 17k

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

$= U \Sigma V^T$.

$Y = X^T \cdot X$

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

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

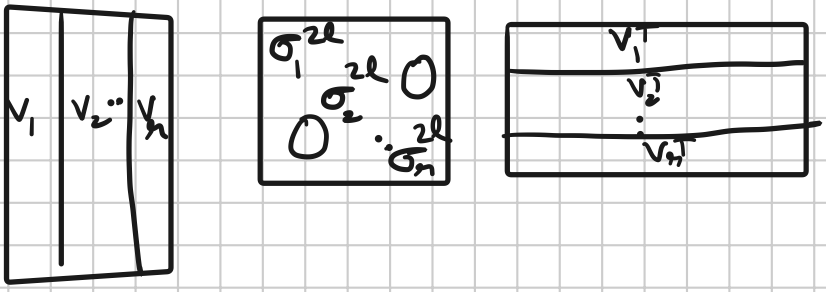
$= V \Sigma^2 V^T$.

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

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

$$Y^3 = V \cdot \Sigma^6 \cdot V^T \cdot$$

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

$=$ 

$$Y^\ell = \sigma_1^{2\ell} V_1 \cdot V_1^T + \sigma_2^{2\ell} V_2 \cdot V_2^T + \cdots + \sigma_n^{2\ell} V_n \cdot V_n^T \cdot$$

Example: Suppose $\sigma_1 = 1$, $\sigma_2 = \frac{1}{2}$, $\sigma_3 \le \frac{1}{2}$, $\cdots$

$$Y^\ell = 1 \cdot V_1 \cdot V_1^T + \left(\frac{1}{2}\right)^{2\ell} V_2 V_2^T + \sigma_3^{2\ell} \cdot V_3 V_3^T + \cdots +$$

$$= V_1 V_1^T + \sum_{i=2}^{n} \sigma_i^{2\ell} \cdot V_i V_i^T$$

As $\ell \to \infty$, $\to 0$.

$$V_1 V_1^T = \left[\, V_1 \cdot (V_{1,1}) \;\middle|\; V_1 (V_{1,2}) \cdots \right]$$

$\Rightarrow$ a single column

Look ... to get v_1!

Idea:

① Compute Y^t for a large $t \approx 100$

② Compute first column of Y^t.

$$\bar{v} = Y^t \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$Y^t = Y \cdot Y \cdot Y \cdots Y$ (t times)

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

$$\bar{v} = \underbrace{\left(Y \cdot \left(Y \cdot \left(Y \cdot \cdots \left(Y \cdot \left(Y \left(Y \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}\right)\right)\right)\right)\right)\right)}_{100 \text{ times}}$$

POWER ITERATION

- $\bar{v}_0 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$
- For $\ell = 1, \ldots, t \sim Y$
  $\bar{v}_\ell = X^T \cdot (X \cdot \bar{v}_{\ell-1})$
- Output $\frac{\bar{v}_t}{\|\bar{v}_t\|}$

$\equiv$ Give first RSV.

LECTURE 12 05/04

Last class

→ SVD

→ Magical properties of SVD

→ Computing top SV: Power Iteration

Today

→ Why/How fast does PI work?

→ Extensions & Connections to GD.

→ Remarks on Practice

→ Application to Matrix Completion

ASSIGNMENT 1, 2 GRADES UP ON GRADESCOPE

→ ASSIGNMENT 3 WILL BE UP TODAY BY 6PM.

RECAP:

INPUT: $X \in \mathbb{R}^{n \times d}$ $(n > d)$.

OUTPUT: v, that $\max_{\|v\|=1} \|Xv\|$.

Power Iteration

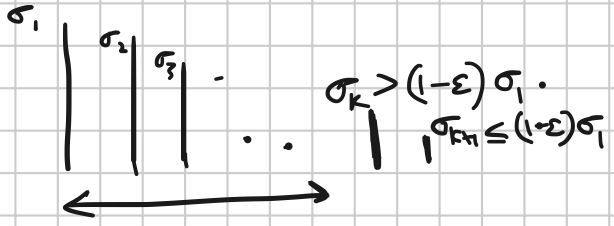
- v_0 is a random unit vector.
- For $t = 1, \ldots, T$:
  - $u = X^T \cdot (X \cdot v_{t-1})$
  - $v_t = \frac{u}{\|u\|}$.

Theorem: PI will converge to a top right S.V.

After $T = O\left(\frac{\log\left(\frac{d}{\varepsilon}\right)}{\varepsilon}\right)$ iterations, $\|X \cdot v_T\| \geq (1-\varepsilon)\sigma_1$.

$X = U \Sigma V^T$ be its SVD

$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_d$



Let k be the index such that

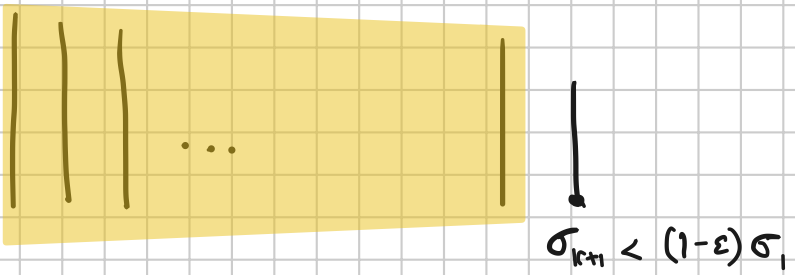
$\sigma_k > (1-\varepsilon)\sigma_1$

$\sigma_{k+1} \leq (1-\varepsilon)\sigma_1$.

Let S_k = Span of the first k singular vectors $v^{(1)}, v^{(2)}, \ldots, v^{(k)}$.

Thm: After $T = O\left(\frac{\log(\frac{d}{\delta})}{\varepsilon}\right)$ iterations V_T is almost entirely within S_k.

$$\| V_T - \text{Proj}_{S_k}(V_T) \| \leq \delta .$$



Thm: After T iterations

$$\| V_T - \text{Proj}_{S_k}(V_T) \| \leq \frac{(1-\varepsilon)^{2T}}{|\langle v_0, v^{(1)} \rangle|}$$

$v^{(1)}$ ↓ first right singular vector

Remark: If $\sigma_2 \leq \frac{\sigma_1}{2}$.

Then $\| V_T - \text{Proj}_{S_1}(V_T) \| \leq \frac{\left(\frac{1}{2}\right)^{2T}}{|\langle v_0, v^{(1)} \rangle|}$.

Cor: Suppose $\sigma_2 < (1-\varepsilon)\sigma_1$. Then,

$$\| V_T - \text{Proj}_{S_1}(V_T) \| \leq \frac{(1-\varepsilon)^{2T}}{|\langle v_0, v^{(1)} \rangle|}.$$

Proof:

$$V_1 = \frac{X^T \cdot (X \cdot (v_0))}{\| X^T (X (v_0)) \|}$$

$$V_2 = \frac{(X^T \cdot (X \cdot V_1))}{\| X^T \cdot (X \cdot V_1) \|}$$

$$\vdots$$

$$V_t = \frac{(X^T \cdot X)^t \cdot v_0}{\| (X^T \cdot X)^t \cdot v_0 \|}.$$

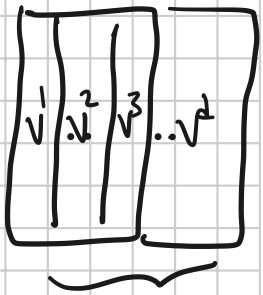
$$X = \sum_{i=1}^{d} \sigma_i u^{(i)} v^{(i)T}$$

$$X^T X = \sum_{i=1}^{d} \sigma_i^2 \cdot v^{(i)} v^{(i)T}.$$

$$(X^T X)^t = \sum_{i=1}^{d} \sigma_i^{2t} \cdot v^{(i)} v^{(i)T}.$$

$$(X^T X)^t \cdot v_0$$

$$V_t = \frac{(X^T X)^t \cdot V_0}{\|(X^T X)^t \cdot V_0\|}$$



Form an orthonormal basis

$$v^1, v^2, \ldots, v^d, v^{d+1}, \ldots, v^n.$$

$$\rightarrow V_0 = c_1 v^1 + c_2 v^2 + \cdots + c_d v^d + c_{d+1} v^{d+1} + \cdots + c_n \cdot v^n.$$

$$(X^T X)^t \cdot V_0 = \left(\sum_{i=1}^{d} \sigma_i^{2t} \cdot v^{(i)} v^{(i)T}\right) \cdot \left(V_0\right).$$

$$= \sum_{i=1}^{d} \sigma_i^{2t} \cdot \langle V_0, v^{(i)} \rangle \cdot v^{(i)}.$$

$$\text{Proj}_{\{v^{(i)}\}}\left((X^T X)^t \cdot V_0\right) = \sigma_1^{2t} \cdot \langle V_0, v^{(1)} \rangle \cdot v^{(1)}.$$

$$\|(X^T X)^t \cdot V_0\|^2 = \sum_{i=1}^{d} \sigma_i^{4t} \cdot \langle V_0, v^{(i)} \rangle^2.$$

$$\|V_t - \text{Proj}_{\{v^{(1)}\}}(V_t)\|^2$$

$$= \frac{\left\| \sum_{i=2}^{d} \sigma_i^{2t} \cdot \langle V_0, v^{(i)} \rangle \cdot v^{(i)} \right\|^2}{\|(X^T X)^t \cdot V_0\|^2}$$

$$= \frac{\sum_{i=2}^{d} \sigma_i^{4t} \cdot \langle V_0, v^{(i)} \rangle^2}{\quad}$$

$$\sum_{i=1}^{d} \sigma_i^{4t} \langle v_0, v^{(i)} \rangle^2$$

$$\text{Numerator} \leq \sum_{i=2}^{d} \left(\sigma_1 \cdot (1-\varepsilon)\right)^{4t} \cdot \langle v_0, v^{(i)} \rangle^2$$

(Recall $\sigma_2 \leq (1-\varepsilon)\sigma_1$)

$$= \sigma_1^{4t} (1-\varepsilon)^{4t} \cdot \left(\sum_{i=2}^{d} \langle v_0, v^{(i)} \rangle^2 \right)$$

$$\leq \sigma_1^{4t} \cdot (1-\varepsilon)^{4t} \cdot$$

$$\text{Denominator} \geq \sigma_1^{4t} \cdot \langle v_0, v^{(1)} \rangle^2 \cdot$$

$$\left\| v_t - \text{Proj}_{\{v_1\}}(v_t) \right\|^2 \leq \frac{\sigma_1^{4t} \cdot (1-\varepsilon)^{4t}}{\sigma_1^{4t} \cdot \langle v_0, v^{(1)} \rangle^2}$$

$$= \frac{(1-\varepsilon)^{4t}}{\langle v_0, v^{(1)} \rangle^2} \cdot$$

$\Rightarrow$ Corollary!

Power Iteration

→ v_0 : random unit vector

→ For $t = 1, \ldots, T$:

$$u = X^T \cdot (X \cdot v_{t-1})$$

$$v_t = \frac{u}{\|u\|}.$$

Practice.

X_1, X_2, X_3 : $X = X_1 \cdot X_2 \cdot X_3$.

$X_1 = U_1 \Sigma_1 V_1^T$, $X_2 = U_2 \Sigma_2 V_2^T$

1. All we need is ability to compute matrix-vector products for X.

Ex: To compute SVD of $X_1 \cdot X_2 \cdot X_3$

$(X_1 \cdot (X_2 \cdot (X_3 \cdot v)))$

LAPACK... have in built support for such computations.

2. How to compute higher singular vectors?

(Recall: If $X = U \Sigma V^T$

$\sigma_1 = \|X \cdot v^{(1)}\|$ ↳ first right SV.

$u^{(1)} = \frac{X \cdot v^{(1)}}{\sigma_1}$.

$X = \sigma_1 u^{(1)} v^{(1)T} + \sigma_2 u^{(2)} v^{(2)T} + \cdots + \sigma_d u^{(d)} v^{(d)T}$.

To compute 2^{nd}:

Compute 1^{st} of $\left(X - \sigma_1 u^{(1)} v^{(1)T}\right)$.

$\vdots$

So on.

(26). **General PI:**

→ Pick k orthonormal vectors $y_0^{(1)}, y_0^{(2)}, \ldots, y_0^{(k)}$.

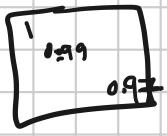
→ For $t = 1, \ldots, T$:

- Let $z^{(1)} = X^T \cdot X \cdot y_{t-1}^{(1)}$ $z^{(2)} = X^T \cdot X \cdot y_{t-1}^{(2)}$ $\ldots$, $z^{(k)} = X^T \cdot X \cdot y_{t-1}^{(k)}$.
- Let $y_t^{(1)}, y_t^{(2)}, \ldots, y_t^{(k)}$ be an orthonormal basis for $\text{Span}(\{z^{(1)}, z^{(2)}, \ldots, z^{(k)}\})$.

(3). Let's say X had singular values

→ $1, 0.99, 0.98, 0.97, 0.97, \ldots, 0.97$.

→ Suppose $X = V \Sigma V^T$



Practical hack...

→ $X - \frac{\Pi}{2}$ has singular values

$0.5, 0.49, 0.48, 0.47, \ldots, 0.47$.

→ $X - 0.96 \cdot \Pi$ has singular values

$0.04, 0.03, 0.02, 0.01, \ldots, 0.01$.

(4) PI gets convergence at the rate $\frac{1}{\varepsilon}$.

PI with momentum gets convergence at the rate $\frac{1}{\sqrt{\varepsilon}}$.

Power Iteration + "Momentum".

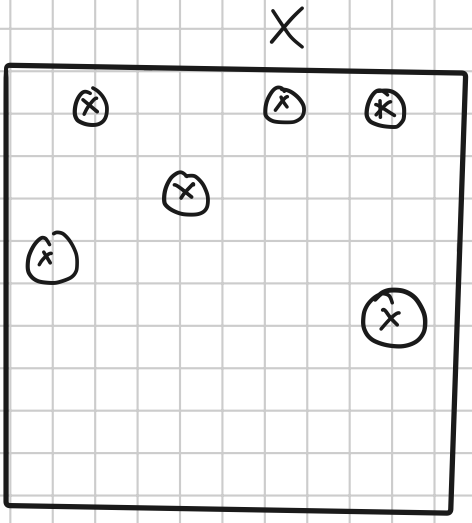
→ v_0 : random unit vector

→ For $t = 1, \dots, T$:

$$u = X^T \cdot (X \cdot v_{t-1}) - \beta \cdot v_{t-2} .$$

$$v_t = \frac{u}{\|u\|} .$$

MATRIX COMPLETION / NETFLIX CHALLENGE



→ We only see X_{ij} for $(i,j) \in$ Some set of {Observed entries}

$\Omega \subseteq [n] \times [d]$

INPUT: Entries of X in Ω, k

OUTPUT: Find $\tilde{X}$ of rank k to

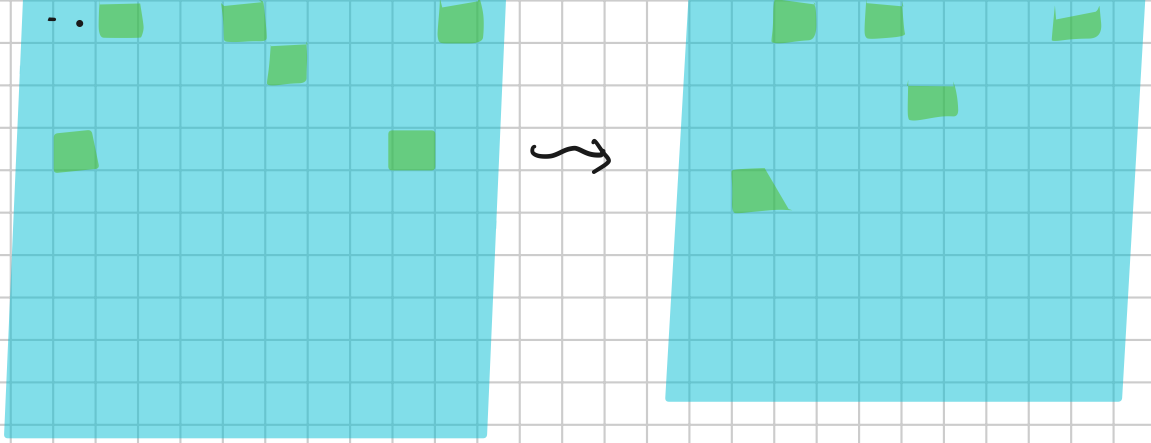
$$\min \sum_{(i,j) \in \Omega} (X_{ij} - \tilde{X}_{ij})^2 .$$

$$C = \{Y : \text{Rank}(Y) \le k\}$$

$$L(\tilde{X}) = \sum_{i,j \in \Omega} (X_{ij} - \tilde{X}_{ij})^2 .$$

$$\min_{\tilde{X} \in C} L(\tilde{X}) \equiv \min_{\text{Rank}(\tilde{X}) \le k} L(\tilde{X}) .$$

→ Non-Convex Optimization problem (NP-HARD).



Singular Value Projection (JMD 09).

→ X_0 :

→ For $t = 1 \ldots, T$:

$$Y_t = X_{t-1} - \eta \cdot \nabla L(X_{t-1})$$

$$X_t = (\text{Take top } k\text{-SVD of } Y_t) \rightarrow \text{Proj}_C(Y_t)$$

↓

The closest rank k matrix.