

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

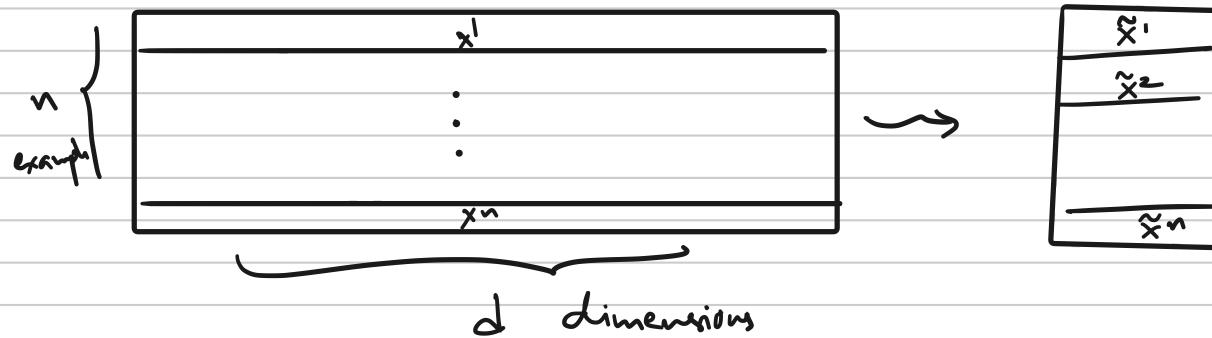
Last class

- Boosting
- Why Principal Components?
- Demo PCA

Today

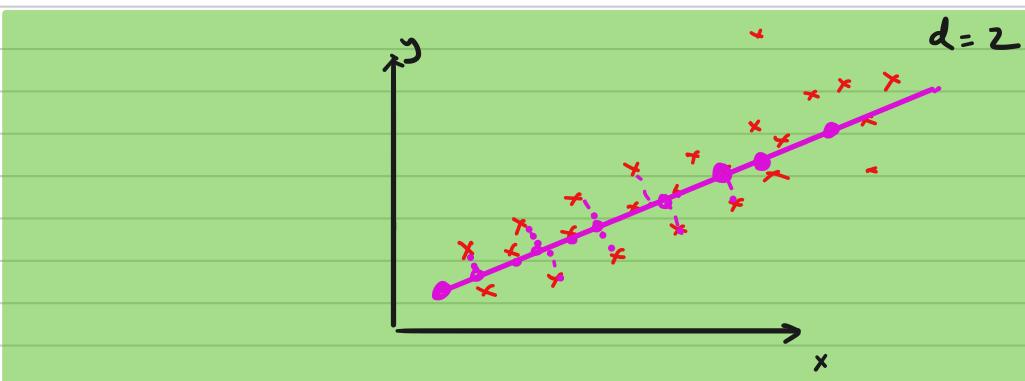
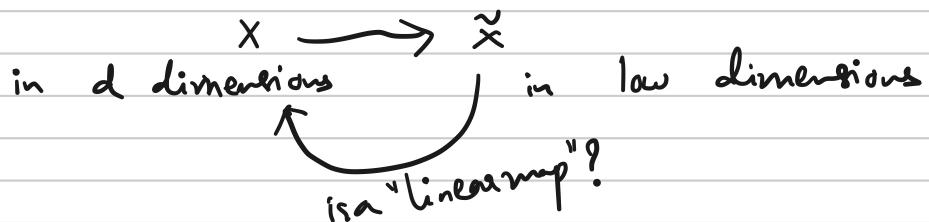
- What is PCA?
- Toward Computing PCs.

ASSIGNMENT 2 IS DUE TODAY @ 9:59 PM.



$$n \ll d.$$

Idea: We want to "represent"



BEST-FIT LINE PROBLEM

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

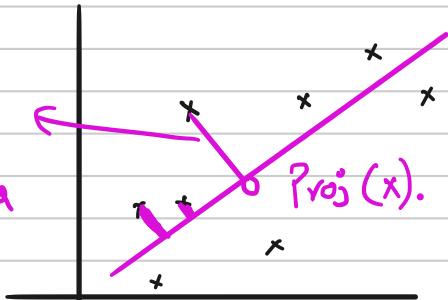
OUTPUT: Find the line that is "closest" to the dataset.

→ Defining Closest:

Minimize aggregate distance
of points to the line.

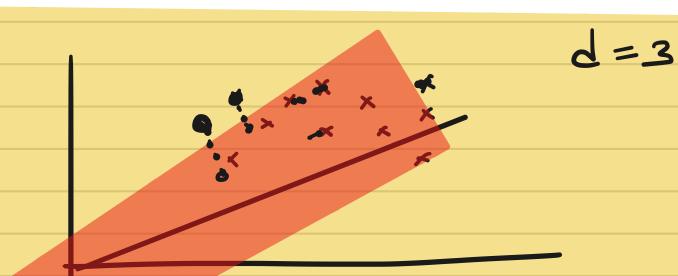
Measuring distance
of x to a line
as length of the perpendicular
to the line

$$= \|x - \text{Proj}(x)\|.$$



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

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



General Best-fit-Subspace Problem

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

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

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

OUTPUT: Find a k -dim. Subspace that minimizes $ERR(S; X)$.



- Return an orthonormal basis for the Subspace.

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

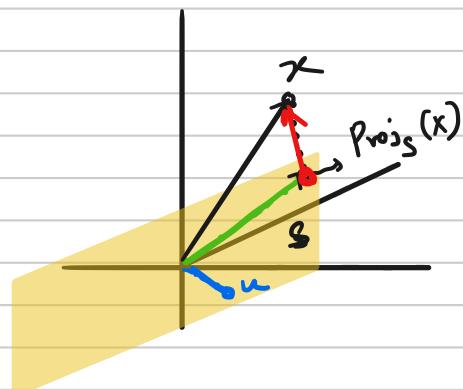
LINEAR ALGEBRA FACTS.

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

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

(angle is 90°)

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



④ For all points $u \in S$

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

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

Goal: Find 1-dimensional subspace S to minimize

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

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

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

So looking for an S to minimize $\text{ERR}(S; X)$

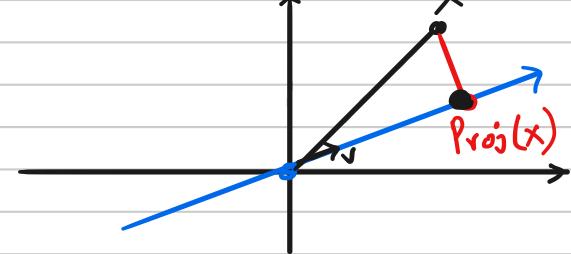
is the same as looking for an S to maximize

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

Minimizing $\text{ERR}(S)$ \equiv Maximizing $\text{Var}(S)$
over k -dim. Subspaces over k -dim. Subspaces

Focusing on $k=1$ Case:

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



v is a unit vector

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

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

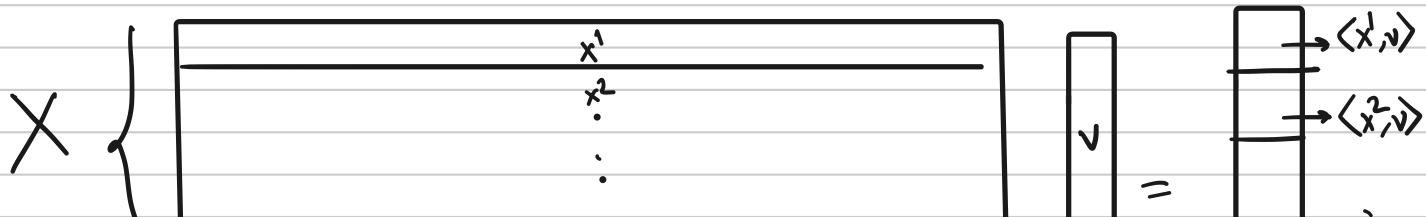
So for $k=1$: if $S = \text{span}\{\mathbf{v}\}$.

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

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

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

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



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

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

The best-fit-line problem is equivalent to

$$\arg \max_{v \neq 0} \frac{\|x \cdot v\|^2}{\|v\|^2}$$

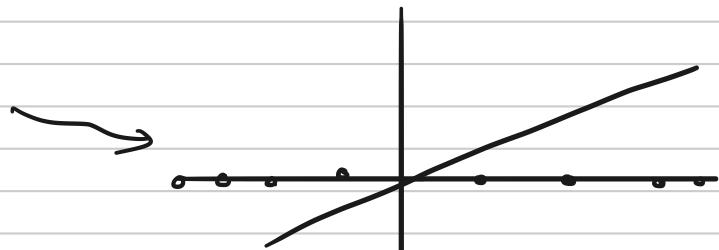
unit norm

- FIRST RIGHT SINGULAR VECTOR OF X .
- FIRST PRINCIPAL COMPONENT OF X .

Example 1:

$x =$	$\begin{array}{c cccccc c} 1 & 0 & 0 & - & \dots & 0 \\ \hline 0 & 0 & \cdot & & & 0 \\ \hline 1 & 0 & 0 & - & \dots & 0 \end{array}$
	$\begin{array}{c cccccc c} 1 & 0 & 0 & \dots & 0 \\ \hline \end{array}$

First-PC of $x = (1, 0, \dots, 0)$



Example 2:

$x =$	$\begin{array}{c} u \\ \hline 0.14 \\ \hline 1.04 \end{array}$
-------	--

First PC of $x = u$.

11 n

Summary:

Solving for \hat{v} \equiv Solving for \hat{v} \equiv Solving for
Best-fit-line $\max_{\hat{v}} \text{Var} = \arg \max_{\hat{v}: \|\hat{v}\|=1} \|X \cdot \hat{v}\|^2 \rightsquigarrow$ (First PC).

How about $k=2$?

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

Idea: To use a "greedy" approach.

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

Idea:

① Find First PC $\rightarrow v_1$

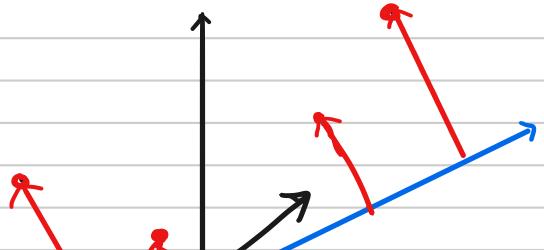
② Replace $\bar{x}^i = x^i - \text{Proj}_{\{v_1\}}(x^i)$

③ Find first PC of the "new"
dataset.

Idea 2:

$\arg \max_{\underline{\text{v}}} \|X \cdot \hat{v}\|^2$

}



$$v: \|v\| = 1$$

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

$$\arg \max_{v: \|v\|=1} \|X \cdot v\|^2 \quad \left. \right\} i^{\text{th}} \text{ PC of } X$$

$$v \perp v_1, v \perp v_2 \\ \dots, v \perp v_{i-1}$$

i^{th} Right singular vector
of X .

Theorem: The Span of first k right singular vectors minimizes $\text{ERR}(S; X)$.

(\equiv maximizes $\text{VAR}(S; X)$).

Remark: For $\text{ERR}_1(S; X) = \sum_{i=1}^n \|x^i - \text{Proj}_S(x^i)\|$,

finding best-fit-line, & then best-fit-line of left-overs does not work.

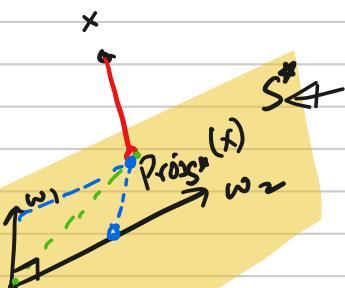
Theorem: Span of first two right singular vectors

maximizes $\text{Var}(S; x)$.

$$\dim(S) = 2$$

Proj: $S^* = \arg \max_{\dim(S)=2} \text{Var}(S; x)$

$$S^* = \text{Span}\{w_1, w_2\}$$



orthonormal basis for S^* .

$$v_1 \equiv \underset{\|v\|=1}{\operatorname{argmax}} \|x \cdot v\|$$

$$S = \text{Span}\{v_1, v_2\}$$

are
 \perp to each
other.

$$v_2 \equiv \underset{\|v\|=1}{\operatorname{argmax}} \|x \cdot v\| .$$

Claim:

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

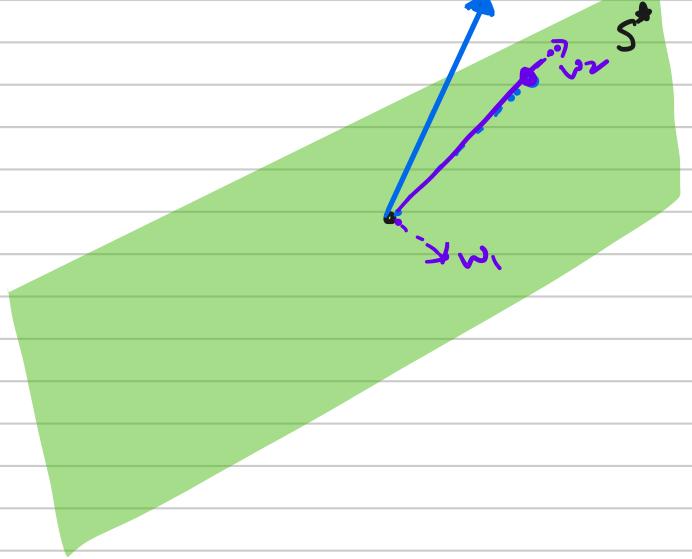
$$2. \quad \left\| \text{Proj}_S(x) \right\|^2 = \langle x, v_1 \rangle^2 + \langle x, v_2 \rangle^2 .$$

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

At the very least, by definition

$$\|x \cdot w_1\|^2 \leq \|x \cdot v_1\|^2 .$$

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



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

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

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

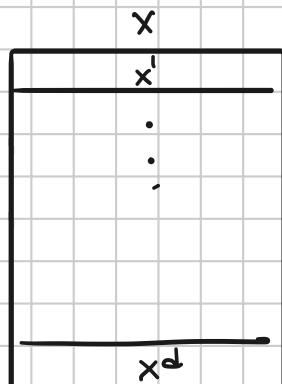
$\Rightarrow S$ maximizes the variance.

LAST CLASS

- Defining PCA
- min Error = max Variance
- Singular vectors
- Greedy strategy will work

Recap

Rows
= individual points



Today

- Low-Rank Approximation
- SVD: The Swiss-Army Knife of Linear Algebra
- Computing top singular vector.

$$\text{ERR}(S; x) = \sum_i \|x^i - \text{Proj}_S(x^i)\|_2^2$$

$$\text{VAR}(S; x) = \sum_i \|\text{Proj}_S(x^i)\|_2^2.$$

Best-Fit-Subspace

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

Right Singular Vector

First RSV: $\arg\max_{\|v\|=1} \|Xv\|$

k^{th} RSV:
 $v_k = \arg\max_{\substack{v \perp v_1 \\ v \perp v_2 \\ \vdots}} \|Xv\|$

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

Given some matrix

X

$n \}$



Approximate X

\Leftarrow in a "Simple" way.

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

SIMPLE = "LOW-RANK"

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

rank $\left(\begin{array}{c|c|c|c} u & u & \dots & u \end{array} \right)$ = 1. rank $\left(\begin{array}{cccc|c} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \dots & \dots & 0 \\ 0 & & \ddots & & 0 \\ \vdots & & & \ddots & 0 \\ 0 & 0 & \dots & \dots & 0 \end{array} \right)$ = n.

→ Why rank as a notion of Simplicity?

→ Probabilistic/statistical models

$$\xrightarrow{\quad} \left(\begin{array}{c|c|c|c} & & & d \\ & & & \times \\ n & & & \end{array} \right)$$

$$= \left(\begin{array}{c|c} n & d \\ \cup & \vee \\ n & n \end{array} \right)$$

$$X = \begin{matrix} U & \cdot & V \\ \downarrow & & \downarrow \\ n \times n & & n \times d \end{matrix} \quad \# \text{ parameters needed to work}$$

$U, V = n \cdot n + n \cdot d$

$= (n+d) \cdot n$.

2. Quantifying Approximation :

$$\text{Squared Error (RMSE):}$$

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

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

$$\boxed{x} = \boxed{x^2}$$

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 .

$$\text{OUTPUT : } \underset{\text{rank}(\tilde{x}) \leq k}{\operatorname{argmin}} \| x - \tilde{x} \|_F$$

NETFLIX CHALLENGE

≤ 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 for the missing entries to get a matrix $\underline{\underline{X}}$.

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

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

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

SINGULAR VALUE DECOMPOSITION.

Thm: Any matrix X

$$\begin{matrix} d \\ n \end{matrix} \begin{matrix} X \\ \end{matrix} = \begin{matrix} n \\ d \end{matrix} \begin{matrix} U \\ \Sigma \end{matrix} \begin{matrix} V^T \\ \end{matrix}$$

① Columns of U are orthonormal. $\underline{\underline{U}}^T \underline{\underline{U}} = \mathbb{I}_n$

② Columns of V are orthonormal. $(\underline{\underline{V}}^T \underline{\underline{V}} = \mathbb{I}_n)$

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

$$\begin{matrix} 1 & & \\ & 3 & \\ & & \textcircled{0} \\ & \textcircled{0} & 4 \\ & & x \end{matrix}$$

=

$$\begin{matrix} \text{II} \\ \cup \end{matrix}$$

$$\begin{matrix} 1 & & 0 \\ & 3 & \\ & & \textcircled{0} \\ & \textcircled{0} & 4 \\ & & z \end{matrix}$$

$$\begin{matrix} \text{II} \\ \cup^T \end{matrix}$$

②

$$\begin{matrix} 1 & 0 \\ 0 & -2 \end{matrix}$$

=

$$\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix}$$

U

$$\begin{matrix} 1 & 0 \\ 0 & 2 \end{matrix}$$

Σ

$$\begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix}$$

V^T

③

Orthogonal matrix

X : has columns

that are orthonormal

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

Two MAGICAL PROPERTIES OF SVD

$$X$$

=

$$\begin{matrix} & & \\ & \vdots & \\ u_1 & u_2 & u_3 \end{matrix}$$

$$\begin{matrix} \Sigma_{11} & & \\ & \Sigma_{22} & \\ & & \ddots \end{matrix}$$

$$\begin{matrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{matrix}$$

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

Thm: v_i is the i^{th} Right Singular Vector of X .

Thm: $X_k = \begin{matrix} n & k \\ \left[\begin{array}{c|c|c} u_1 & \dots & u_k \end{array} \right] & \left[\begin{array}{c} \Sigma_{11} \\ \Sigma_{22} \\ \vdots \\ \Sigma_{kk} \end{array} \right] & \left[\begin{array}{c} v_1^T \\ v_2^T \\ \vdots \\ v_k^T \end{array} \right] \end{matrix}$ is a solution to best rank k approximation!

(Eckart - Young - Mirsky Theorem 1936).

How do you get the SVD?

Some more properties of SVD.

$$X = U \sum V^T$$
$$(\Sigma_{11} > \Sigma_{22} > \dots > \Sigma_{nn}).$$

① $X = U \sum V^T$

$$X = \left[\begin{array}{c|c} u_1 & \dots u_n \end{array} \right] \left[\begin{array}{c} \Sigma \\ \vdots \\ \Sigma_{nn} \end{array} \right] \left[\begin{array}{c} v_1^T \\ \vdots \\ v_n^T \end{array} \right]$$

$$= \Sigma_{11} \left[\begin{array}{c} u_1 \\ \vdots \\ u_n \end{array} \right] + \Sigma_{22} \left[\begin{array}{c} u_2 \\ \vdots \\ u_n \end{array} \right] + \dots + \Sigma_{nn} \left[\begin{array}{c} u_n \\ \vdots \\ u_n \end{array} \right]$$
$$\sigma_1 \quad \sigma_2 \quad \dots \quad \sigma_n$$

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

$$\textcircled{2} \quad X \cdot v_i = \sigma_i \cdot u_i$$

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

$$= \sum_{l=1}^n \sigma_l \cdot u_l \cdot \underbrace{\langle u_l, v_i \rangle}_{\begin{array}{ll} 0 & \text{if } l \neq i \\ 1 & \text{if } l = i \end{array}}$$

$$= \sigma_i \cdot u_i \cdot)$$

$$\textcircled{3} \quad u_i^T X = \sigma_i v_i^T \cdot$$

$$\textcircled{4} \quad \sigma_i = \|X \cdot v_i\| \quad \left. \right\} \quad \begin{array}{l} (\text{as } X \cdot v_i = \sigma_i \cdot u_i \text{ and } u_i \text{ is a unit vector}) \end{array}$$

$$\textcircled{5} \quad u_i = \frac{X \cdot v_i}{\|X \cdot v_i\|} \cdot \quad \downarrow$$

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

Thm: Can compute full SVD directly in time

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

$n \cdot d = 8GB$ 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:

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

$$= U \underbrace{\Sigma}_{V^T} V^T.$$

$$Y = X^T \cdot X$$

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

$$= V \underbrace{\Sigma}_{U^T U} \Sigma V^T$$

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

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

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

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

$$Y^l = V \cdot \Sigma^{2l} \cdot V^T.$$

$$= \begin{array}{c} | \\ V_1 \quad V_2 \cdots \quad V_n \\ | \end{array} \quad \begin{array}{c} \sigma_1^{2l} \quad \sigma_2^{2l} \quad 0 \\ 0 \quad \sigma_2^{2l} \quad \cdots \\ 0 \quad 0 \quad \sigma_n^{2l} \end{array} \quad \begin{array}{c} V_1^T \\ V_2^T \\ \vdots \\ V_n^T \end{array}$$

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

Example: Suppose $\sigma_1 = 1, \sigma_2 = \frac{1}{2}, \sigma_3 = \frac{1}{3}, \dots$

$$\begin{aligned} Y^l &= 1 \cdot V_1 \cdot V_1^T + \left(\frac{1}{2}\right)^{2l} V_2 \cdot V_2^T + \sigma_3^{2l} V_3 \cdot V_3^T + \cdots + \\ &= V_1 \cdot V_1^T + \sum_{i=2}^n \sigma_i^{2l} \cdot V_i \cdot V_i^T \end{aligned}$$

As $l \rightarrow \infty$,

$$V_1 V_1^T = \begin{array}{c} | \\ V_1 \cdot (V_{1,1}) \quad V_1 \cdot (V_{1,2}) \cdots \\ | \end{array}$$

at a single column

Look
to get v_i !

Idea:

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

② Compute first column of y^t .

$$\bar{v} = \begin{matrix} y^t \\ \vdots \\ v \cdot v \cdot v \dots v \end{matrix}$$

$\underbrace{\quad \quad \quad}_{t \text{ times}}$

③ Output

$$\frac{\bar{v}}{\|\bar{v}\|}$$

$$\bar{v} = \left(y \cdot \left(y \cdot \left(y \cdot \left(\dots \cdot y \cdot \left(y \left(\begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix} \right) \right) \right) \right) \right)$$

$\underbrace{\quad \quad \quad}_{100 \text{ times}}$

POWER ITERATION

- $\bar{v}_0 = \begin{matrix} 1 \\ 0 \\ \vdots \\ 0 \end{matrix}$
- For $l=1, \dots, t$ \rightsquigarrow
- $\bar{v}_l = \frac{X^T \cdot (X \cdot \bar{v}_{l-1})}{\|\bar{v}_l\|}$
- Output

$\} \equiv$ Give
first RSV.

Last class

→ SVD

→ Magical properties of SVD

→ Computing top SV: Power Iteration

Today

→ Why/How fast does PI work?

→ Extensions & Connections to GID.

→ 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

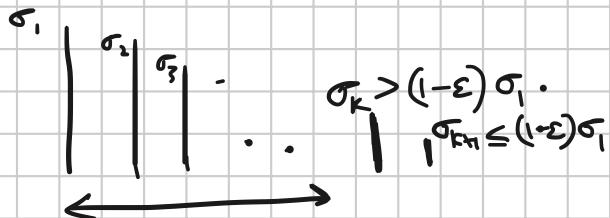
- v_0 is a random unit vector.- For $t = 1, \dots, T$:

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

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

Theorem: PI will converge to a top right S.V.After $T = O\left(\log\left(\frac{d}{\epsilon}\right)\right)$ iterations, $\|X \cdot v_T\| \geq (1-\epsilon) \sigma_1$. $X = U \Sigma V^T$ be its SVD

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_d$$



Let k be the index such that

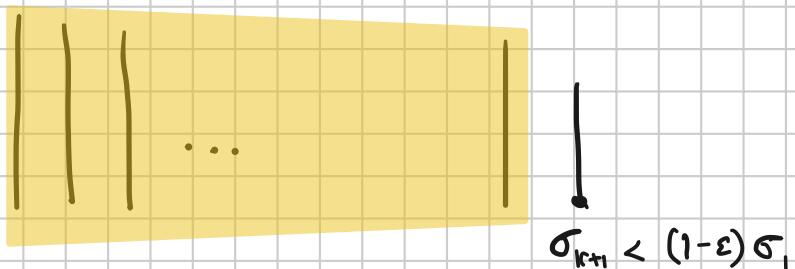
$$\sigma_k > (1 - \varepsilon)\sigma_1$$

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

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

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

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

↓
first right singular vector

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

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

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

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

Proof:

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

$$V_2 = \frac{(x^T \cdot (x \cdot v_1))}{\| x^T \cdot (x \cdot v_1) \|}$$

$$\vdots$$

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

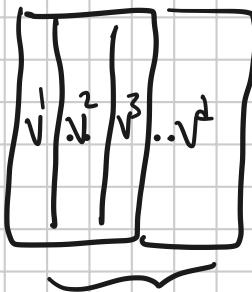
$$x = \sum_{i=1}^d \sigma_i u^{(i)} v^{(i)T}$$

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

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

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

$$v_t = \frac{(x^T x)^t \cdot v_0}{\|(x^T x)^t \cdot v_0\|}$$



form an orthonormal basis

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

$$\rightarrow v_0 = c_1 v^1 + c_2 v^2 + \dots + c_d v^d + c_{d+1} v^{d+1} + \dots + c_n v^n$$

$$(x^T x)^t \cdot v_0 = \left(\sum_{i=1}^d \sigma_i^{2t} \cdot v^{(i)} v^{(i)T} \right) \cdot (v_0)$$

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

$$\text{Proj}_{\{v^{(i)}\}} ((x^T x)^t \cdot v_0) = \sigma_1^{2t} \cdot \langle v_0, v^{(1)} \rangle \cdot v^{(1)}$$

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



$$\|v_t - \text{Proj}_{\{v^{(i)}\}} (v_t)\|^2$$

$$= \left\| \sum_{i=2}^d \sigma_i^{2t} \cdot \langle v_0, v^{(i)} \rangle \cdot v^{(i)} \right\|^2$$

$$\|(x^T x)^t \cdot v_0\|^2$$

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

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

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

↓

Numerator $\leq \sum_{i=2}^d \underbrace{(\sigma_i \cdot (1-\varepsilon))^{4t}}_{\leq} \cdot \langle v_0, v^{(i)} \rangle^2$ (Recall $\sigma_2 \leq (1-\varepsilon)\sigma_1$)

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

↓

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

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

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

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

\Rightarrow Corollary!

Power Iteration

$\rightarrow v_0$: random unit vector

\rightarrow For $t=1, \dots, T$:

$$u = x^\top (x \cdot v_{t-1})$$

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

Practice.

$$x_1, x_2, x_3 : X = \frac{x_1}{\|x_1\|} \cdot \frac{x_2}{\|x_2\|} \cdot \frac{x_3}{\|x_3\|}$$

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

Ex: To compute SVD of x_1, x_2, x_3
 $(x_1, (x_2, (x_3, v)))$

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

2. How to compute higher singular vectors?

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

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

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

$$X = \sigma_1 u^{(1)} v^{(1)T} + \sigma_2 u^{(2)} v^{(2)T} + \dots + \sigma_n u^{(n)} v^{(n)T}.$$

To compute z^{nd} :

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

do on.

② b.

General PI:

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

→ For $t=1, \dots, T$:

- Let $\hat{z}^{(1)} = \hat{X}^T \cdot X \cdot y_{t-1}^{(1)} z = \hat{X}^T \cdot X \cdot y_{t-1}^{(1)}$
 $\dots, \hat{z}^{(k)} = \hat{X}^T \cdot X \cdot y_{t-1}^{(k)}$.

- Let $y_t^{(1)}, y_t^{(2)}, \dots, y_t^{(k)}$ be an orthonormal basis for $\text{Span}(\{\hat{z}^{(1)}, \hat{z}^{(2)}, \dots, \hat{z}^{(k)}\})$.

③. Let's say X had singular values

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

→ Suppose $X = \sqrt{\sum} \sqrt{V^T}$

Practical work... → $X - \frac{\pi}{2} \cdot$ has singular values

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

→ $X - 0.96 \cdot \frac{\pi}{2}$ has singular values

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

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".

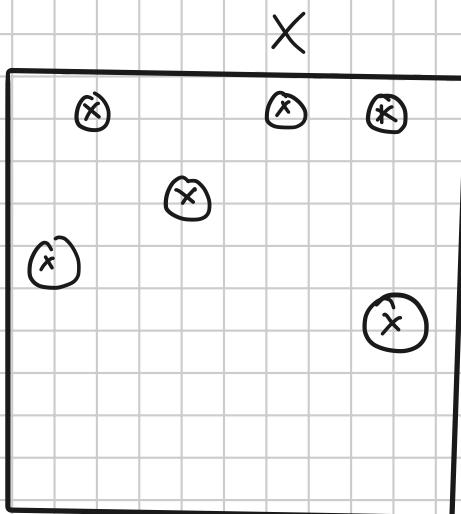
→ v_0 : random unit vector

→ For $t=1, \dots, T$:

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

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

MATRIX COMPLETION / NETFLIX CHALLENGE



→ We only see X_{ij} for $(i, j) \in \Theta$ some set of observed entries.

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

INPUT: Entries of X in Θ , k

OUTPUT: Find \tilde{X} of rank k to
 $\min \sum_{(i,j) \in \Theta} (X_{ij} - \tilde{X}_{ij})^2$.

$$C = \{Y : \text{Rank}(Y) \leq k\} \quad L(\tilde{X}) = \sum_{i,j \in \Theta} (X_{ij} - \tilde{X}_{ij})^2.$$

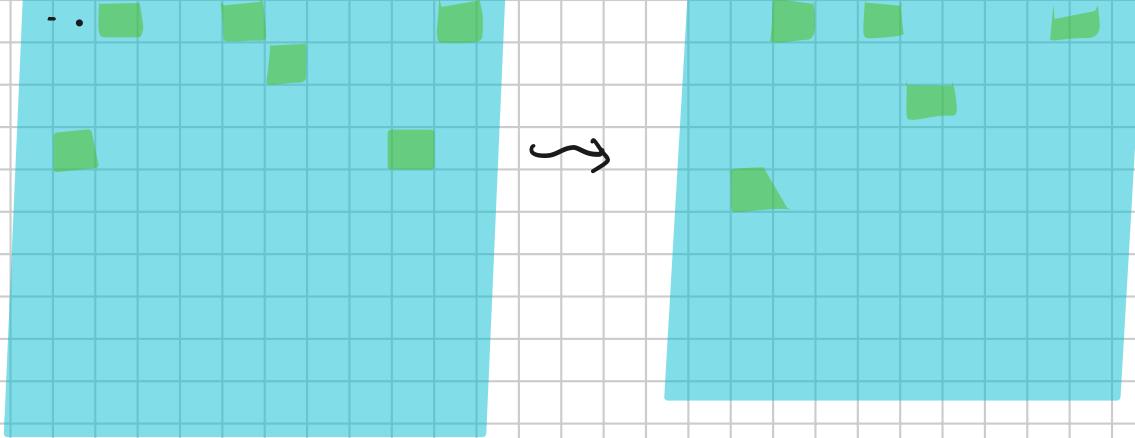
$$\min_L(\tilde{X})$$

$$\tilde{X} \in C.$$

$$\min L(\tilde{X})$$

$$\text{Rank}(\tilde{X}) \leq k.$$

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



Singular Value Projection (TMD 09).

$\rightarrow X_0 :$

$\rightarrow \text{for } t=1 \dots, T :$

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

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

The closest rank
k matrix.