

Machine Learning for Data Science (CS4786)

Lecture 4

PCA and Random Projections

DIM REDUCTION: LINEAR TRANSFORMATION

$$n \begin{matrix} X \\ \times d W \\ K \end{matrix} = n Y$$

d

$$\mathbf{y}_i^\top = \mathbf{x}_i^\top W$$

K

PCA: VARIANCE MAXIMIZATION

- Pick directions along which data varies the most

$$\text{Spread} = \frac{1}{n} \sum_{t=1}^n \text{dist}^2 \left(y_t, \frac{1}{n} \sum_{t=1}^n y_t \right) = \sum_{j=1}^K \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\begin{aligned} & \text{Maximize} && \sum_{j=1}^K \mathbf{w}_j^\top \Sigma \mathbf{w}_j \\ & \text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \text{ \& } \mathbf{w}_j \perp \mathbf{w}_i \end{aligned}$$

Σ is the covariance matrix

This solution is given by $W = \text{Top } K \text{ eigenvectors of } \Sigma$

PRINCIPAL COMPONENT ANALYSIS (PCA)

Turk & Pentland'91

Eigen Face:

$$x_t = \mu + y_t[1] * w_1 + y_t[2] * w_2 + y_t[3] * w_3$$

Xt $\mu =$ **Mean face**

$y_t[1] = -0.1945 * w_1$

$y_t[2] = 0.0461 * w_2$

$y_t[3] = 0.0586 * w_3$

The diagram shows the decomposition of a face image x_t into a mean face μ and three eigen faces w_1, w_2, w_3 . The equation is $x_t = \mu + y_t[1] * w_1 + y_t[2] * w_2 + y_t[3] * w_3$. Below the equation, it is shown that μ is the "Mean face". The eigen faces w_1, w_2, w_3 are orthogonal patterns of grayscale values. The coefficients $y_t[1], y_t[2], y_t[3]$ are numerical values indicating the contribution of each eigen face to the original face x_t .

- Each x_t (each row of X) is a face image (vectorized version)
- Each y_t is the set of coefficients we multiply to the eigen face
- w_i 's are orthogonal to each other and of unit length

PCA: MINIMIZING RECONSTRUCTION ERROR

- Goal: find the basis that minimizes reconstruction error,
- (Centered) Data-points as linear combination of some orthonormal basis, i.e.



$$\mathbf{x}_t = \boldsymbol{\mu} + \sum_{j=1}^d \mathbf{y}_t[j] \mathbf{w}_j$$

where $\mathbf{w}_1, \dots, \mathbf{w}_d \in \mathbb{R}^d$ are the orthonormal basis and $\boldsymbol{\mu} = \frac{1}{n} \sum_{t=1}^n \mathbf{x}_t$.

- Represent data as linear combination of just K orthonormal basis,



$$\hat{\mathbf{x}}_t = \boldsymbol{\mu} + \sum_{j=1}^K \mathbf{y}_t[j] \mathbf{w}_j$$

PCA: MINIMIZING RECONSTRUCTION ERROR

- Goal: find the basis that minimizes reconstruction error,

$$\begin{aligned}\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 &= \left\| \sum_{j=1}^K \mathbf{y}_t[j] \mathbf{w}_j + \mu - \mathbf{x}_t \right\|_2^2 \\ &= \left\| \sum_{j=1}^K \mathbf{y}_t[j] \mathbf{w}_j + \mu - \sum_{j=1}^d \mathbf{y}_t[j] \mathbf{w}_j - \mu \right\|_2^2 \\ &= \left\| \sum_{j=K+1}^d \mathbf{y}_t[j] \mathbf{w}_j \right\|_2^2 \quad (\text{but } \|a+b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 + 2a^\top b) \\ &= \left(\sum_{j=K+1}^d \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2 + 2 \sum_{j=K+1}^d \sum_{i=j+1}^d \mathbf{y}_t[j] \mathbf{y}_t[i] \mathbf{w}_j^\top \mathbf{w}_i \right) \\ &= \sum_{j=k+1}^d \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2 \quad (\text{last step because } \mathbf{w}_j \perp \mathbf{w}_i)\end{aligned}$$

PCA: MINIMIZING RECONSTRUCTION ERROR

$$\frac{1}{n} \sum_{t=1}^n \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \frac{1}{n} \sum_{t=1}^n \sum_{j=k+1}^d \mathbf{y}_t[j]^2 \|\mathbf{w}_j\|_2^2 \quad (\text{but } \|\mathbf{w}_j\| = 1)$$

$$= \frac{1}{n} \sum_{t=1}^n \sum_{j=k+1}^d \mathbf{y}_t[j]^2 \quad (\text{now } \mathbf{y}_j = \mathbf{w}_j^\top (\mathbf{x}_t - \mu))$$

$$= \frac{1}{n} \sum_{t=1}^n \sum_{j=k+1}^d (\mathbf{w}_j^\top (\mathbf{x}_t - \mu))^2$$

$$= \frac{1}{n} \sum_{t=1}^n \sum_{j=k+1}^d \mathbf{w}_j^\top (\mathbf{x}_t - \mu) (\mathbf{x}_t - \mu)^\top \mathbf{w}_j$$

$$= \sum_{j=k+1}^d \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

PCA: MINIMIZING RECONSTRUCTION ERROR

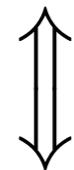
$$\begin{aligned} \text{Minimize} \quad & \sum_{j=K+1}^d \mathbf{w}_j^\top \Sigma \mathbf{w}_j \\ \text{s.t.} \quad & \forall j, \|\mathbf{w}_j\|_2^2 = 1 \text{ \& } \mathbf{w}_j \perp \mathbf{w}_i \end{aligned}$$

Maximize Total Spread

Minimize Reconstruction
Error

$$\frac{1}{n} \sum_{t=1}^n \text{dist}^2 \left(y_t, \frac{1}{n} \sum_{t=1}^n y_t \right)$$

$$\frac{1}{n} \sum_{t=1}^n \| \mathbf{x}_t - \hat{\mathbf{x}}_t \|_2^2$$



$$\text{Maximize} \quad \sum_{j=1}^K \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \text{ \& } \mathbf{w}_j \perp \mathbf{w}_i$$

$$\text{Minimize} \quad \sum_{j=K+1}^d \mathbf{w}_j^\top \Sigma \mathbf{w}_j$$

$$\text{s.t. } \forall j, \|\mathbf{w}_j\|_2^2 = 1 \text{ \& } \mathbf{w}_j \perp \mathbf{w}_i$$

Claim

Maximize Total Spread = Minimize Reconstruction
Error

PCA: MINIMIZING RECONSTRUCTION ERROR

Claim: $\sum_{j=1}^d \mathbf{w}_j^\top \Sigma \mathbf{w}_j = \text{Constant} = \frac{1}{n} \sum_{t=1}^n \|x_t - \mu\|_2^2$

Recall that: $\frac{1}{n} \sum_{t=1}^n \|\hat{\mathbf{x}}_t - \mathbf{x}_t\|_2^2 = \sum_{j=K+1}^d \mathbf{w}_j^\top \Sigma \mathbf{w}_j$

Take $K = 0$ so that $\hat{\mathbf{x}}_t = \mu$

Maximize Total Spread = Minimize Reconstruction Error

$$\text{Minimize} \sum_{j=K+1}^d \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\sum_{j=1}^d \mathbf{w}_j^\top \Sigma \mathbf{w}_j - \sum_{j=1}^K \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(\frac{1}{n} \sum_{t=1}^n \|\mathbf{x}_t - \mu\|_2^2 - \sum_{j=1}^K \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Minimize} \left(- \sum_{j=1}^K \mathbf{w}_j^\top \Sigma \mathbf{w}_j \right) \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

$$\iff \text{Maximize} \sum_{j=1}^K \mathbf{w}_j^\top \Sigma \mathbf{w}_j \quad \text{s.t. } \|\mathbf{w}_j\|_2 = 1, \mathbf{w}_j \perp \mathbf{w}_k$$

PRINCIPAL COMPONENT ANALYSIS

1.

$$\Sigma = \text{cov}(X)$$

2.

$$W = \text{eigs}(\Sigma, K)$$

3.

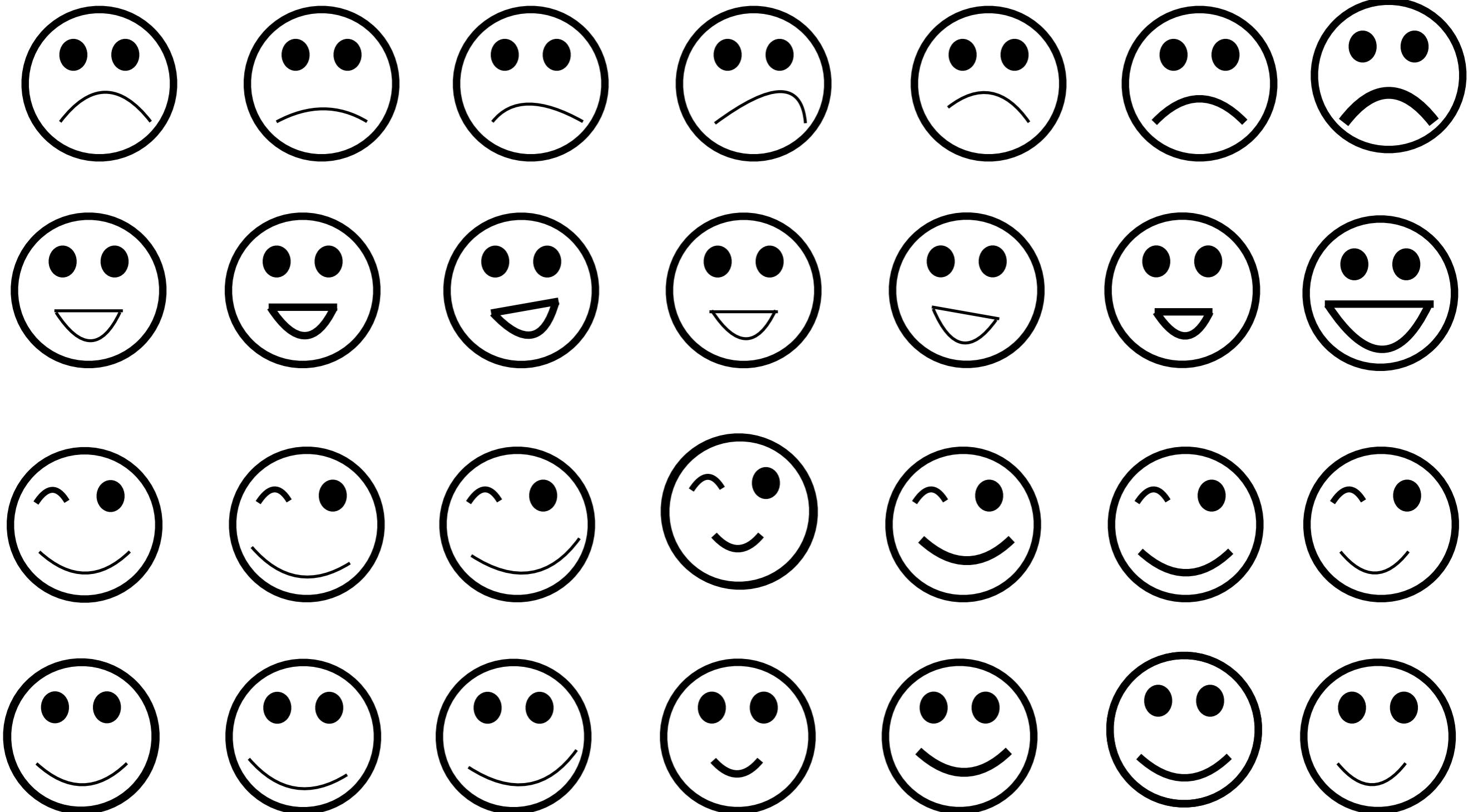
$$Y = X - \mu \times W$$

RECONSTRUCTION

4.

$$\hat{X} = Y \times W^\top + \mu$$

PRINCIPAL COMPONENT ANALYSIS: DEMO



WHEN $d \gg n$

- If $d \gg n$ then Σ is large
- But we only need top K eigen vectors.
- Idea: use SVD

$$X - \mu = UDV^\top$$

$$\begin{aligned} V^\top V &= I \\ U^\top U &= I \end{aligned}$$

Then note that, $\Sigma = (X - \mu)^\top(X - \mu) = VD^2V$

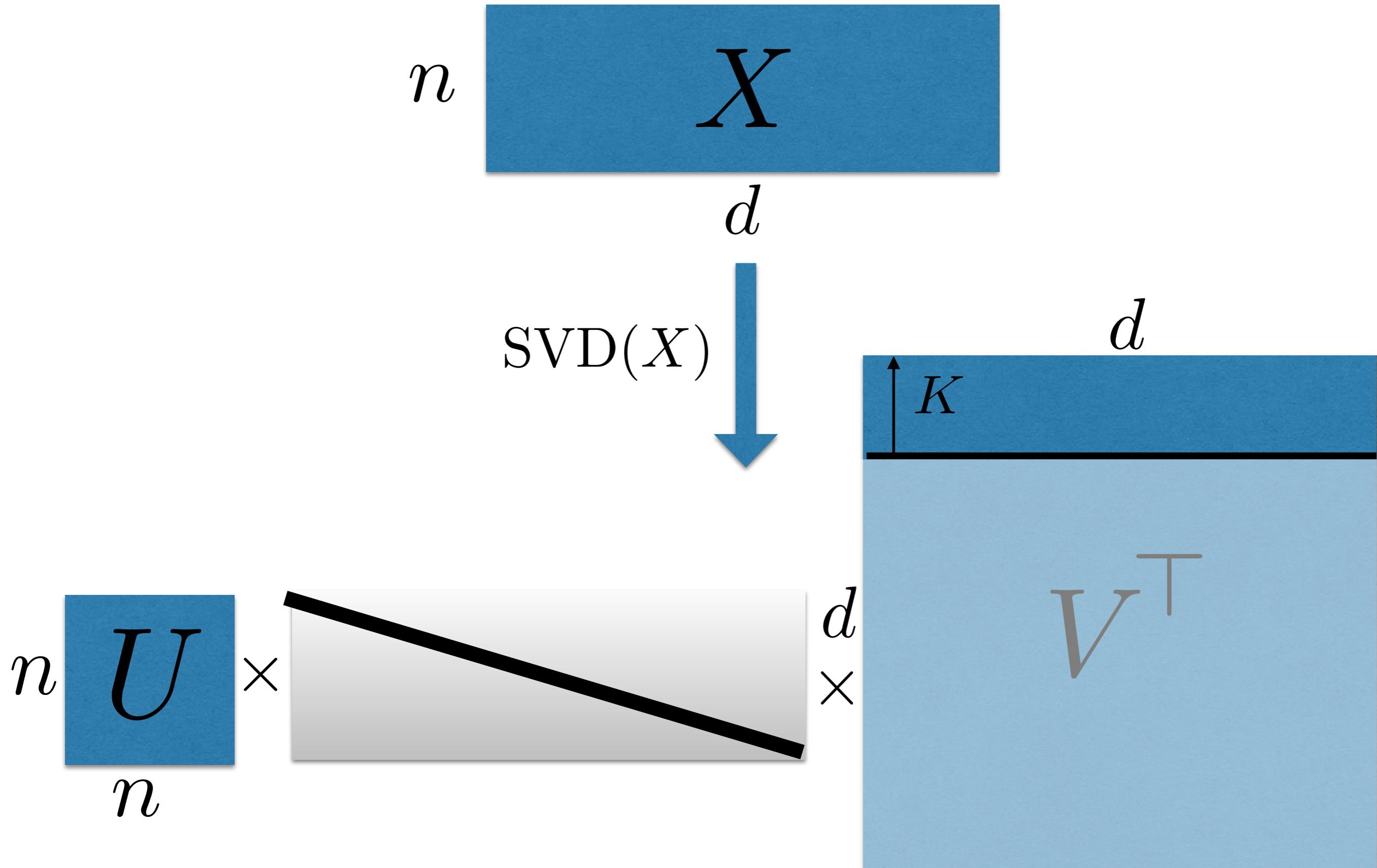
- Hence, matrix V is the same as matrix W got from eigen decomposition of Σ , eigenvalues are diagonal elements of D^2
- Alternative algorithm:

$$[U, V] = \text{SVD}(X - \mu, K) \quad W = V$$

The Tall, THE FAT AND THE UGLY

$$\begin{matrix} d & X^\top \\ n & \end{matrix} \times_n \begin{matrix} d \\ X \end{matrix} \quad \diagdown \quad n = d \sum_{\Sigma}^d$$
$$d \begin{matrix} W \\ K \end{matrix} = \text{Eigs}\left(\sum, K\right)$$

THE TALL, the Fat AND THE UGLY



THE TALL, THE FAT AND the Ugly

X



- d and n so large we can't even store in memory
- Only have time to be linear in $\text{size}(X) = n \times d$

Is there any hope?

PICK A RANDOM W

$$Y = X \times \begin{bmatrix} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \ddots & \\ & \ddots & \\ +1 & \dots & -1 \end{bmatrix}^d \sqrt{K}$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

RANDOM PROJECTION

- What does “it works” even mean?

Distances between all pairs of data-points in low dim. projection is roughly the same as their distances in the high dim. space.

That is, when K is “large enough”, with “high probability”, for all pairs of data points $i, j \in \{1, \dots, n\}$,

$$(1 - \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2 \leq \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq (1 + \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Lets start with a one dimensional projection ($K = 1$)

$$y_t = \mathbf{x}_t^\top \mathbf{u} \quad \text{where each } \mathbf{u}[i] = \text{random } \pm 1$$

- What is the expected value of:

$$1. \quad y_t - y_s ?$$

$$2. \quad (y_t - y_s)^2 ?$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence for any $s, t \in \{1, \dots, n\}$,

$$\mathbb{E}[|\mathbf{y}_s - \mathbf{y}_t|^2] = \|\mathbf{x}_s - \mathbf{x}_t\|_2^2$$

Lets try ...

Law of large numbers says that average over multiple draws is close to expectation

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

$$\mathbf{y}_t[k] = \mathbf{x}_t^\top \mathbf{u}_k / \sqrt{K} \quad \text{where each } \mathbf{u}_k[i] = \text{random } \pm 1$$

$$(\mathbf{y}_s[k] - \mathbf{y}_t[k])^2 = (\mathbf{x}_t^\top \mathbf{u}_k - \mathbf{x}_s^\top \mathbf{u}_k)^2 / K$$

$$\|\mathbf{y}_t - \mathbf{y}_s\|_2^2 = \sum_{k=1}^K (\mathbf{y}_s[k] - \mathbf{y}_t[k])^2 = \frac{1}{K} \sum_{k=1}^K (\mathbf{x}_t^\top \mathbf{u}_k - \mathbf{x}_s^\top \mathbf{u}_k)^2$$

This is an average over K trials

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

For any $\epsilon > 0$, if $K \approx \log(n/\delta) / \epsilon^2$, with probability $1 - \delta$ over draw of W , for all pairs of data points $i, j \in \{1, \dots, n\}$,

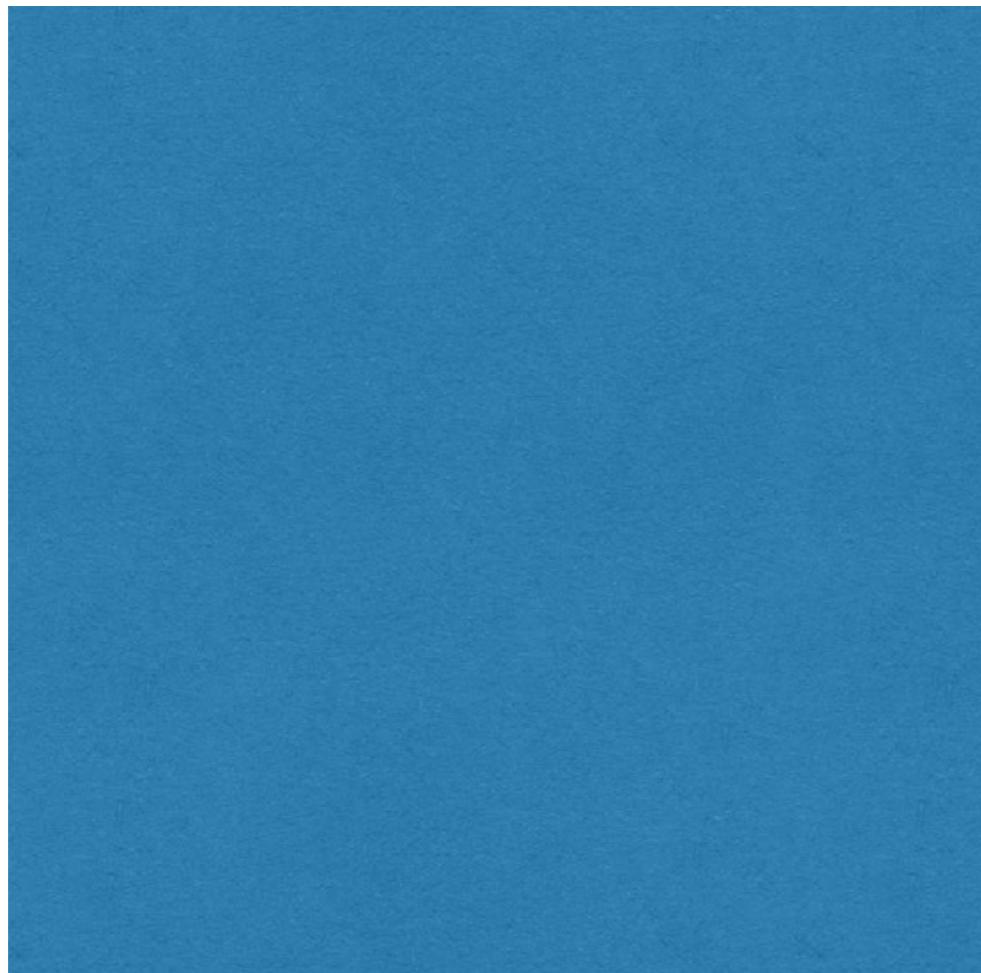
$$(1 - \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \leq \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \leq (1 + \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$$

Lets try ...

This is called the Johnson-Lindenstrauss lemma or JL lemma for short.

WHY IS THIS SO RIDICULOUSLY MAGICAL?

$n =$
1000

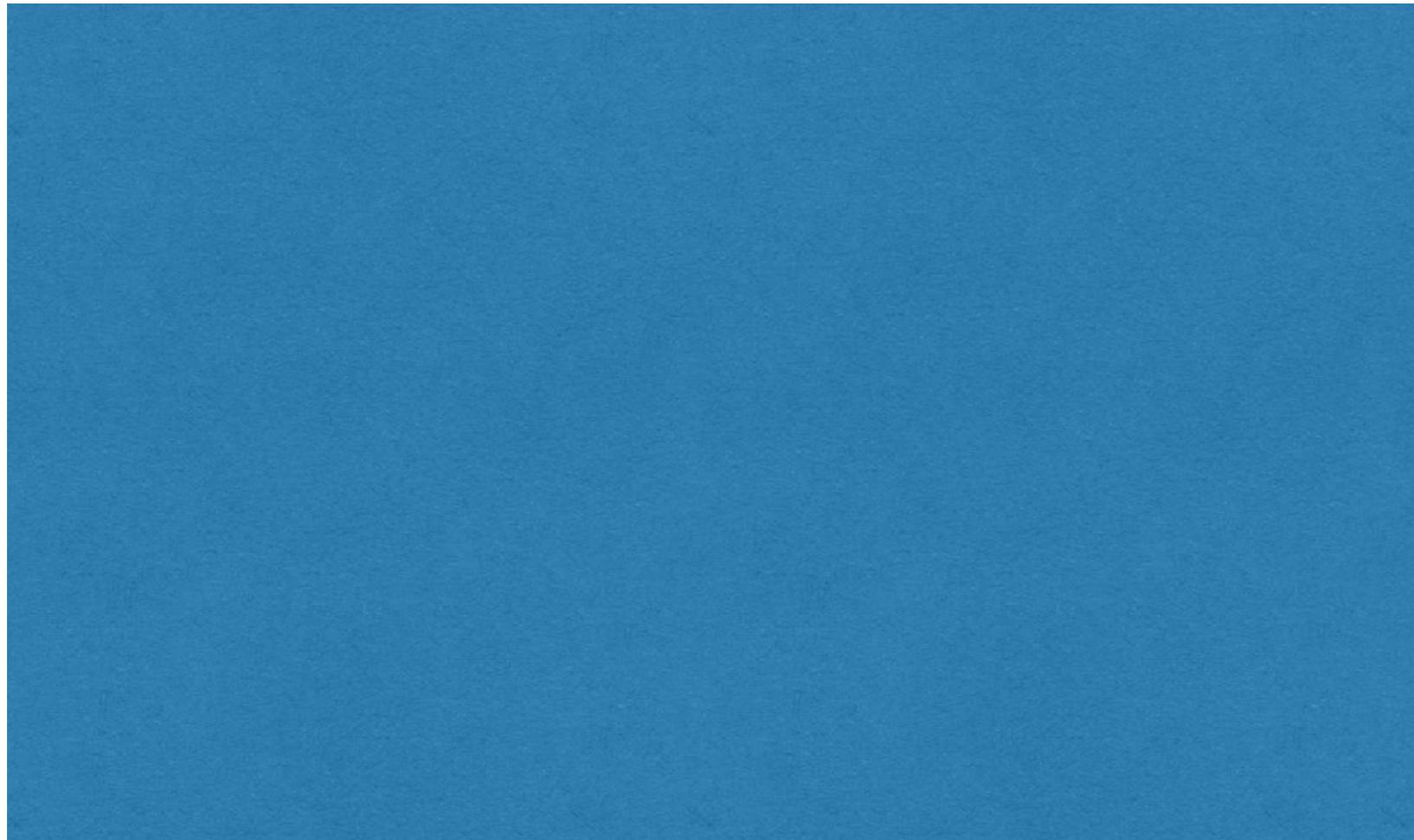


$d = 1000$

If we take $K = 69.1/\epsilon^2$, with probability
0.99 distances are preserved to accuracy ϵ

WHY IS THIS SO RIDICULOUSLY MAGICAL?

$n =$
1000



$d = 10000$

If we take $K = 69.1/\epsilon^2$, with probability
0.99 distances are preserved to accuracy ϵ

WHY IS THIS SO RIDICULOUSLY MAGICAL?

$n =$
1000

$d = 1000000$

If we take $K = 69.1/\epsilon^2$, with probability
0.99 distances are preserved to accuracy ϵ