

Machine Learning for Data Science (CS4786)

Lecture 6

Random Projections

Feb 09, 2015

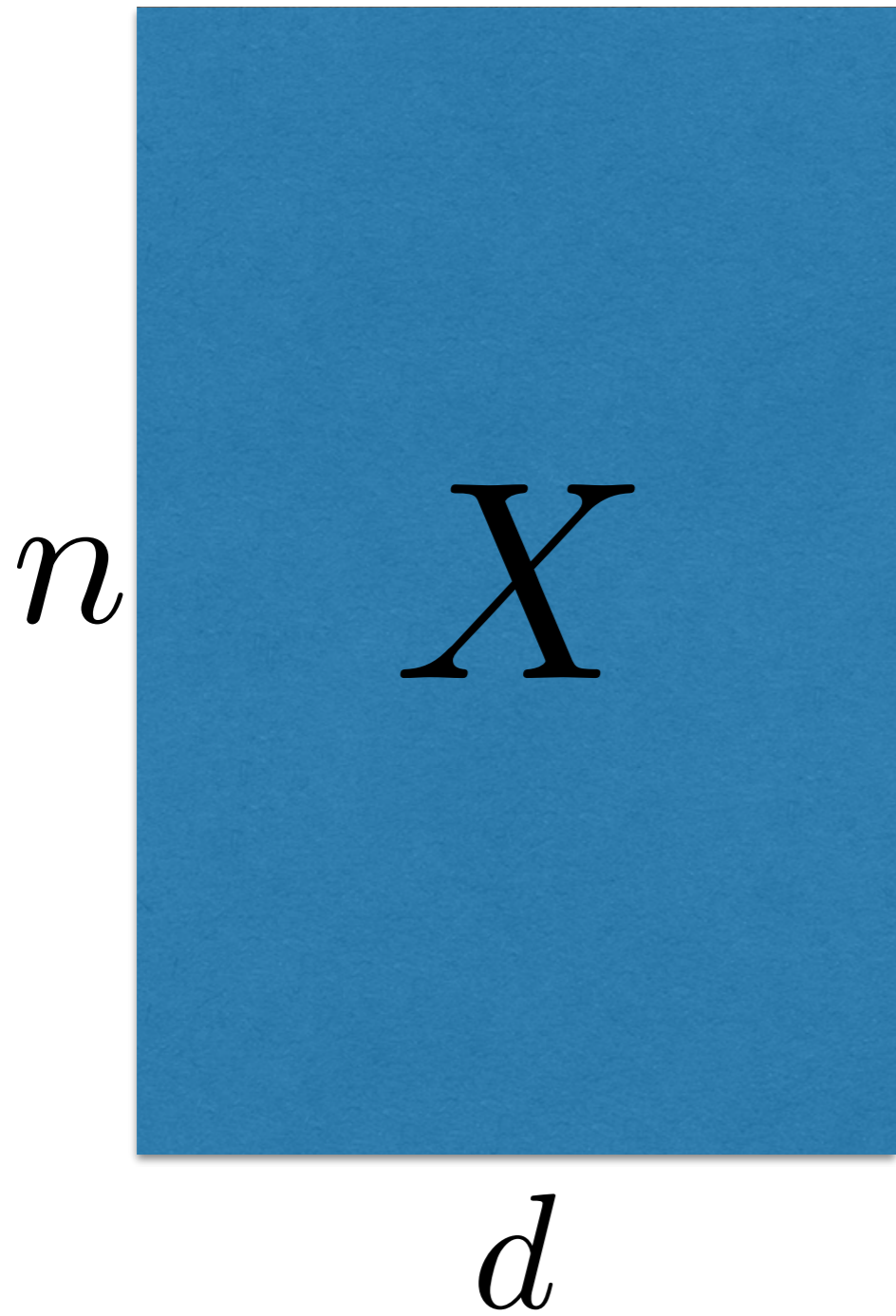
Course Webpage :

<http://www.cs.cornell.edu/Courses/cs4786/2015sp/>

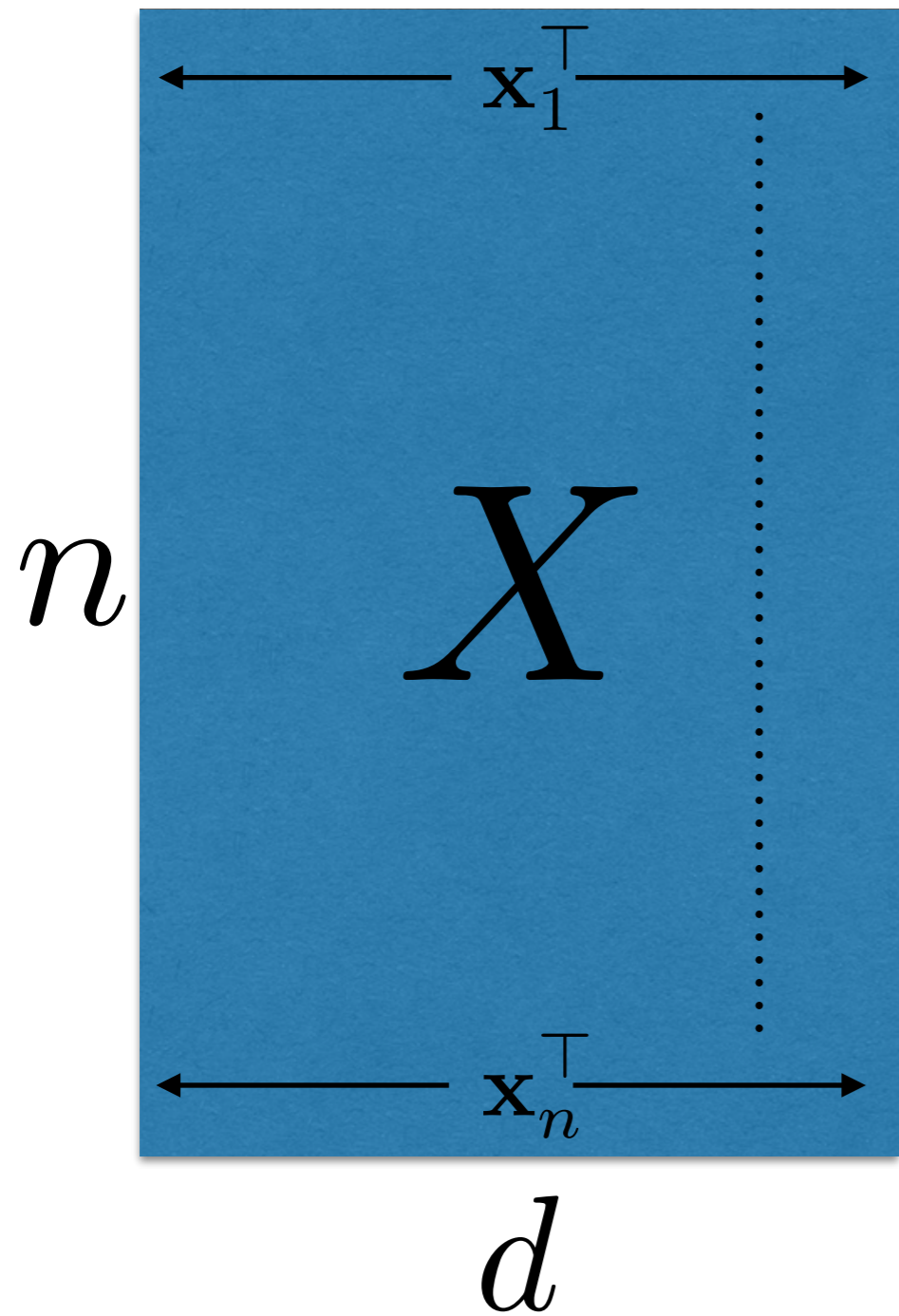
CCA DEMO REDO?

BACK TO SINGLE VIEW: RECAP

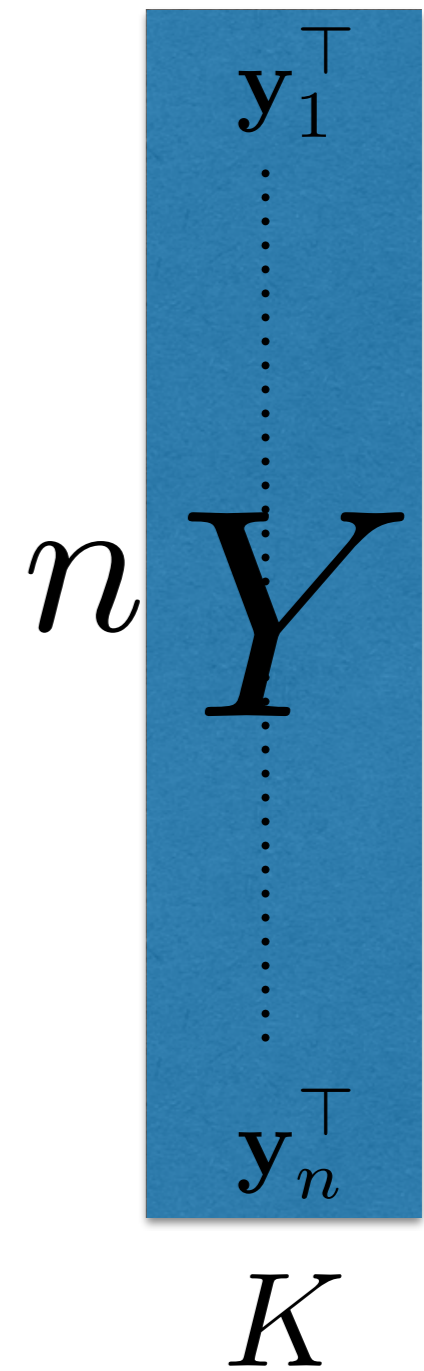
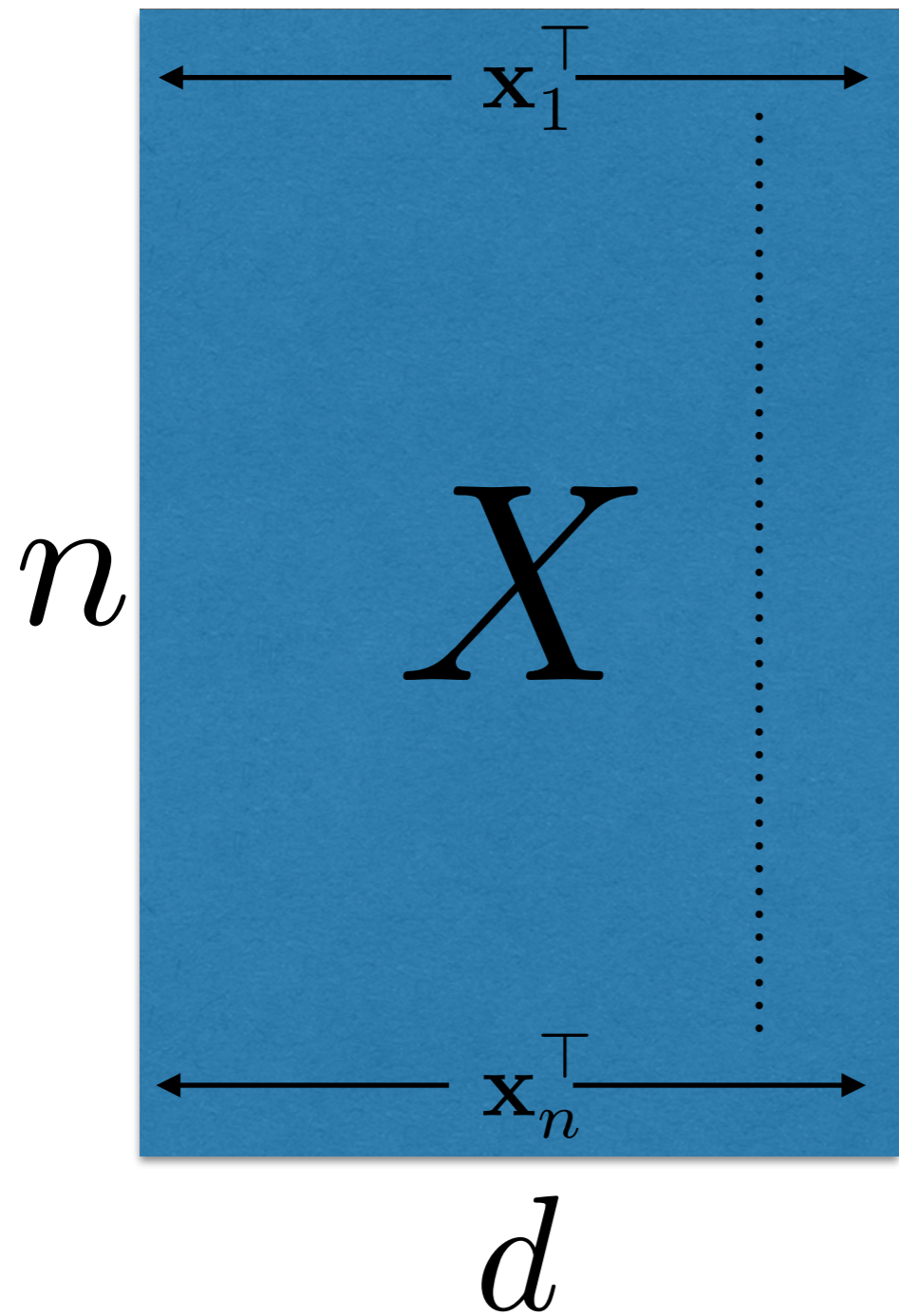
BACK TO SINGLE VIEW: RECAP



BACK TO SINGLE VIEW: RECAP



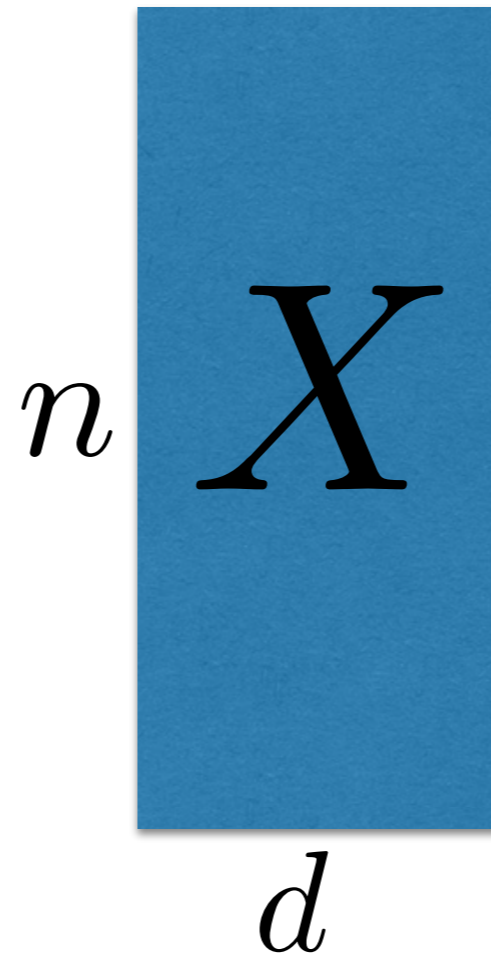
BACK TO SINGLE VIEW: RECAP



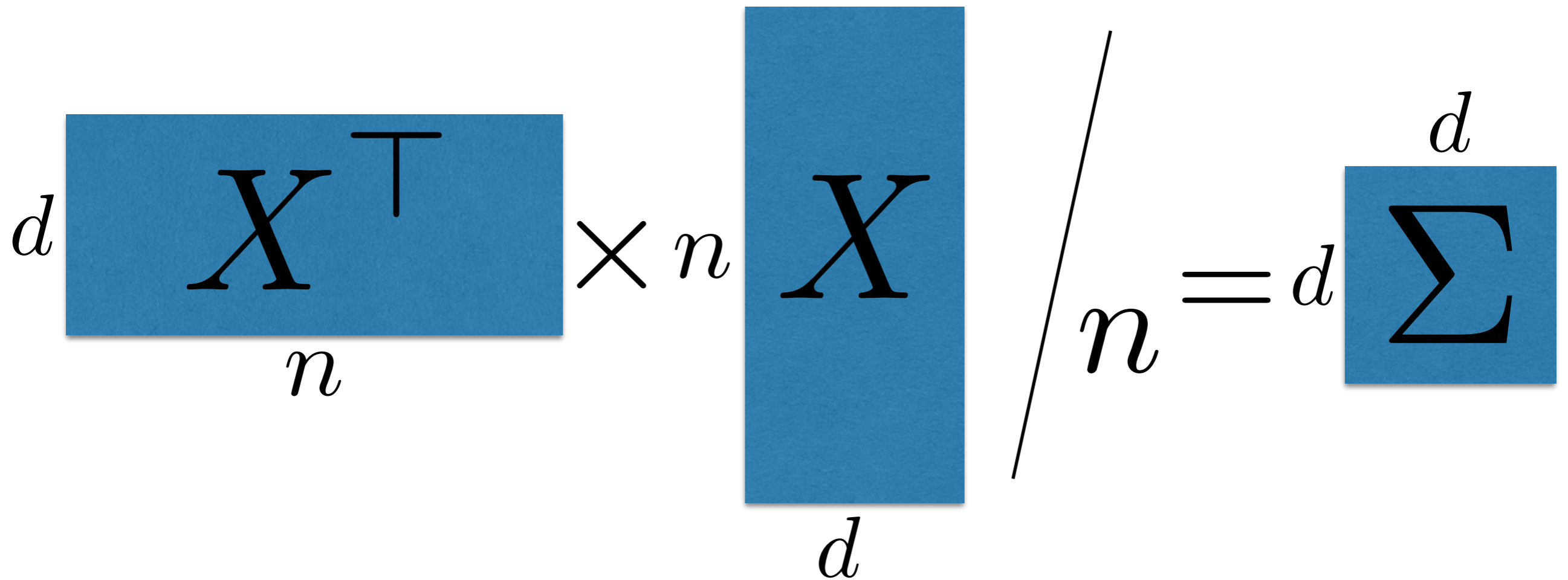
BACK TO SINGLE VIEW: RECAP

The diagram illustrates the matrix multiplication $X \times W = Y$. Matrix X is a large blue rectangle with height n and width d . Its rows are labeled x_1^\top at the top and x_n^\top at the bottom, with a vertical dotted line indicating intermediate rows. Matrix W is a smaller blue rectangle with height d and width K . Matrix Y is a tall blue rectangle with height n and width K , with its rows labeled y_1^\top at the top and y_n^\top at the bottom, also with a vertical dotted line. The multiplication is shown as $X \times W = Y$.

The Tall, THE FAT AND THE UGLY



The Tall, THE FAT AND THE UGLY


$$\begin{matrix} d \\ \text{---} \\ X^T \\ \text{---} \\ n \end{matrix} \times \begin{matrix} n \\ \text{---} \\ X \\ \text{---} \\ d \end{matrix} / n = \begin{matrix} d \\ \text{---} \\ \Sigma \\ \text{---} \\ d \end{matrix}$$

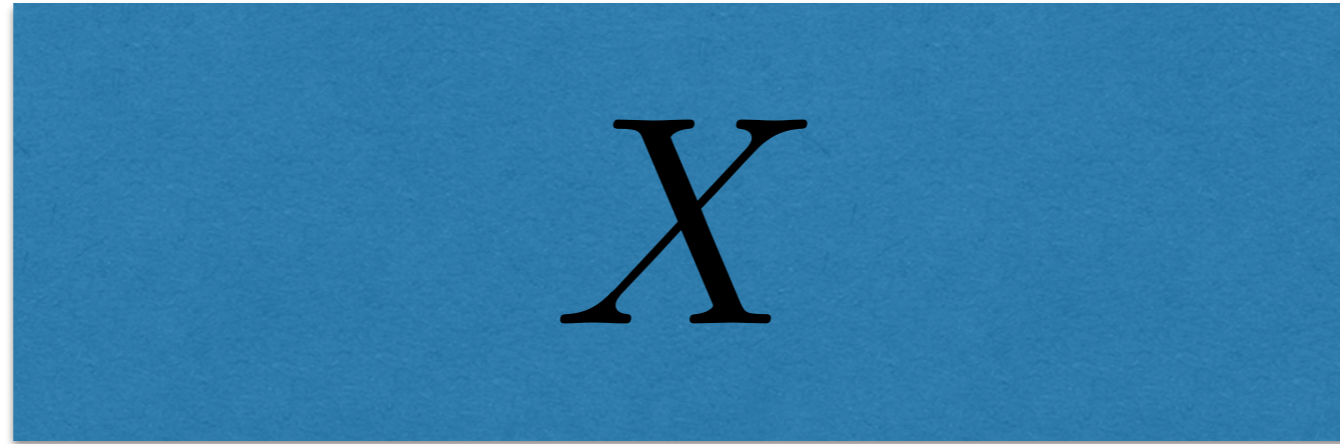
The Tall, THE FAT AND THE UGLY

$$\begin{array}{c} d \\ \times \\ n \end{array} X^T \times \begin{array}{c} n \\ \times \\ d \end{array} X \Big/ n = \begin{array}{c} d \\ \times \\ d \end{array} \Sigma$$

$$\begin{array}{c} d \\ \times \\ K \end{array} W = \text{Eigs} \left(\begin{array}{c} d \\ \times \\ d \end{array} \Sigma, K \right)$$

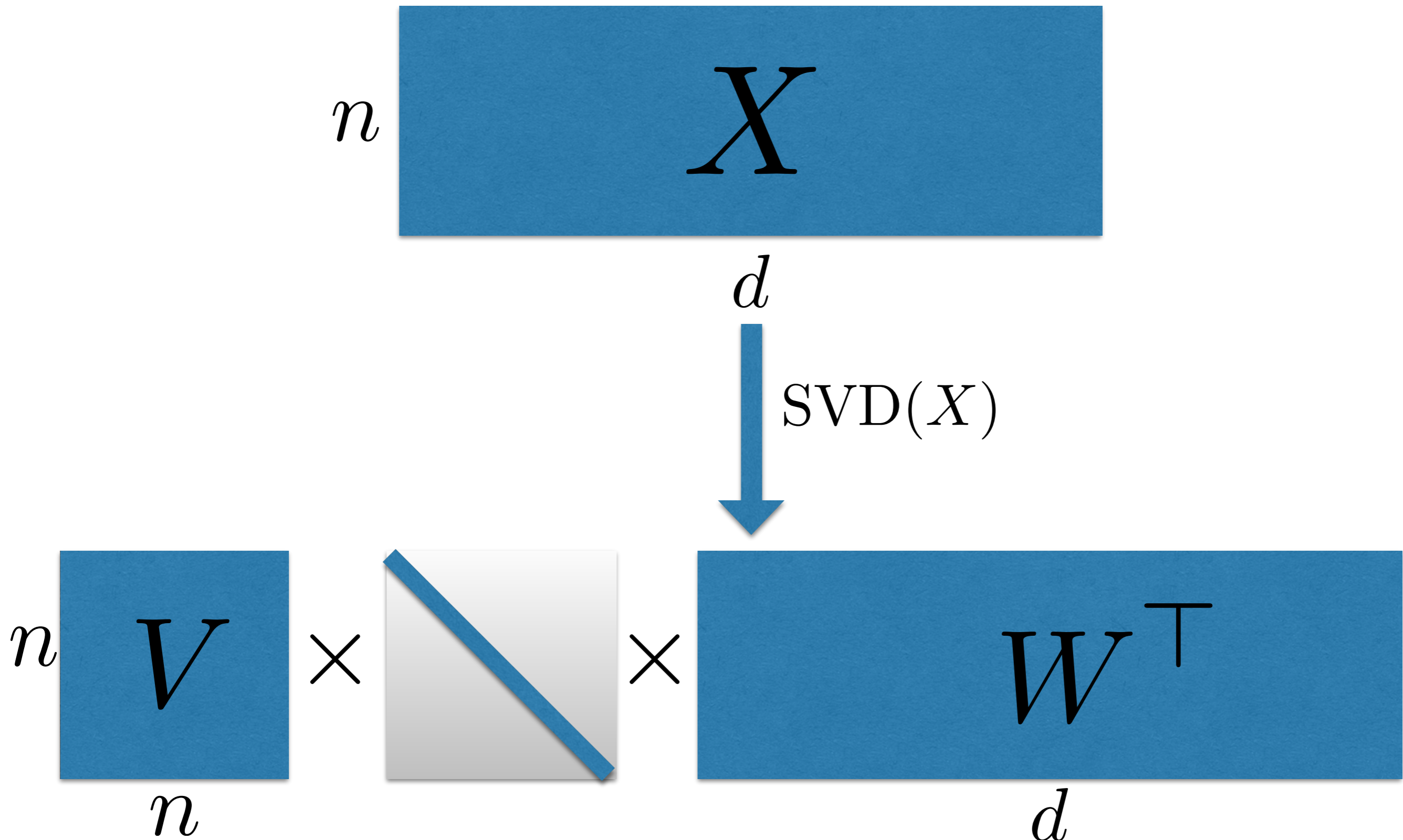
THE TALL, the Fat AND THE UGLY

n

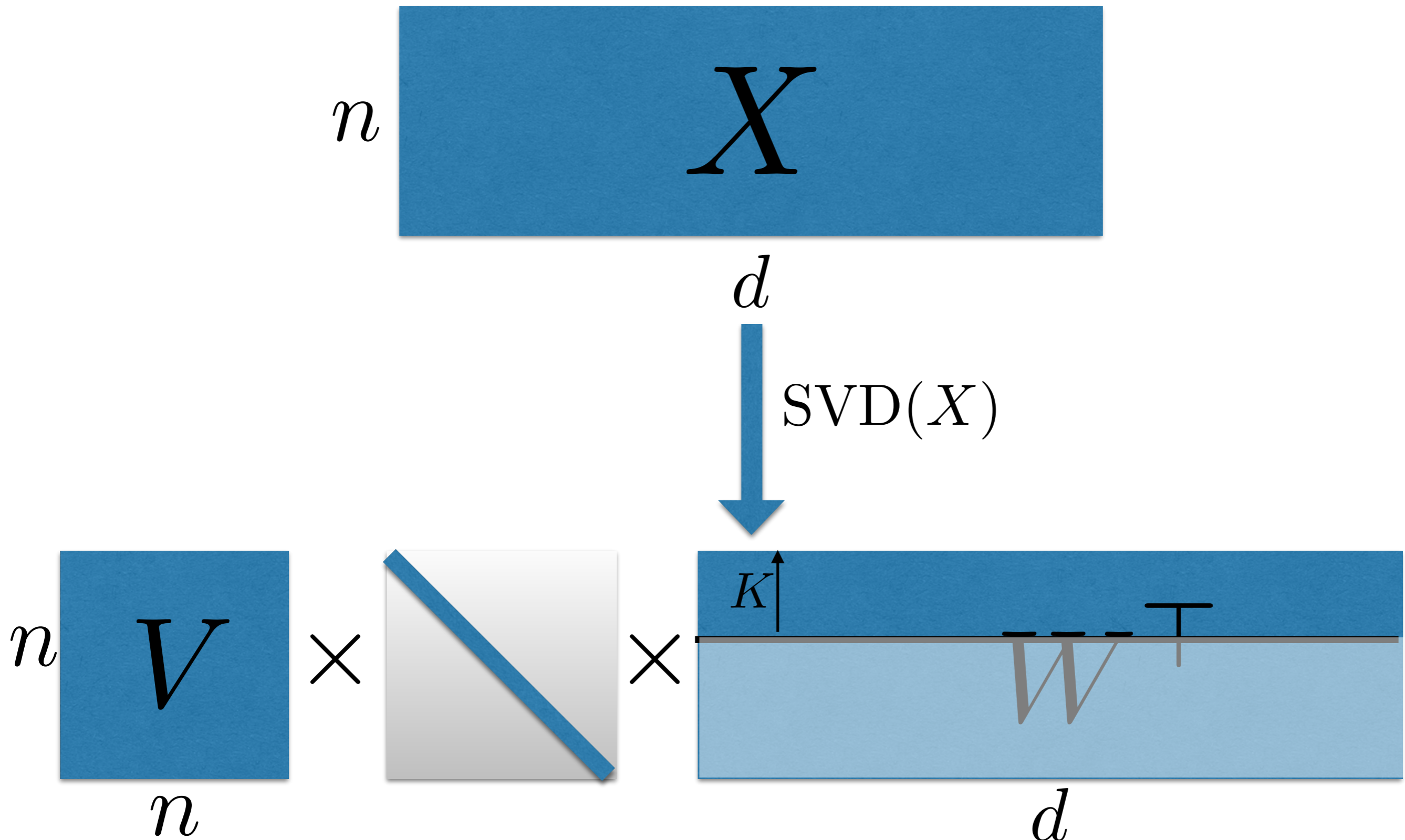


d

THE TALL, the Fat AND THE UGLY



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 $\text{size}(X) = n \times d$

I there any hope?

PICK A RANDOM W

PICK A RANDOM W

$$Y = X \times \left[\begin{array}{ccc} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \cdot & \\ & \cdot & \\ & \cdot & \\ +1 & \dots & -1 \end{array} \right] \Bigg/ \sqrt{K}$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

RANDOM PROJECTION

- What does “it works” even mean?

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.

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?!

Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = W^\top \tilde{\mathbf{x}}$. Note that

$$\tilde{\mathbf{y}}[j]^2 = \left(\sum_{i=1}^d W[i, j] \cdot \tilde{\mathbf{x}}[i] \right)^2$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = W^\top \tilde{\mathbf{x}}$. Note that

$$\tilde{\mathbf{y}}[j]^2 = \left(\sum_{i=1}^d W[i, j] \cdot \tilde{\mathbf{x}}[i] \right)^2 = \sum_{i, i'} (W[i, j] \cdot \tilde{\mathbf{x}}[i]) \cdot (W[i', j] \cdot \tilde{\mathbf{x}}[i'])$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = W^\top \tilde{\mathbf{x}}$. Note that

$$\begin{aligned}\tilde{\mathbf{y}}[j]^2 &= \left(\sum_{i=1}^d W[i, j] \cdot \tilde{\mathbf{x}}[i] \right)^2 = \sum_{i, i'} (W[i, j] \cdot \tilde{\mathbf{x}}[i]) \cdot (W[i', j] \cdot \tilde{\mathbf{x}}[i']) \\ &= \sum_{i, i'} (W[i, j] \cdot W[i', j]) \cdot (\tilde{\mathbf{x}}[i] \cdot \tilde{\mathbf{x}}[i'])\end{aligned}$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence,

$$\mathbb{E}[\tilde{\mathbf{y}}[j]^2] = \sum_{i,i'=1}^d \mathbb{E}[(W[i,j] \cdot W[i',j])] \cdot (\tilde{\mathbf{x}}[i] \cdot \tilde{\mathbf{x}}[i'])$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence,

$$\mathbb{E}[\tilde{\mathbf{y}}[j]^2] = \sum_{i,i'=1}^d \mathbb{E}[(W[i,j] \cdot W[i',j])] \cdot (\tilde{\mathbf{x}}[i] \cdot \tilde{\mathbf{x}}[i'])$$

if $i \neq i'$, $W[i,j]$ and $W[i',j]$ are independent and so

$$= \sum_{i=1}^d \mathbb{E}[(W[i,j]^2)] \tilde{\mathbf{x}}[i]^2 + \sum_{i \neq i'} (\mathbb{E}[W[i,j]] \cdot \mathbb{E}[W[i',j]]) \cdot (\tilde{\mathbf{x}}[i] \cdot \tilde{\mathbf{x}}[i'])$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence,

$$\mathbb{E}[\tilde{\mathbf{y}}[j]^2] = \sum_{i,i'=1}^d \mathbb{E}[(W[i,j] \cdot W[i',j])] \cdot (\tilde{\mathbf{x}}[i] \cdot \tilde{\mathbf{x}}[i'])$$

if $i \neq i'$, $W[i,j]$ and $W[i',j]$ are independent and so

$$\begin{aligned} &= \sum_{i=1}^d \mathbb{E}[(W[i,j]^2)] \tilde{\mathbf{x}}[i]^2 + \sum_{i \neq i'} (\mathbb{E}[W[i,j]] \cdot \mathbb{E}[W[i',j]]) \cdot (\tilde{\mathbf{x}}[i] \cdot \tilde{\mathbf{x}}[i']) \\ &= \sum_{i=1}^d \tilde{\mathbf{x}}[i]^2 / \sqrt{K}^2 = \|\tilde{\mathbf{x}}\|_2^2 / K \end{aligned}$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence,

$$\mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2] = \sum_{j=1}^K \mathbb{E}[\tilde{\mathbf{y}}[j]^2] = \sum_{j=1}^K \|\tilde{\mathbf{x}}\|_2^2 / K = \|\tilde{\mathbf{x}}\|_2^2$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence,

$$\mathbb{E}[\|\tilde{\mathbf{y}}\|_2^2] = \sum_{j=1}^K \mathbb{E}[\tilde{\mathbf{y}}[j]^2] = \sum_{j=1}^K \|\tilde{\mathbf{x}}\|_2^2 / K = \|\tilde{\mathbf{x}}\|_2^2$$

If we let $\tilde{\mathbf{x}} = \mathbf{x}_s - \mathbf{x}_t$ then

$$\tilde{\mathbf{y}} = W^\top \tilde{\mathbf{x}} = W^\top \mathbf{x}_s - W^\top \mathbf{x}_t = \mathbf{y}_s - \mathbf{y}_t$$

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence,

$$\mathbb{E}\left[\|\tilde{\mathbf{y}}\|_2^2\right] = \sum_{j=1}^K \mathbb{E}[\tilde{\mathbf{y}}[j]^2] = \sum_{j=1}^K \|\tilde{\mathbf{x}}\|_2^2 / K = \|\tilde{\mathbf{x}}\|_2^2$$

If we let $\tilde{\mathbf{x}} = \mathbf{x}_s - \mathbf{x}_t$ then

$$\tilde{\mathbf{y}} = W^\top \tilde{\mathbf{x}} = W^\top \mathbf{x}_s - W^\top \mathbf{x}_t = \mathbf{y}_s - \mathbf{y}_t$$

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

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

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Hence,

$$\mathbb{E}\left[\|\tilde{\mathbf{y}}\|_2^2\right] = \sum_{j=1}^K \mathbb{E}[\tilde{\mathbf{y}}[j]^2] = \sum_{j=1}^K \|\tilde{\mathbf{x}}\|_2^2 / K = \|\tilde{\mathbf{x}}\|_2^2$$

If we let $\tilde{\mathbf{x}} = \mathbf{x}_s - \mathbf{x}_t$ then

$$\tilde{\mathbf{y}} = W^\top \tilde{\mathbf{x}} = W^\top \mathbf{x}_s - W^\top \mathbf{x}_t = \mathbf{y}_s - \mathbf{y}_t$$

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

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

Lets try this in Matlab ...

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

For large K , not only true in expectation but also with high probability

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

For large K , not only true in expectation but also with high probability

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 $i, j \in \{1, \dots, n\}$,

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

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

For large K , not only true in expectation but also with high probability

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 $i, j \in \{1, \dots, n\}$,

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

Lets try on Matlab ...

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

For large K , not only true in expectation but also with high probability

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 $i, j \in \{1, \dots, n\}$,

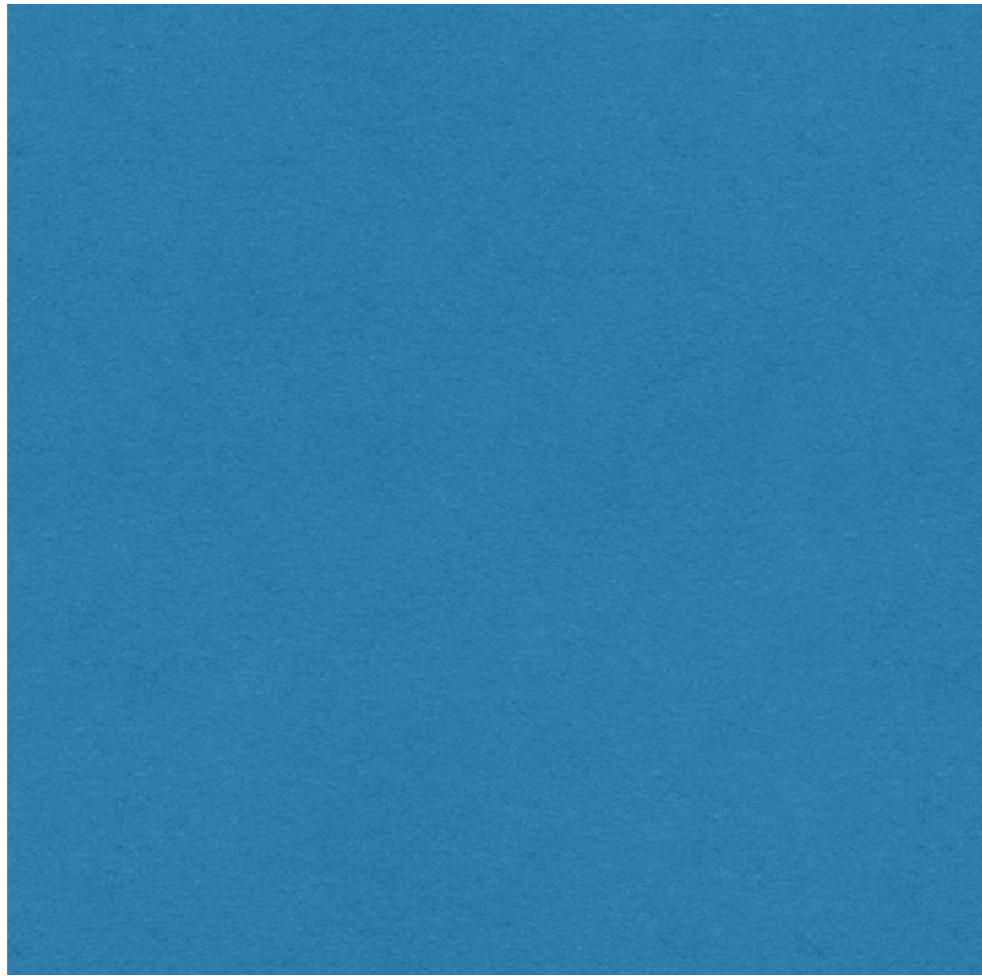
$$(1 - \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2 \leq \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq (1 + \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2$$

Lets try on Matlab ...

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

WHY IS THIS SO RIDICULOUSLY MAGICAL?

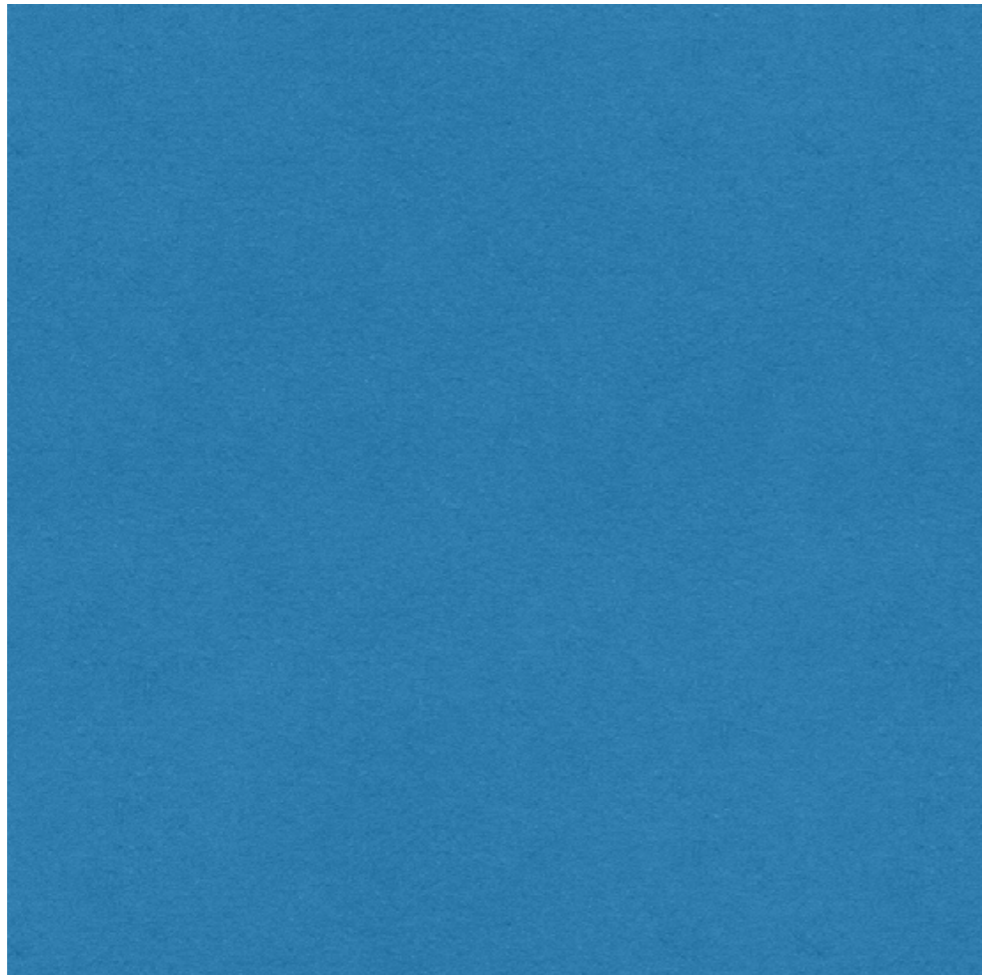
$n =$
1000



$d = 1000$

WHY IS THIS SO RIDICULOUSLY MAGICAL?

$n =$
1000

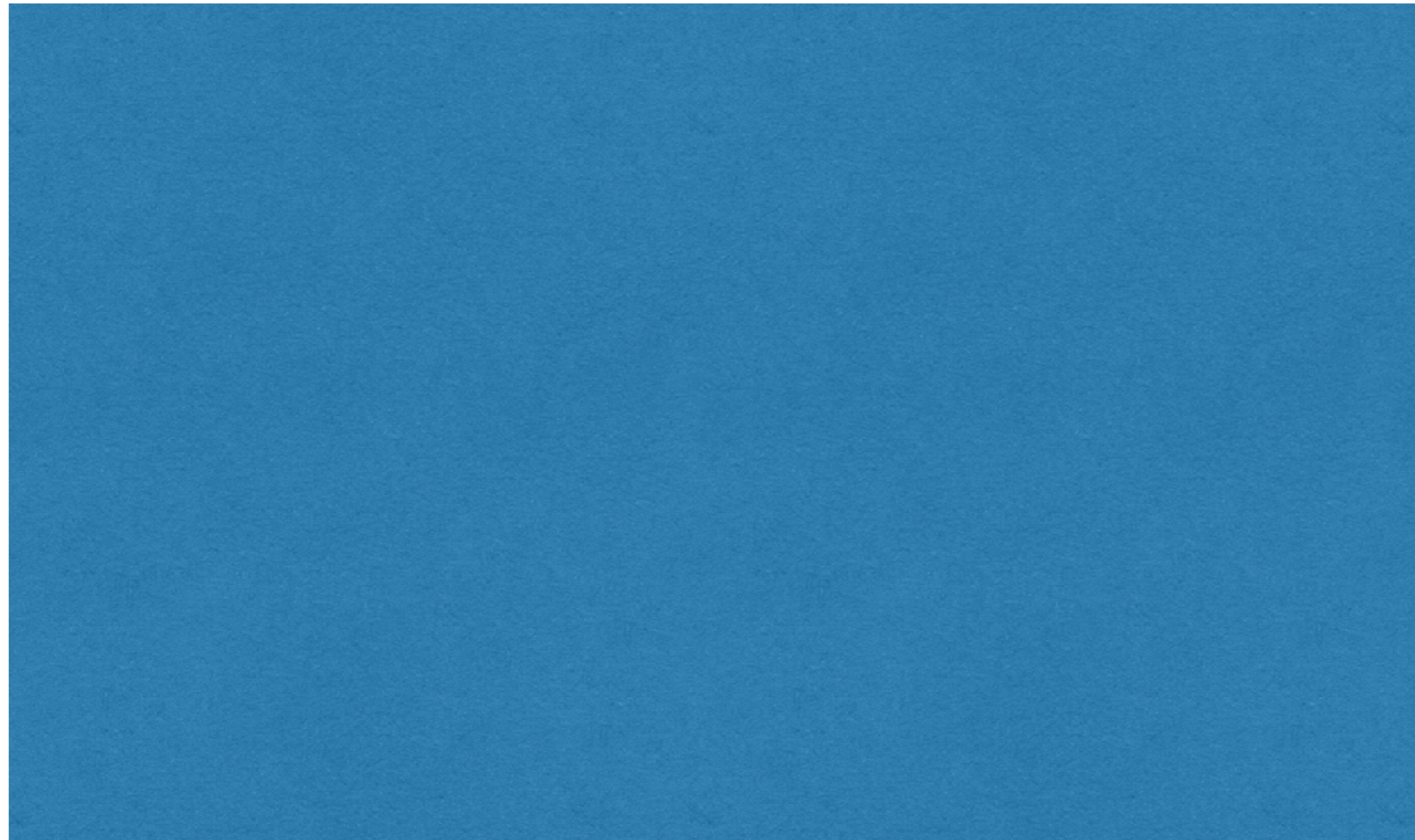


$d = 1000$

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