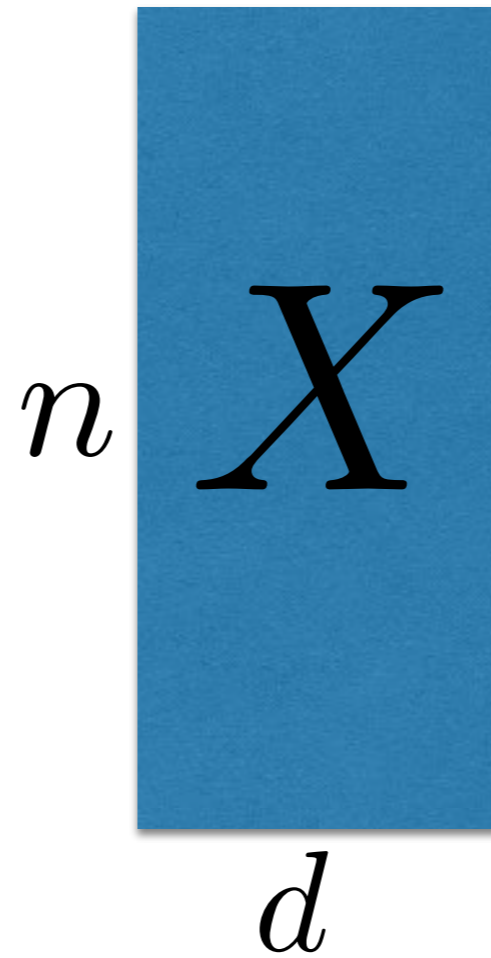


Machine Learning for Data Science (CS4786)

Lecture 5

Random Projections & Canonical Correlation Analysis

The Tall, THE FAT AND THE UGLY



The Tall, THE FAT AND THE UGLY

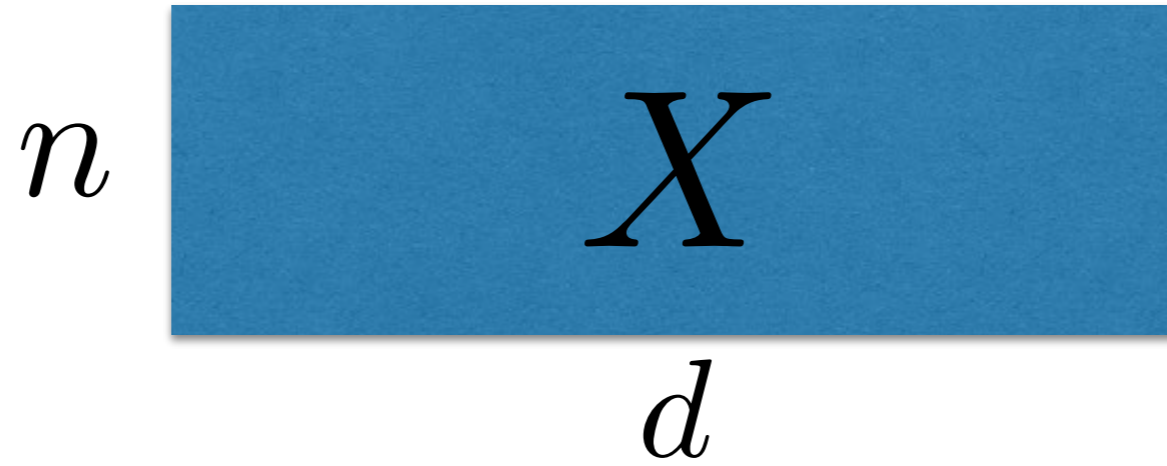
$$\begin{matrix} d \\ \text{---} \\ X^T \\ \text{---} \\ n \end{matrix} \times \begin{matrix} n \\ \text{---} \\ X \\ \text{---} \\ d \end{matrix} \Big/ n = \begin{matrix} d \\ \text{---} \\ \Sigma \\ \text{---} \\ d \end{matrix}$$

The Tall, THE FAT AND THE UGLY

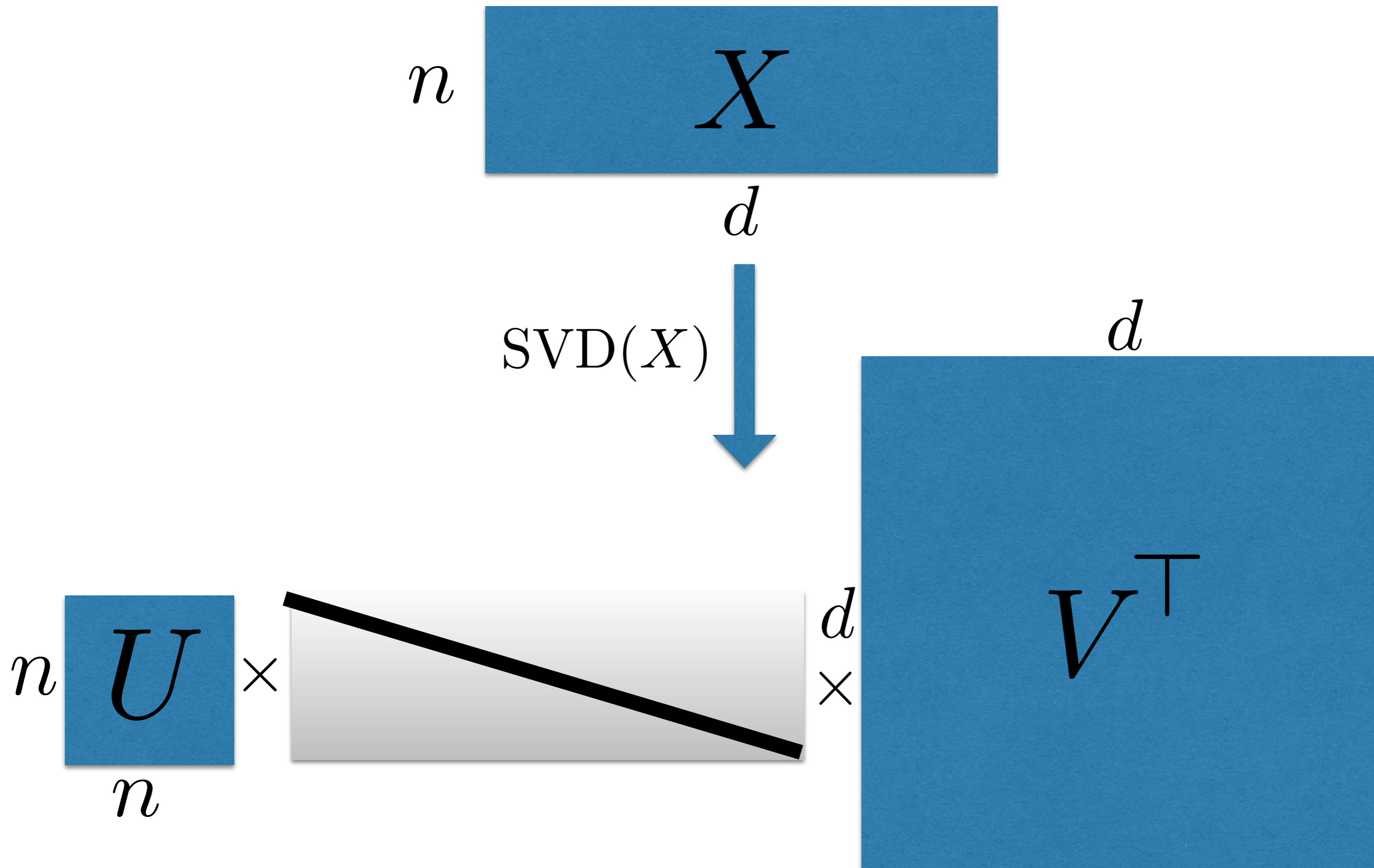
$$\begin{array}{c} d \\ \times \\ n \end{array} X^T \times \begin{array}{c} n \\ \times \\ d \end{array} X \Big/ n = \begin{array}{c} d \\ \times \\ d \end{array} \Sigma$$

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

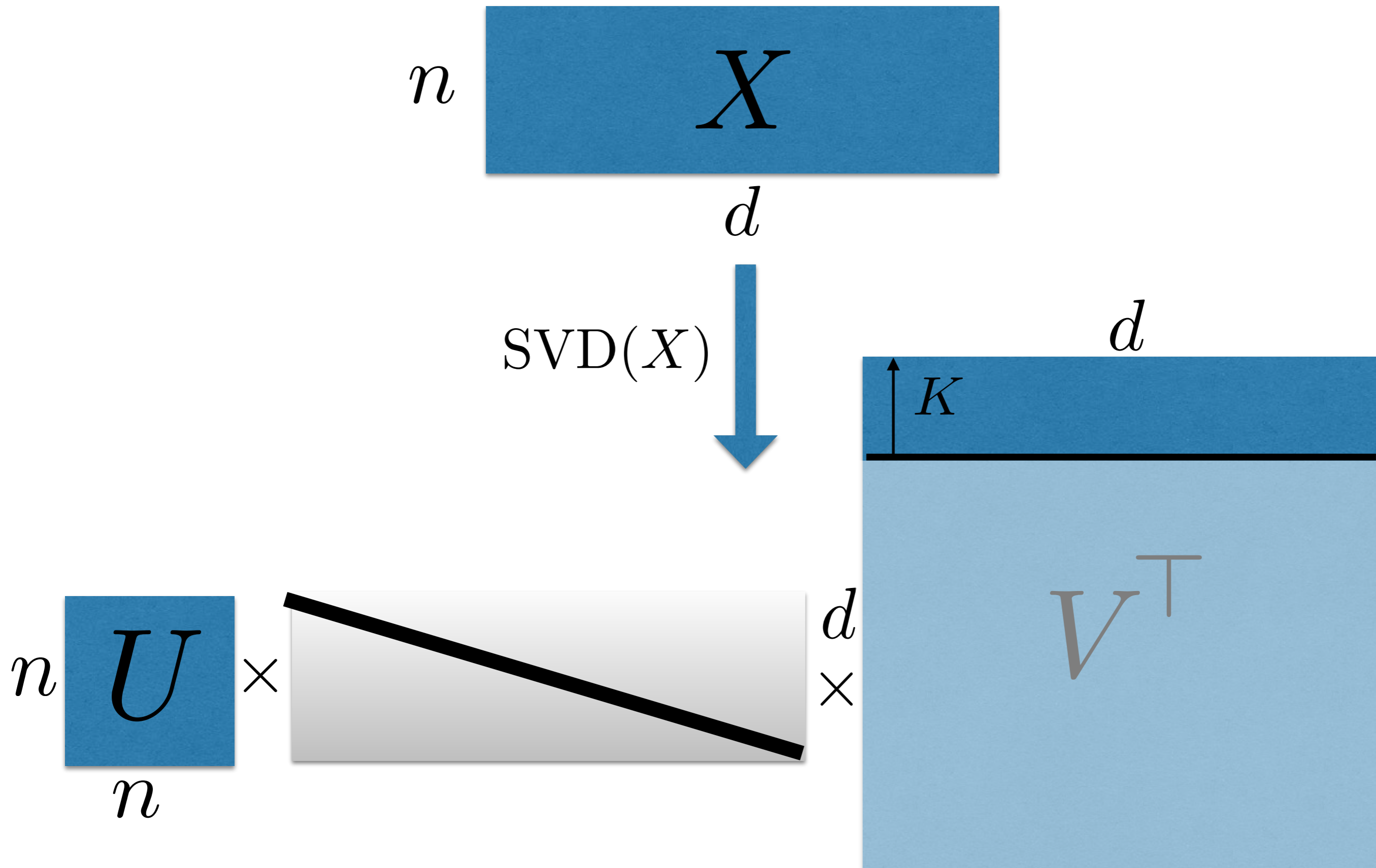
THE TALL, the Fat AND THE UGLY



THE TALL, the Fat AND THE UGLY

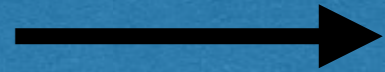


THE TALL, the Fat AND THE UGLY



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?!

- 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. $y_t - y_s$?

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

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 ...

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

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?!

- Like repeating the experiment K times and averaging

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

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

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

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

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

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

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

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

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

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

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

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

Lets try ...

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

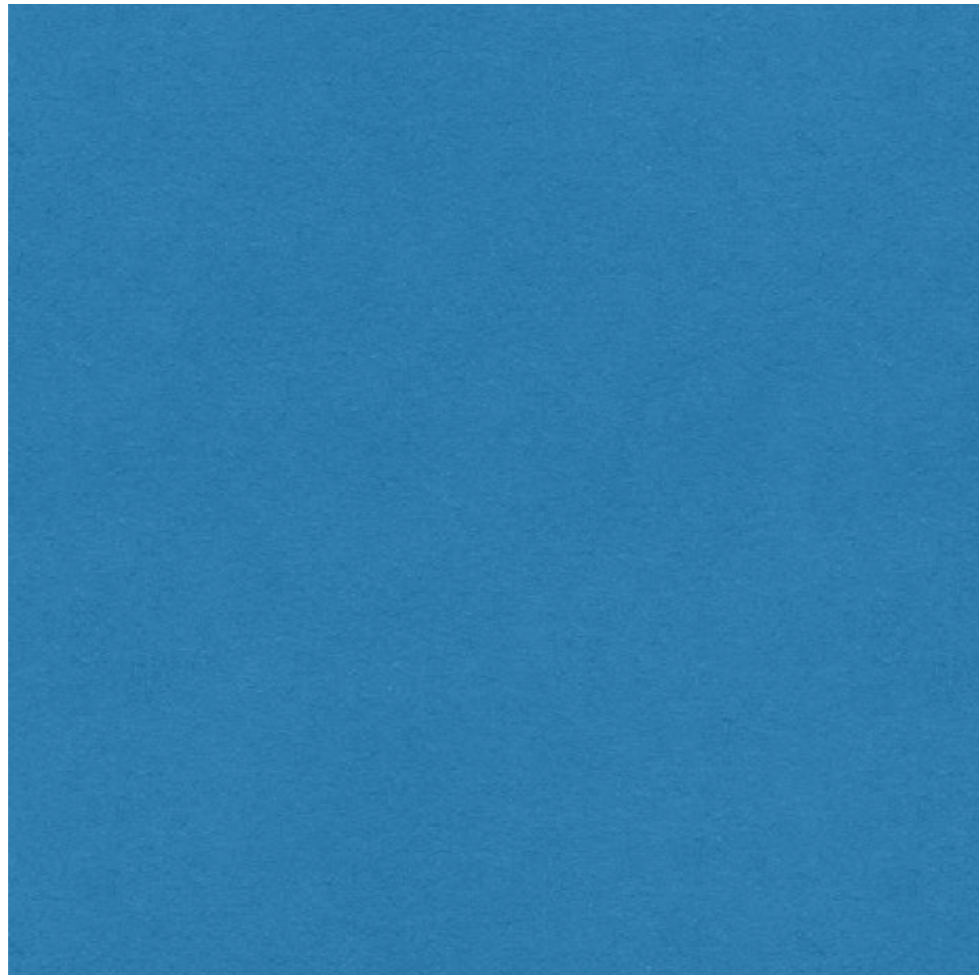
$$(1 - \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2^2 \leq \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq (1 + \epsilon) \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$$

Lets try ...

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

WHY IS THIS SO RIDICULOUSLY MAGICAL?

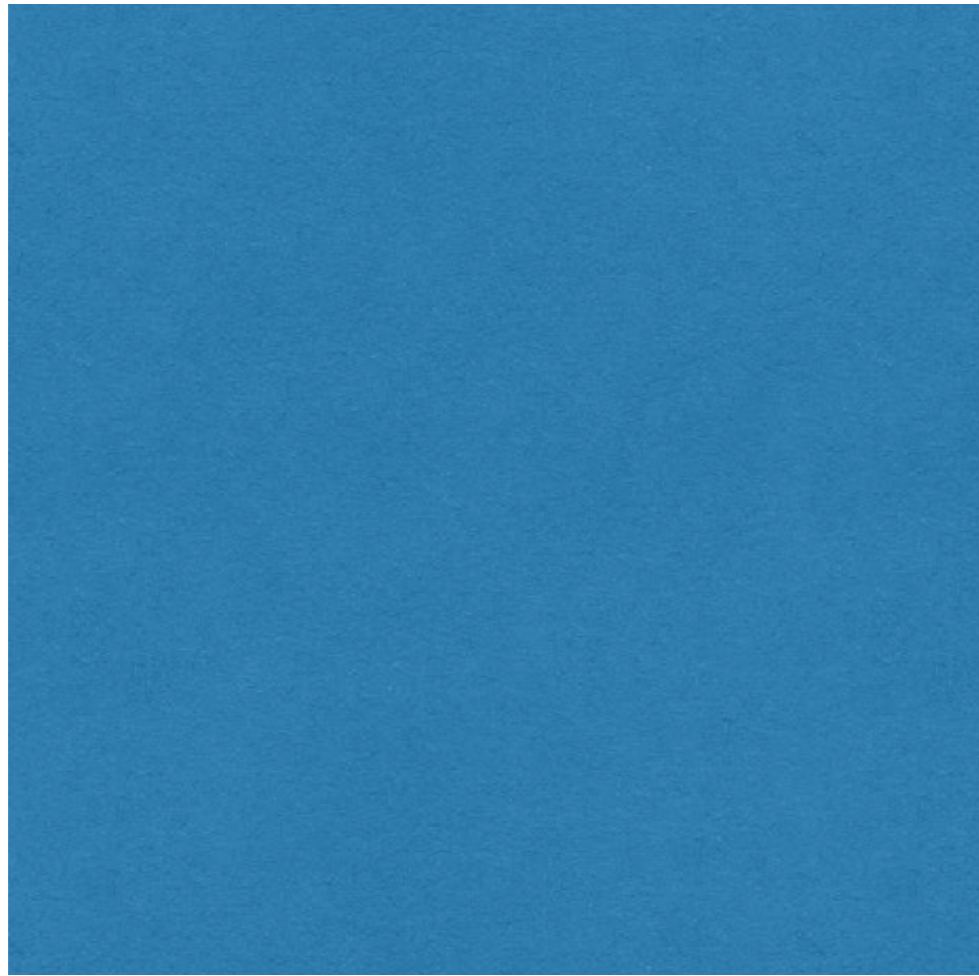
$n =$
1000



$d = 1000$

WHY IS THIS SO RIDICULOUSLY MAGICAL?

$n =$
1000

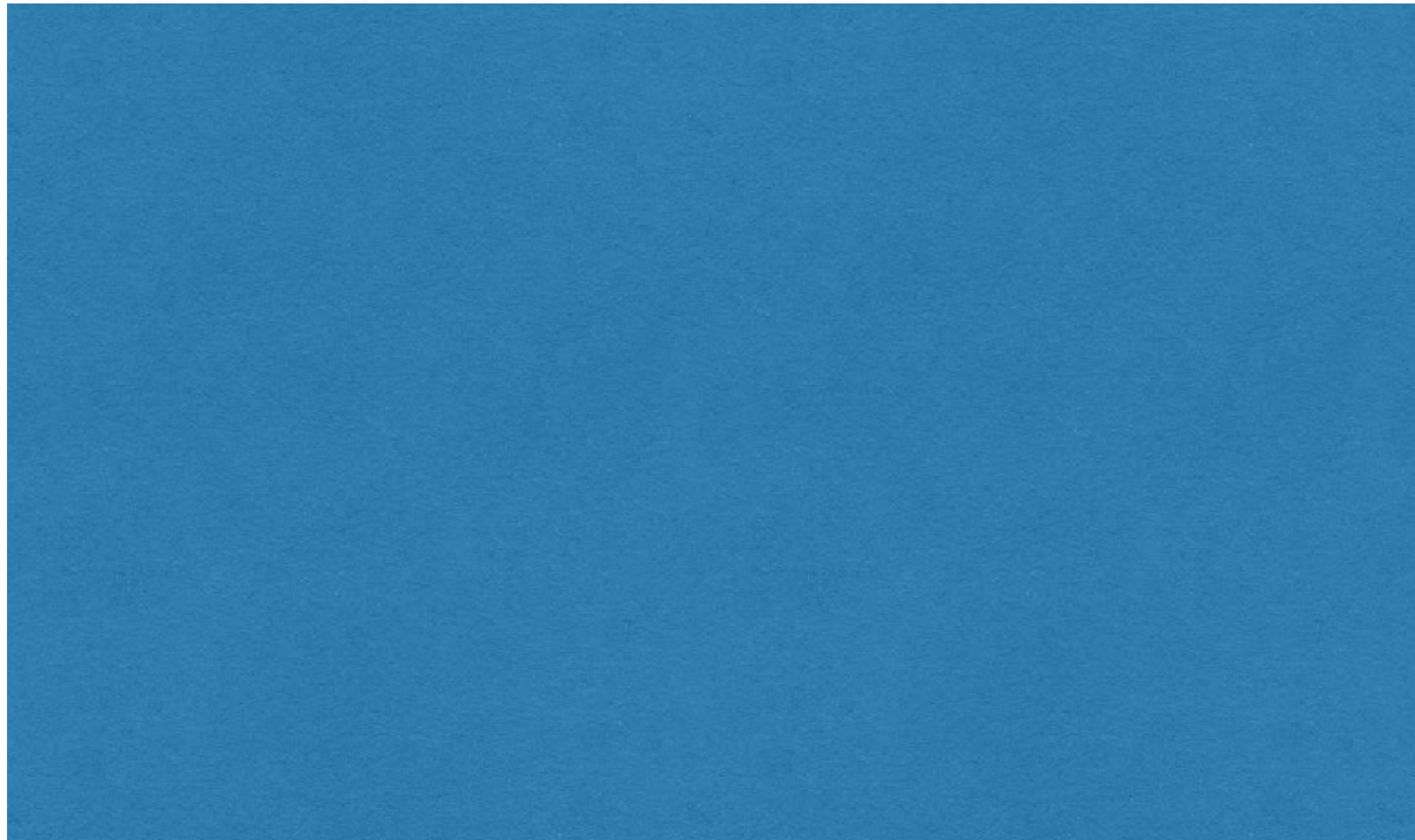


$d = 1000$

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

TWO VIEW DIMENSIONALITY REDUCTION

- Data comes in pairs $(\mathbf{x}_1, \mathbf{x}'_1), \dots, (\mathbf{x}_n, \mathbf{x}'_n)$ where \mathbf{x}_t 's are d dimensional and \mathbf{x}'_t 's are d' dimensional
- Goal: Compress say view one into $\mathbf{y}_1, \dots, \mathbf{y}_n$, that are K dimensional vectors
 - Retain information redundant between the two views
 - Eliminate “noise” specific to only one of the views

Canonical Correlation Analysis

Analysis



Canonical Correlation Analysis

Analysis

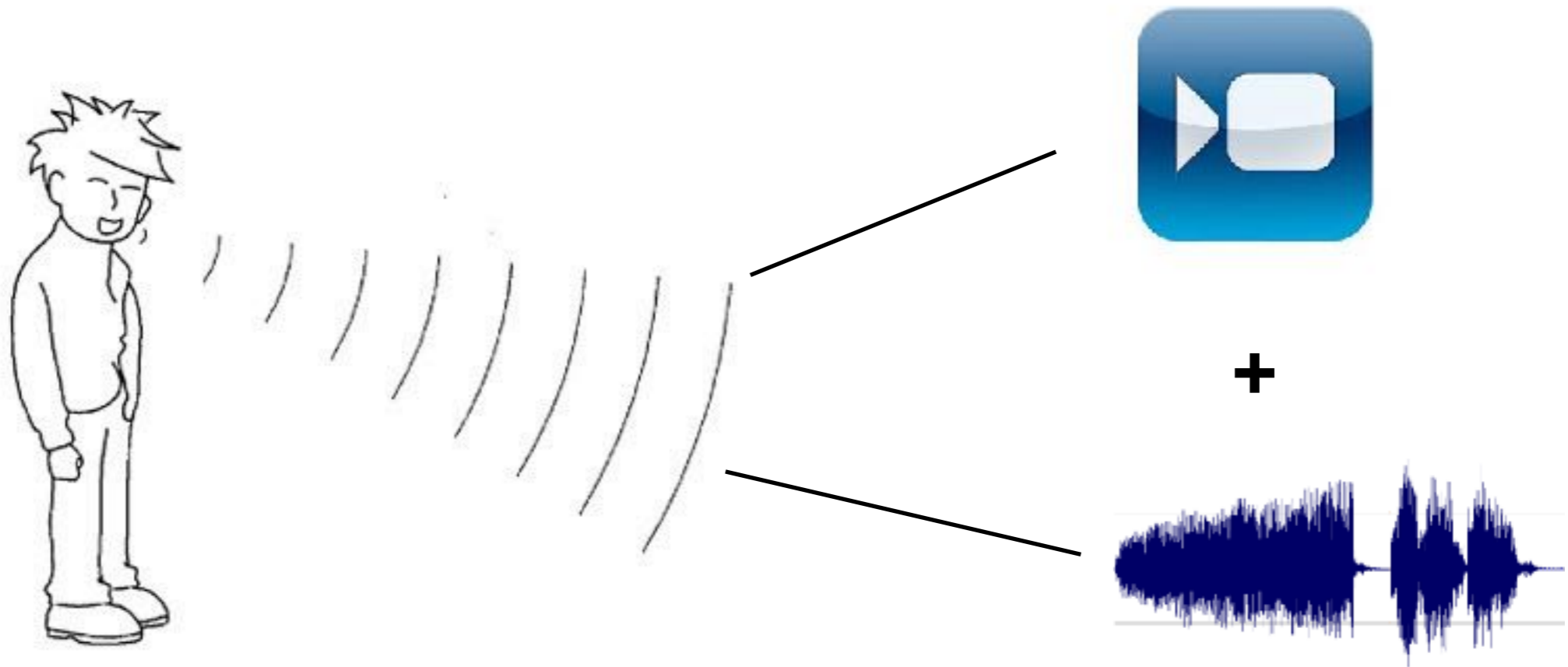


Age

+ Gender

Candies per week

EXAMPLE I: SPEECH RECOGNITION



- Audio might have background sounds uncorrelated with video
- Video might have lighting changes uncorrelated with audio
- Redundant information between two views: the speech

EXAMPLE II: COMBINING FEATURE EXTRACTIONS

- Method A and Method B are both equally good feature extraction techniques
- Concatenating the two features blindly yields large dimensional feature vector with redundancy
- Applying techniques like CCA extracts the key information between the two methods
- Removes extra unwanted information

How do we get the right direction? (say $K = 1$)



How do we get the right direction? (say $K = 1$)

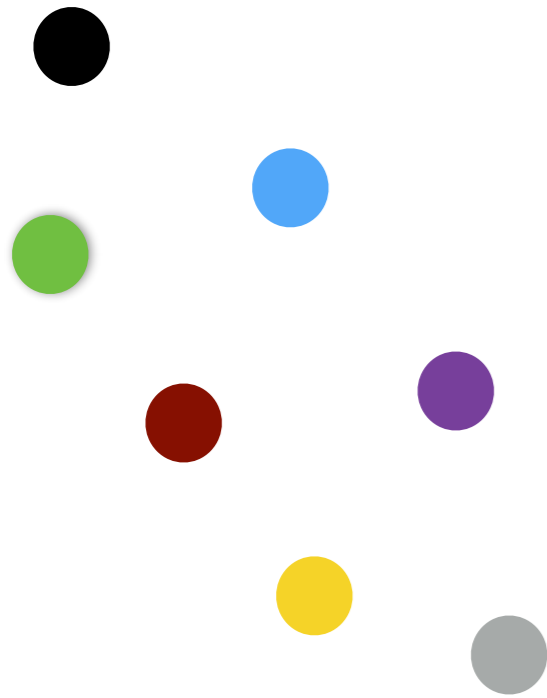


Age

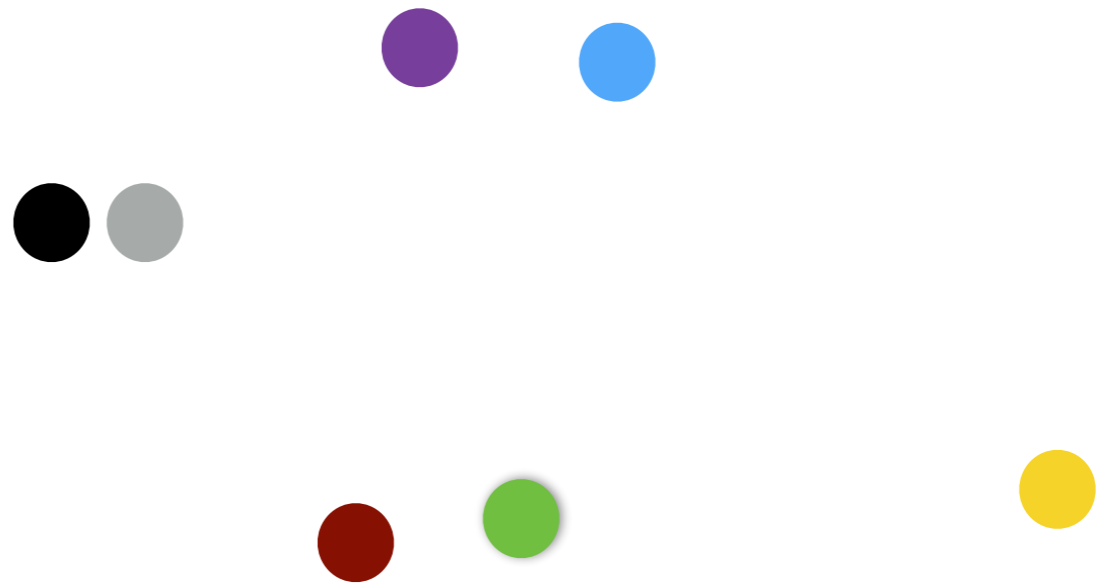
+ Gender

Candies per week

WHICH DIRECTION TO PICK?

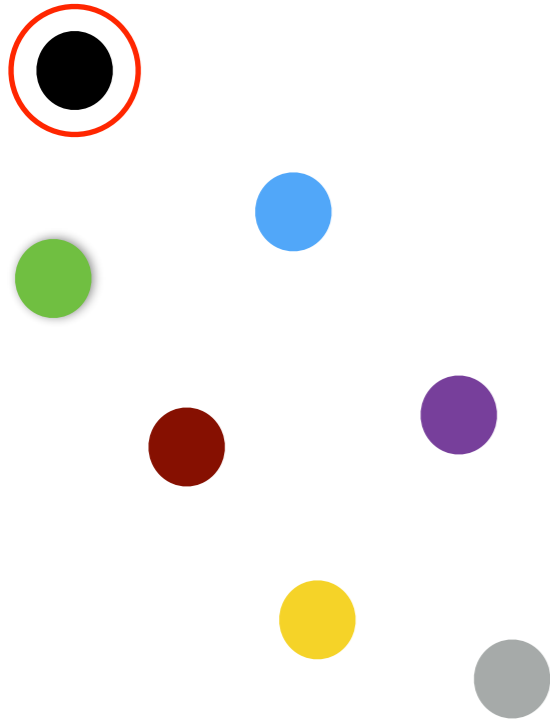


View I

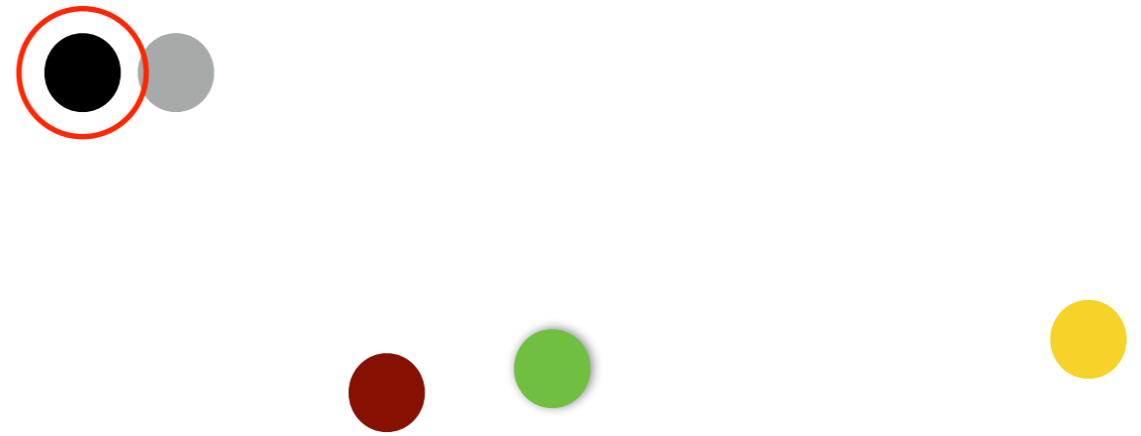


View II

WHICH DIRECTION TO PICK?

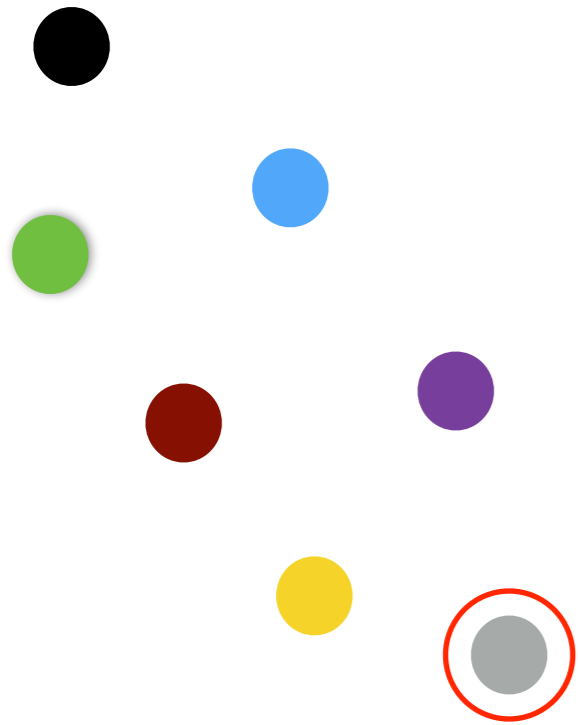


View I

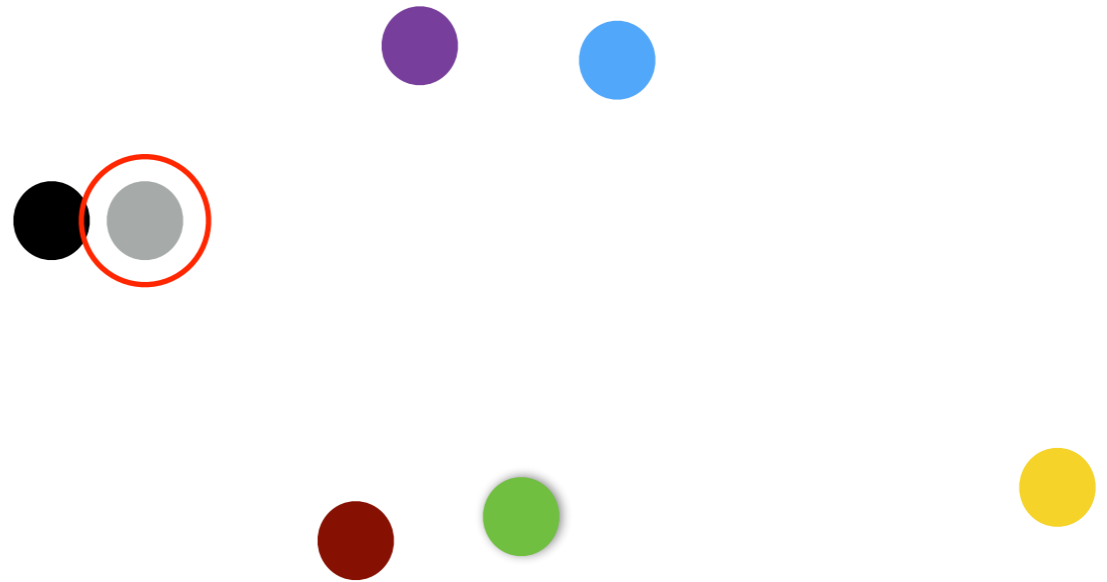


View II

WHICH DIRECTION TO PICK?

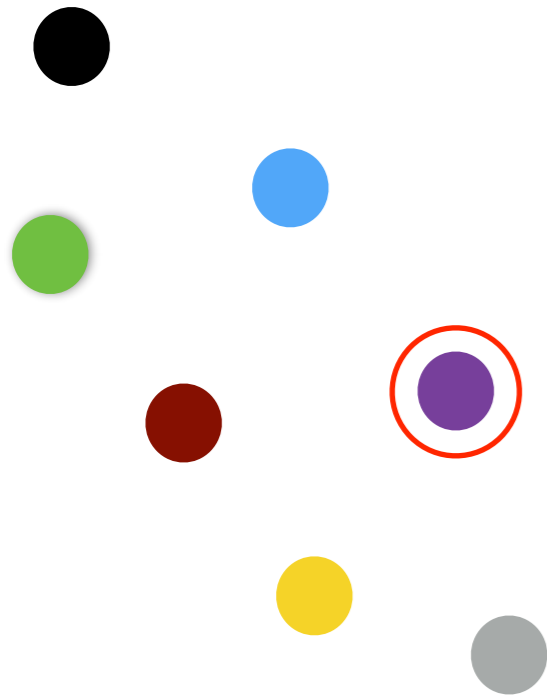


View I

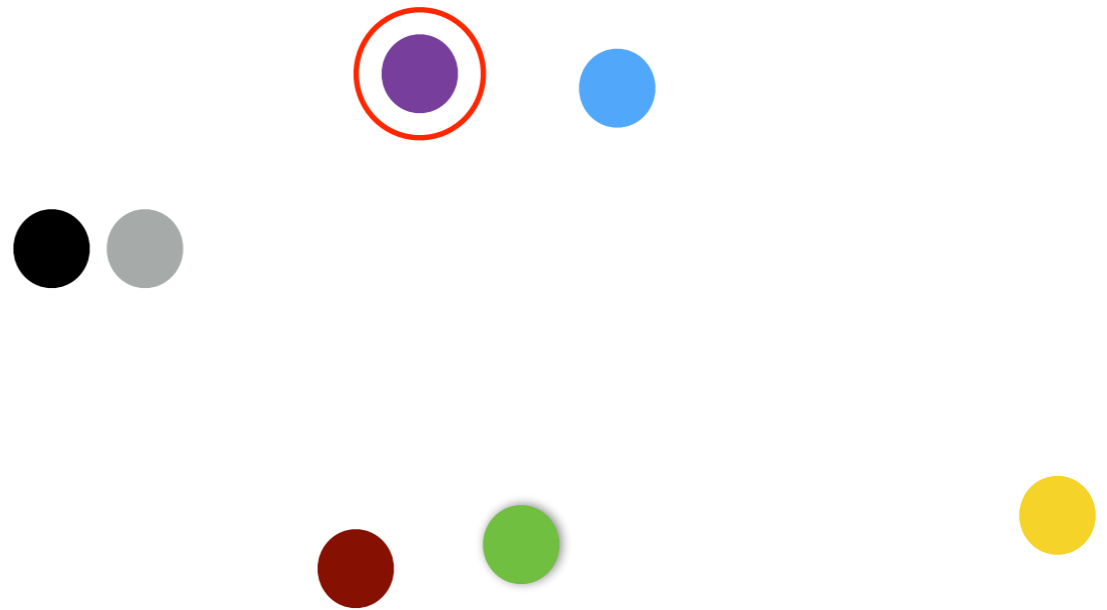


View II

WHICH DIRECTION TO PICK?



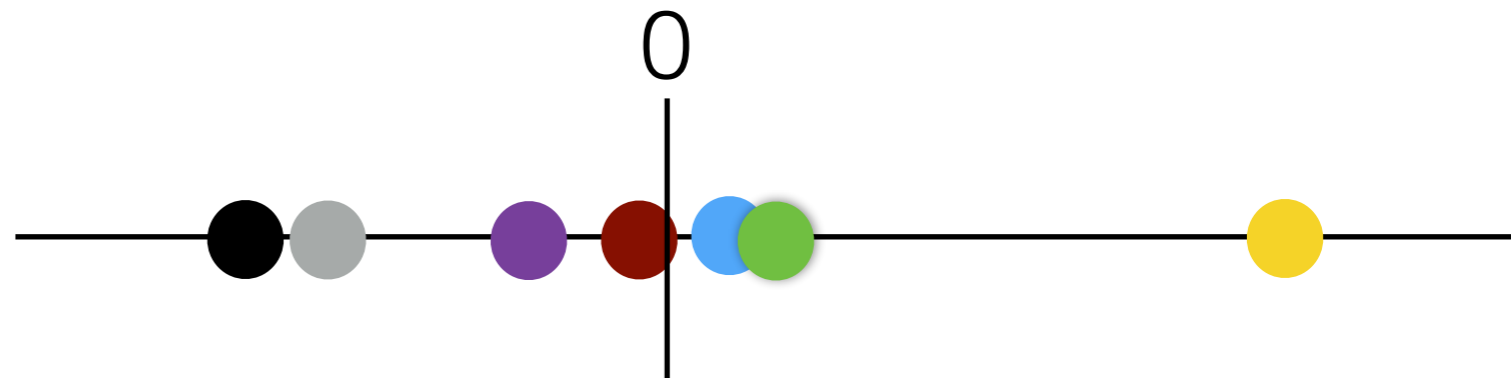
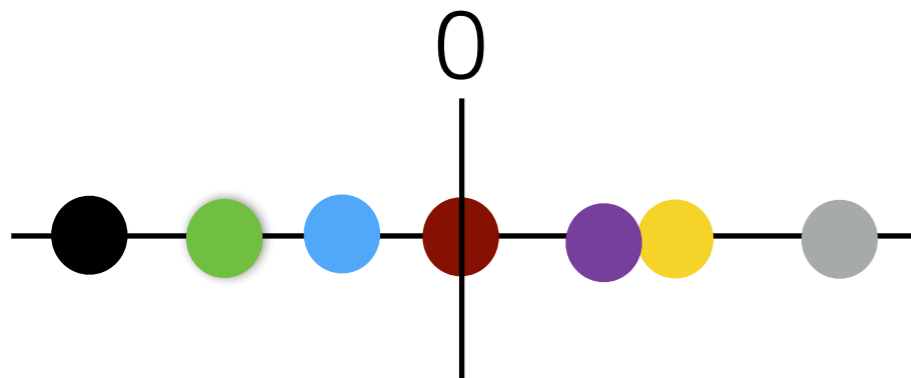
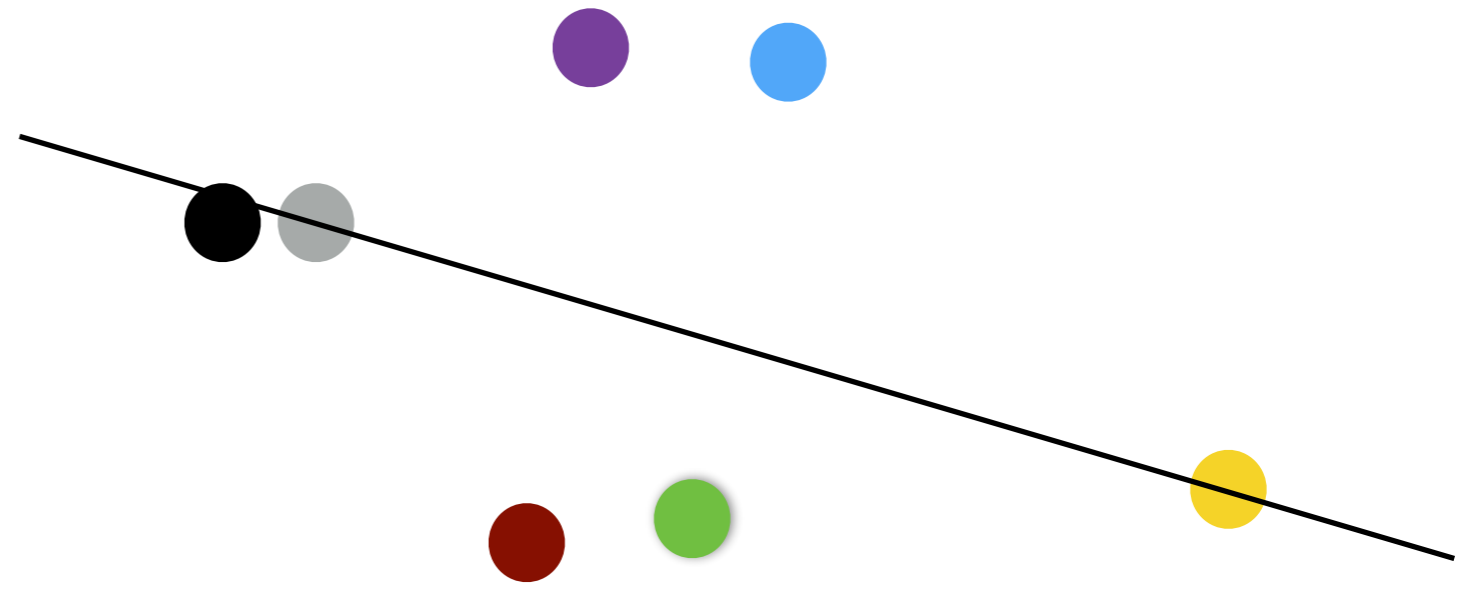
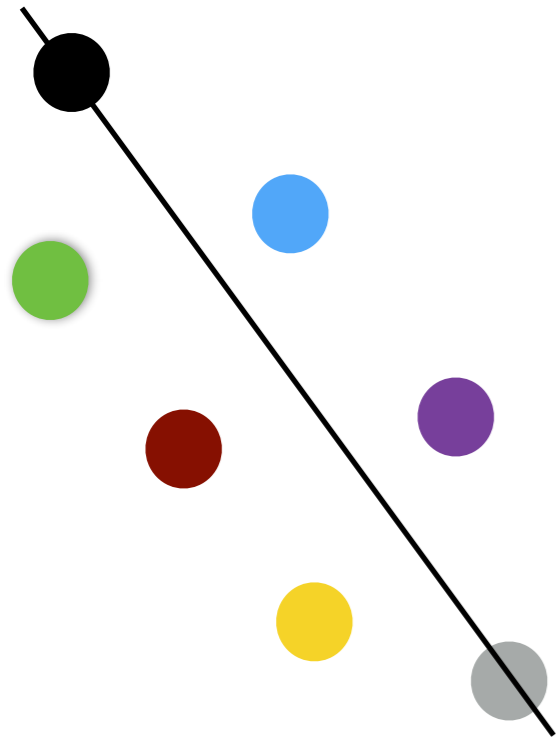
View I



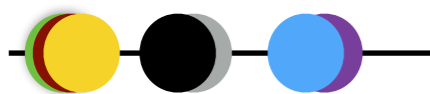
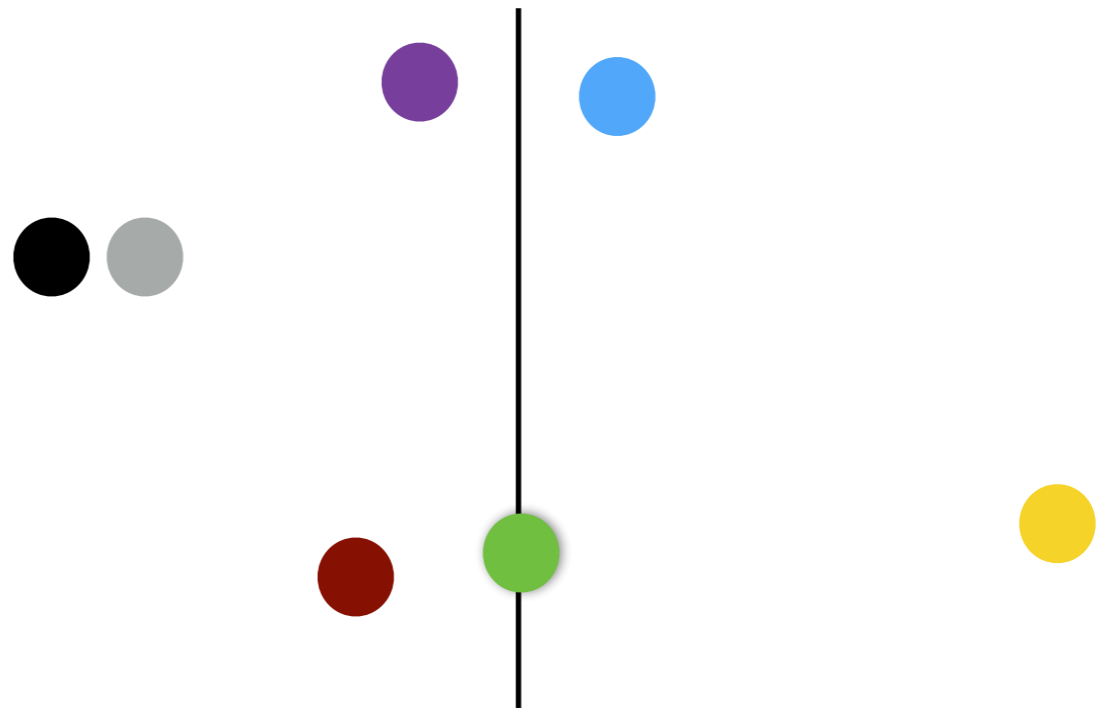
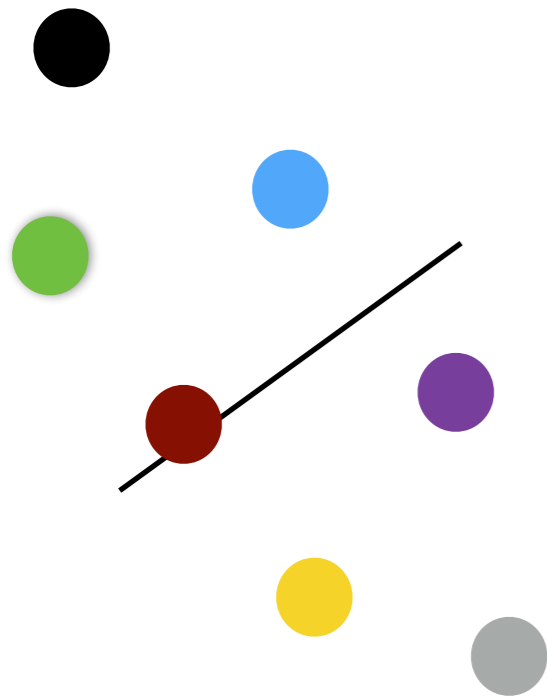
View II

WHICH DIRECTION TO PICK?

PCA direction



WHICH DIRECTION TO PICK?



Direction has large covariance

How do we pick the right direction to project to?