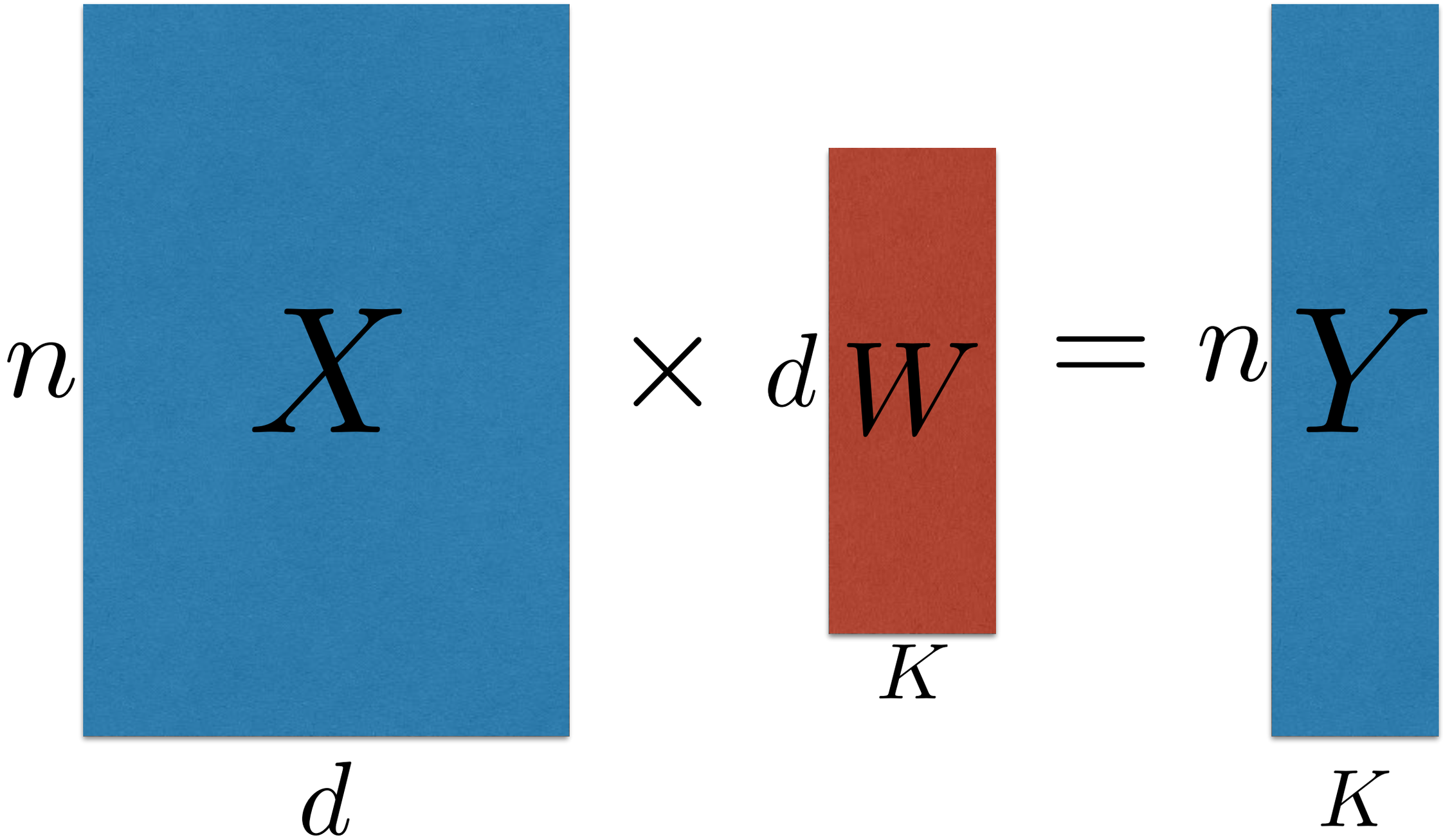
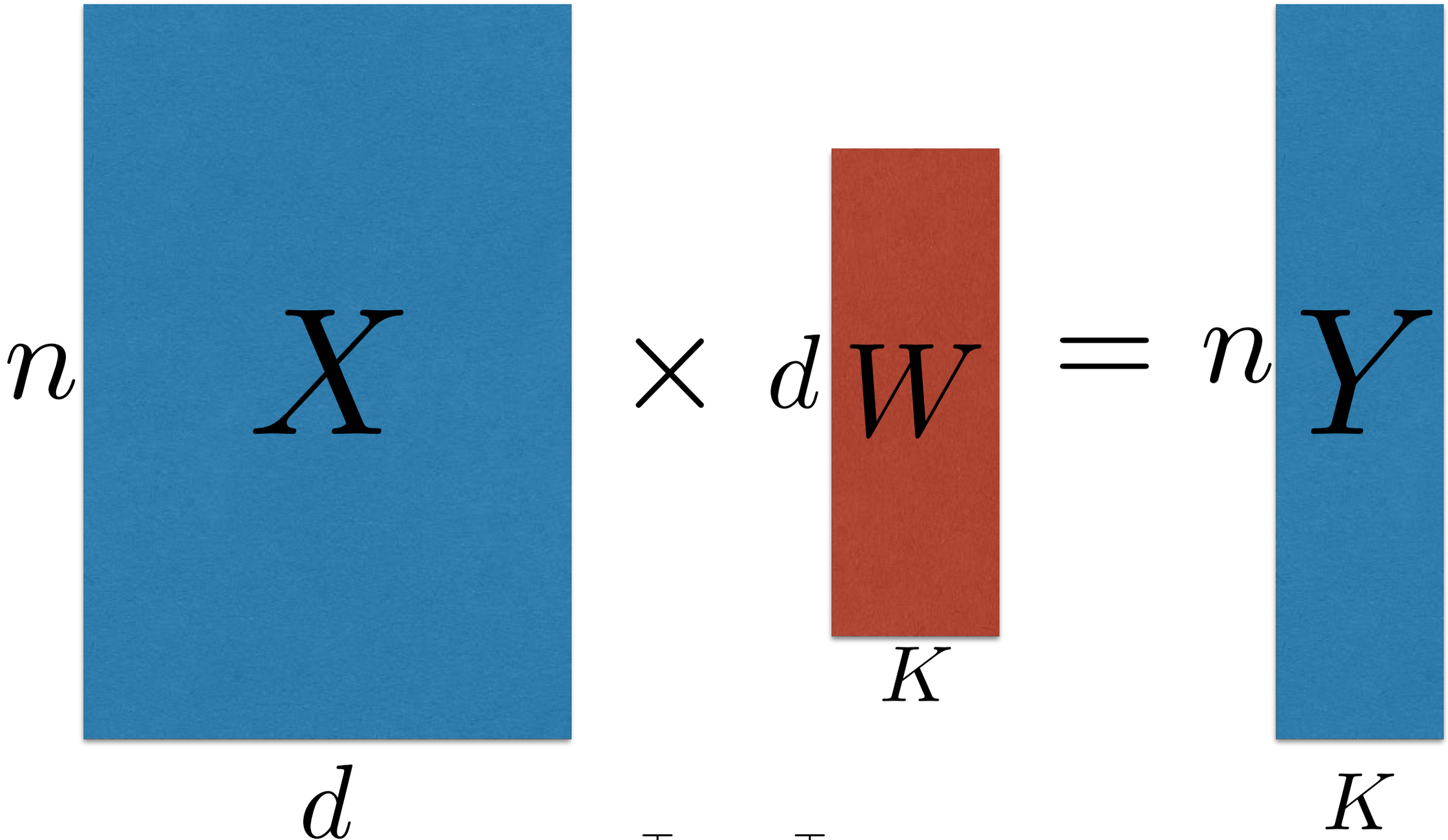
Machine Learning for Data Science (CS4786)
Lecture 5

Random Projections & Canonical Correlation Analysis

Dim Reduction: Linear Transformation



Dim Reduction: Linear Transformation



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

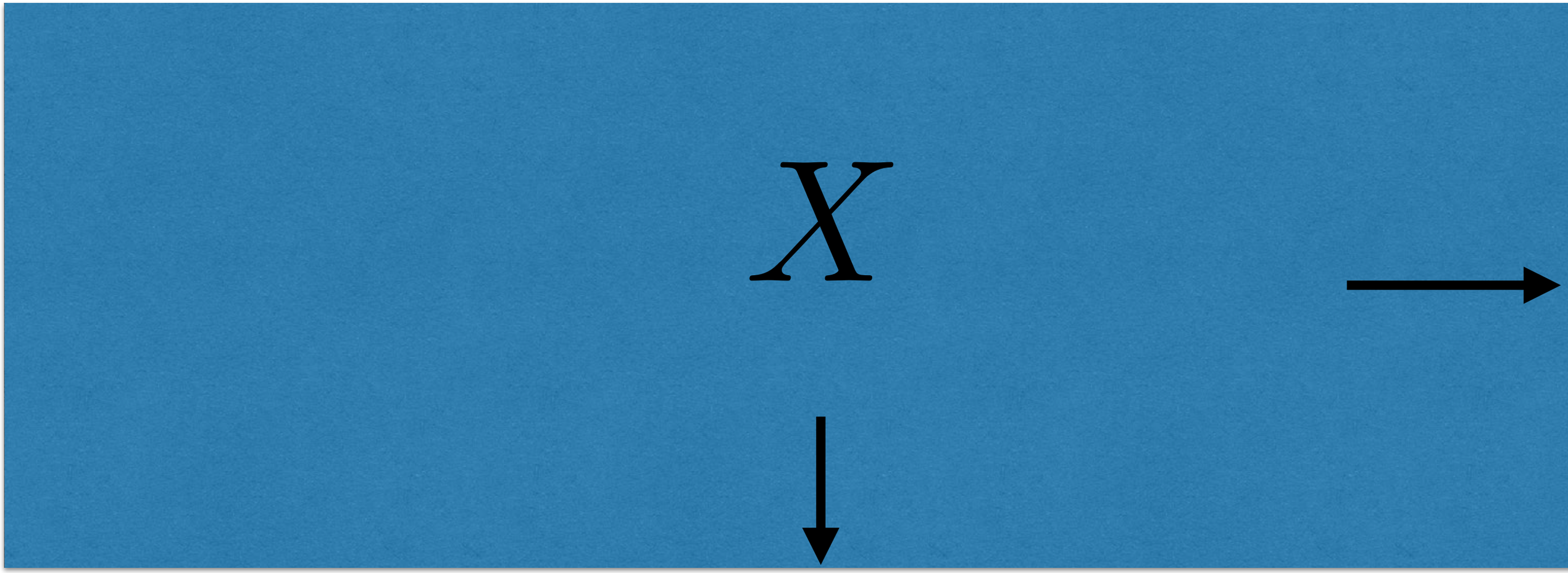
Principal Component Analysis

1. $$\Sigma = \mathrm{cov}\left(X\right)$$

2. $$W = \mathrm{eigs}\left(\Sigma, K\right)$$

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

The Tall, the Fat and **the Ugly**



- 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

$$Y = X \times \begin{bmatrix} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \vdots & \\ +1 & \dots & -1 \end{bmatrix}_K \Big/ \sqrt{d}$$

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

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

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?!

$$y_t - y_s = \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u}$$

Why should Random Projections even work?!

$$
\begin{aligned}
y_t - y_s &= \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u} \\
&= (\mathbf{x}_t - \mathbf{x}_s)^\top \mathbf{u}
\end{aligned}
$$

Why should Random Projections even work?!

$$
\begin{aligned}
y_t - y_s &= \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u} \\
&= (\mathbf{x}_t - \mathbf{x}_s)^\top \mathbf{u} \\
&= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \mathbf{u}[k]
\end{aligned}
$$

Why should Random Projections even work?!

$$
\begin{aligned}
y_t - y_s &= \mathbf{x}_t^\top \mathbf{u} - \mathbf{x}_s^\top \mathbf{u} \\
&= (\mathbf{x}_t - \mathbf{x}_s)^\top \mathbf{u} \\
&= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \mathbf{u}[k]
\end{aligned}
$$

$$
\mathbb{E}\left[y_t - y_s\right] = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \mathbb{E}\left[\mathbf{u}[k]\right] = 0
$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$(y_t - y_s)^2$$

$$= \left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k] \right)^2$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$(y_t - y_s)^2$$

$$= \left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k] \right)^2$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2 \sum_{j=1}^{d} \sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j]) \mathbf{u}[k] \cdot \mathbf{u}[j]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$(y_t - y_s)^2$$

$$= \left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k] \right)^2$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2 \sum_{j=1}^{d} \sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j]) \mathbf{u}[k] \cdot \mathbf{u}[j]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 + 2 \sum_{j=1}^{d} \sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j] \ \mathbf{.} \ [\mathbf{u}[k] \cdot \mathbf{u}[j]]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$\mathbb{E}(y_t - y_s)^2$$

$$= \mathbb{E}\left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]\right)^2$$

$$= \mathbb{E}\left[\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbb{E}\left[\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

Why should Random Projections even work?!

$$y_t - y_s = \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]$$

$$\mathbb{E}(y_t - y_s)^2$$

$$= \mathbb{E}\left(\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])\mathbf{u}[k]\right)^2$$

$$= \mathbb{E}\left[\sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 \mathbf{u}[k]^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 + 2\sum_{j=1}^{d}\sum_{k=j+1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k]) \cdot (\mathbf{x}_t[j] - \mathbf{x}_s[j])\mathbb{E}\left[\mathbf{u}[k] \cdot \mathbf{u}[j]\right]$$

$$= \sum_{k=1}^{d} (\mathbf{x}_t[k] - \mathbf{x}_s[k])^2 = \|\mathbf{x}_t - \mathbf{x}_s\|_2^2$$

Why should Random Projections even work?!

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

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

Lets try ...

Why should Random Projections even work?!

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

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

Lets try ...

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

Pick a Random W

Pick a Random W

$$Y = X \times \begin{bmatrix} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \vdots & \\ +1 & \dots & -1 \end{bmatrix}_K \Big/ \sqrt{d}$$

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

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

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

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

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

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

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

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

- Like repeating the experiment K times and averaging

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

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

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

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

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

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

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

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

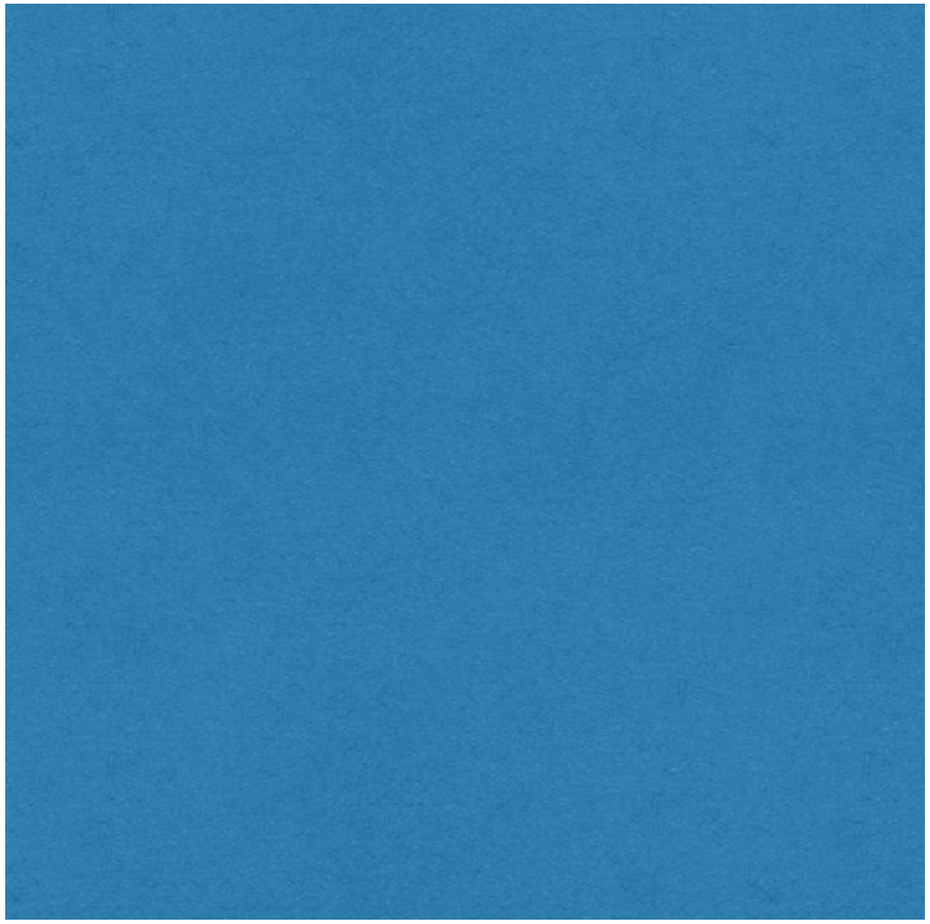
$$(1-\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2 \leq \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2 \leq (1+\epsilon)\left\|\mathbf{y}_i - \mathbf{y}_j\right\|_2^2$$

Lets try ...

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

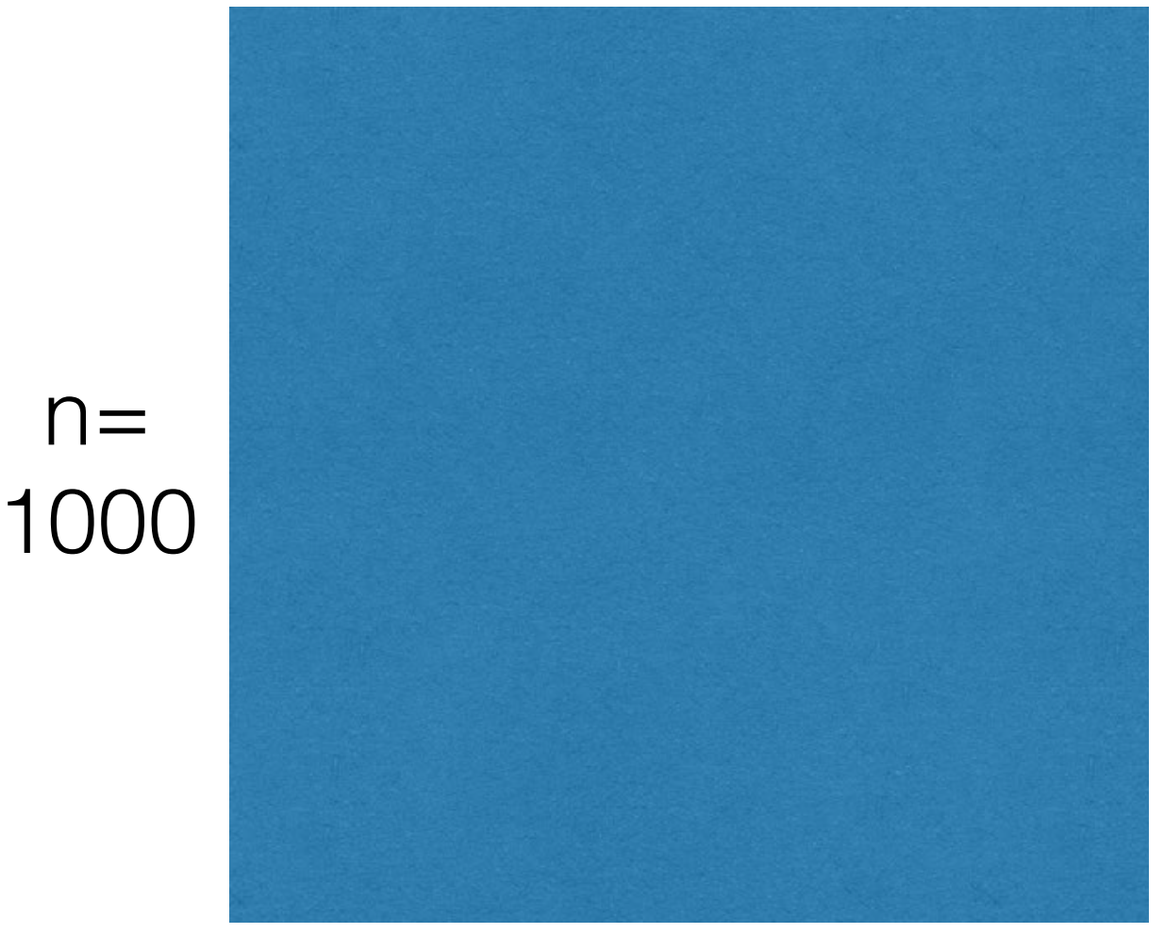
Why is this so Ridiculously Magical?

n=
1000



d = 1000

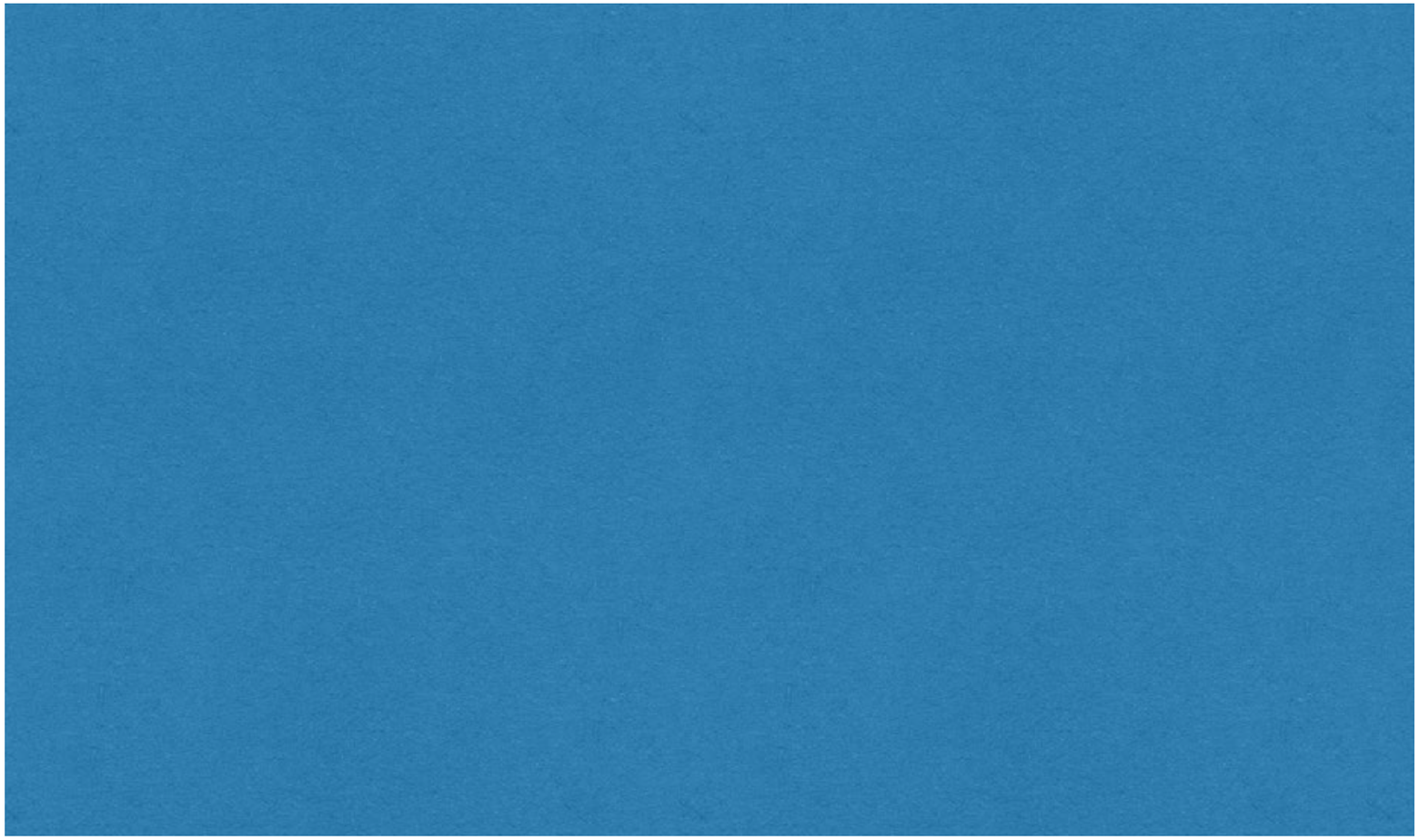
Why is this so Ridiculously Magical?



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

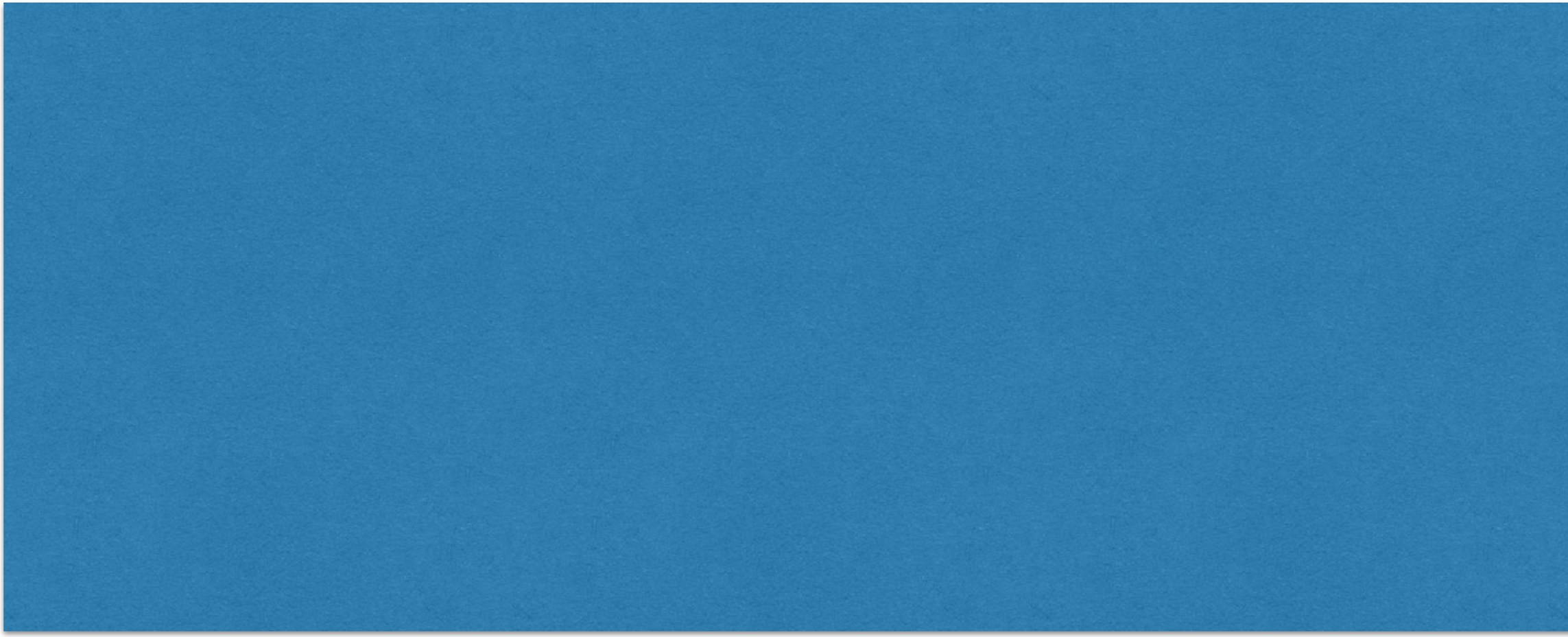


d = 10000

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

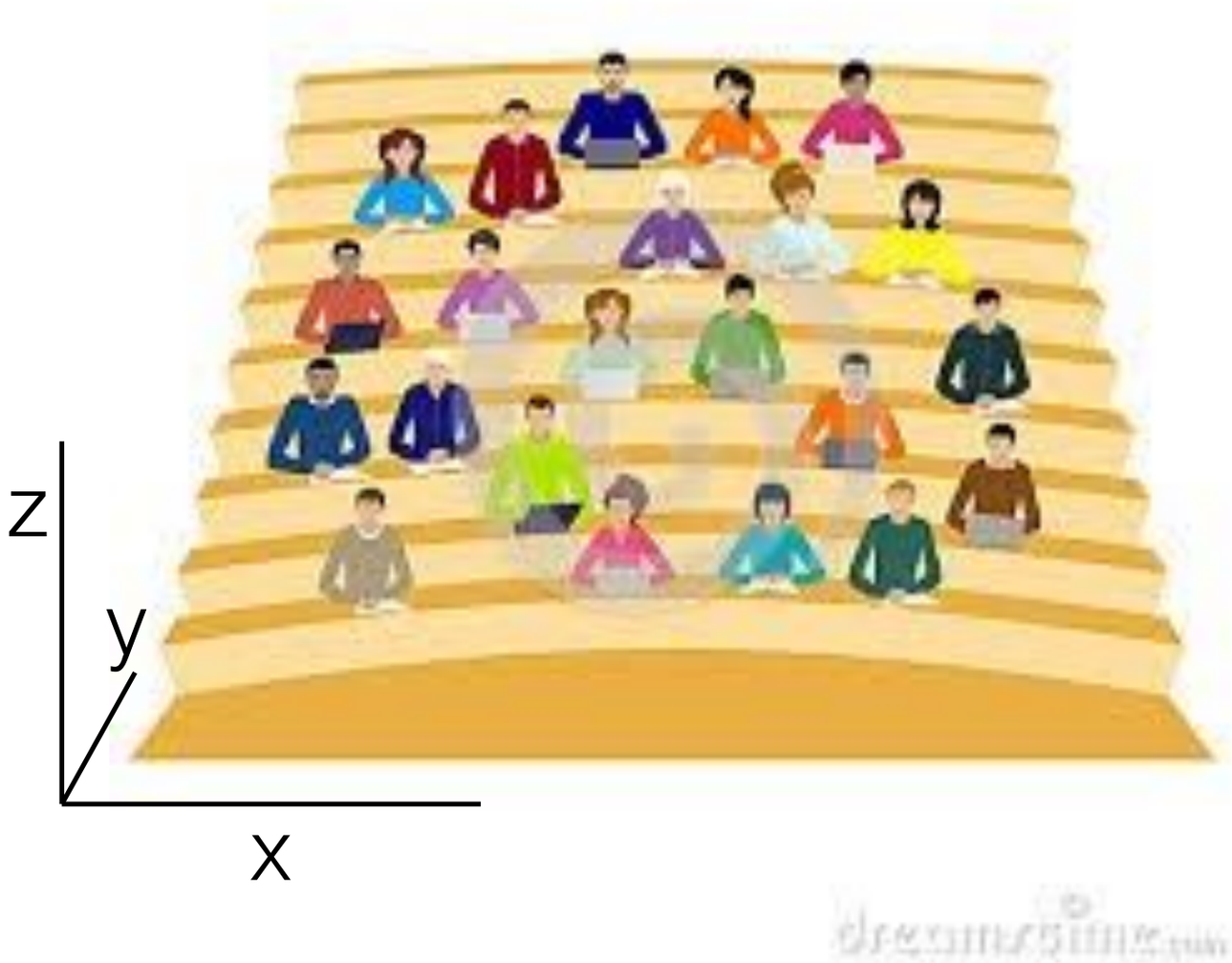
Two View Dimensionality Reduction

- Data comes in pairs $(\mathbf{x}_1, \mathbf{x}'_1), \ldots, (\mathbf{x}_n, \mathbf{x}'_n)$ where $\mathbf{x}_t$'s are d dimensional and $\mathbf{x}'_t$'s are d' dimensional

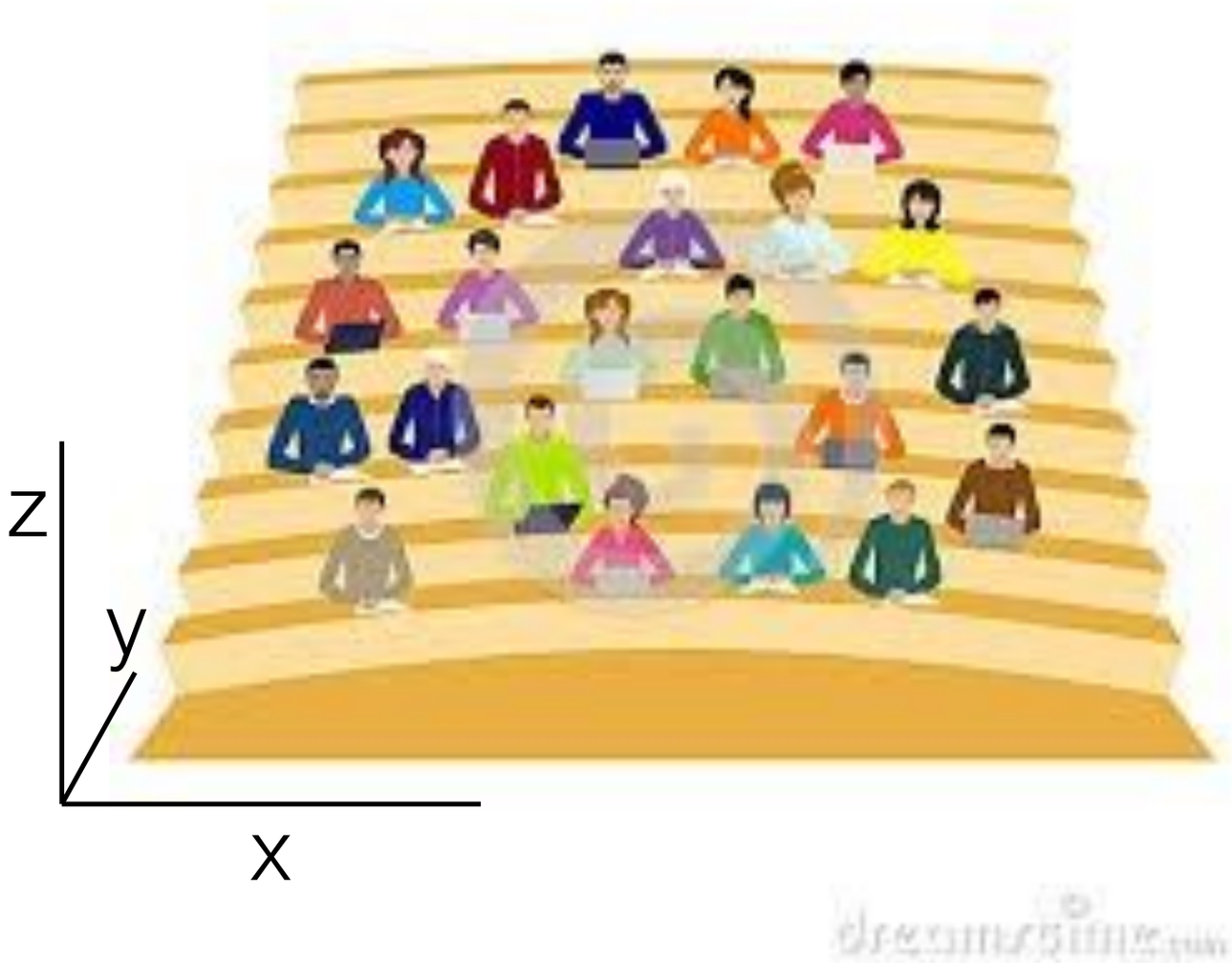
- Goal: Compress say view one into $\mathbf{y}_1, \ldots, \mathbf{y}_n$, that are K dimensional vectors

  - Retain information redundant between the two views
  - Eliminate "noise" specific to only one of the views

Canonical Correlation Analysis



Canonical Correlation Analysis

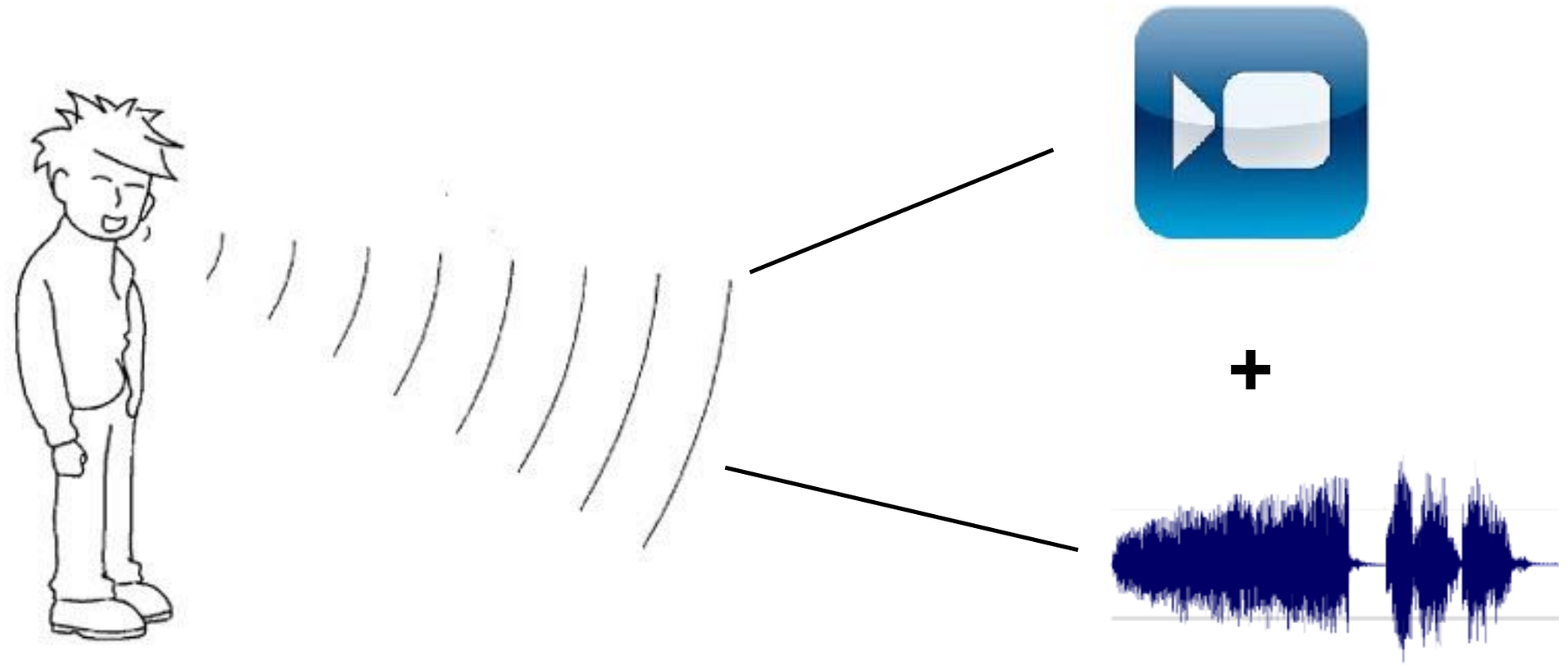


Age

\+ 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)

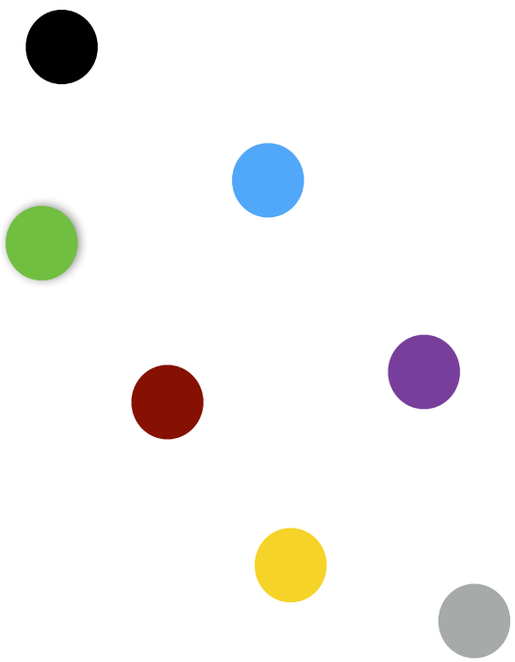
How do we get the right direction? (say K = 1)

Age

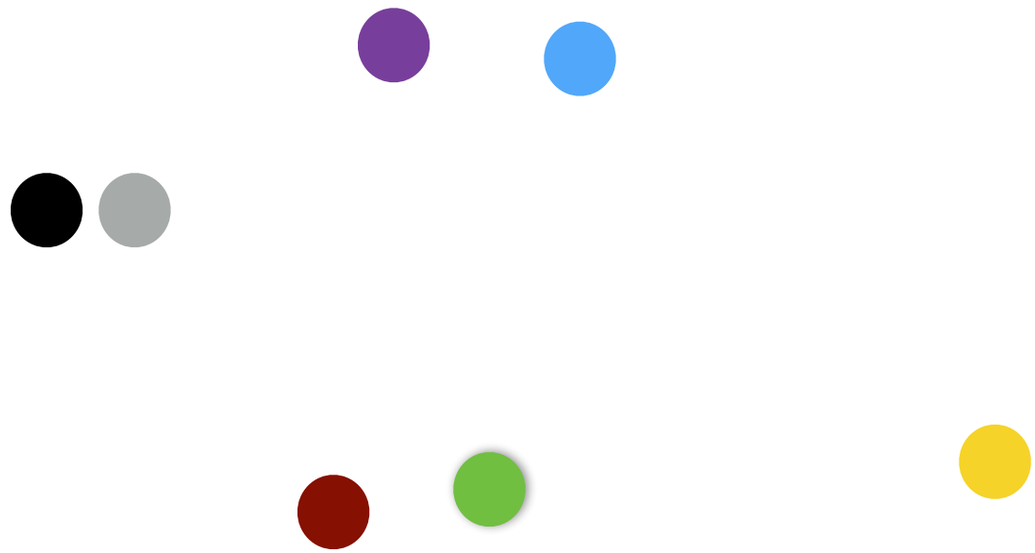
\+ Gender

Candies per week

Which Direction to Pick?

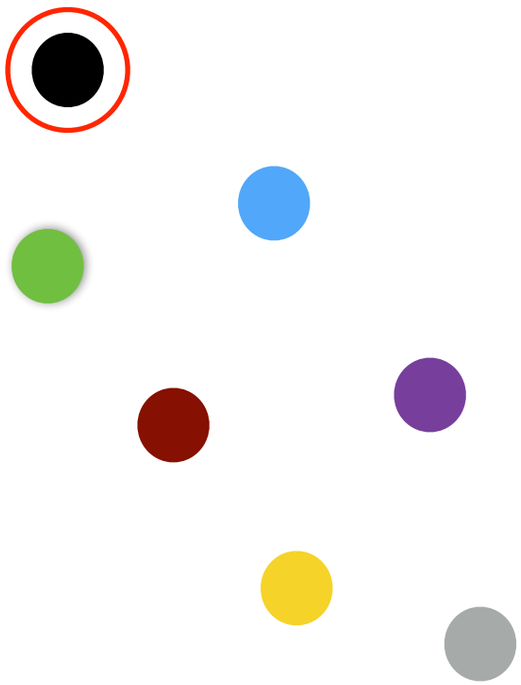


View I

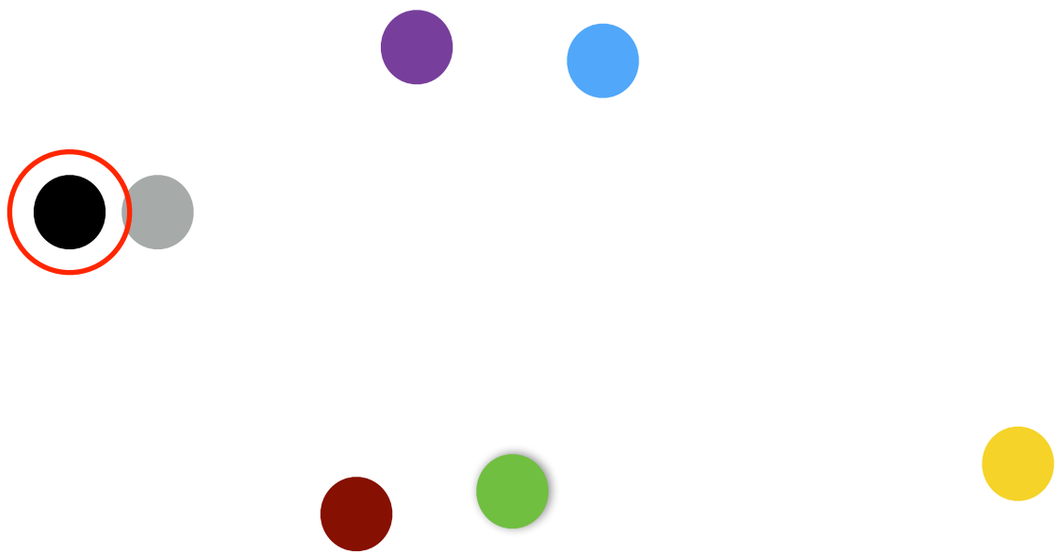


View II

Which Direction to Pick?

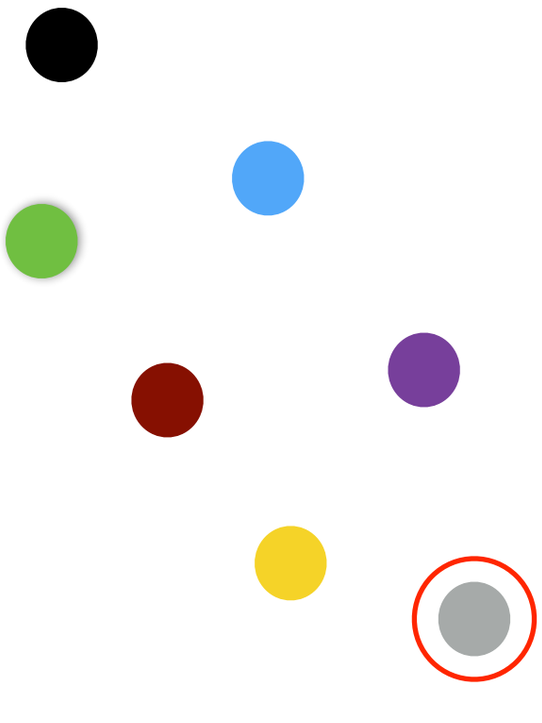


View I

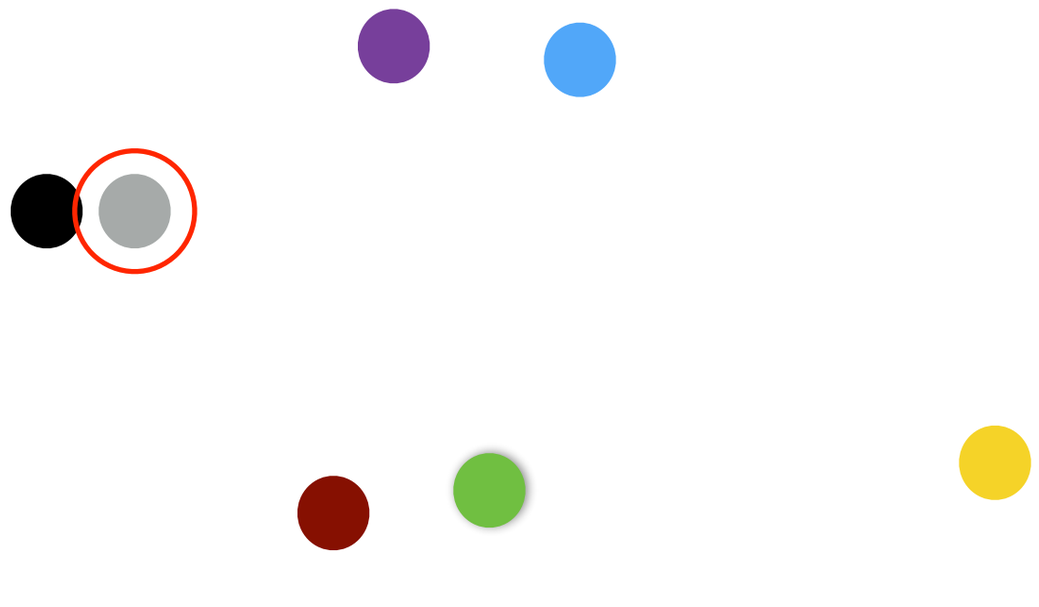


View II

Which Direction to Pick?

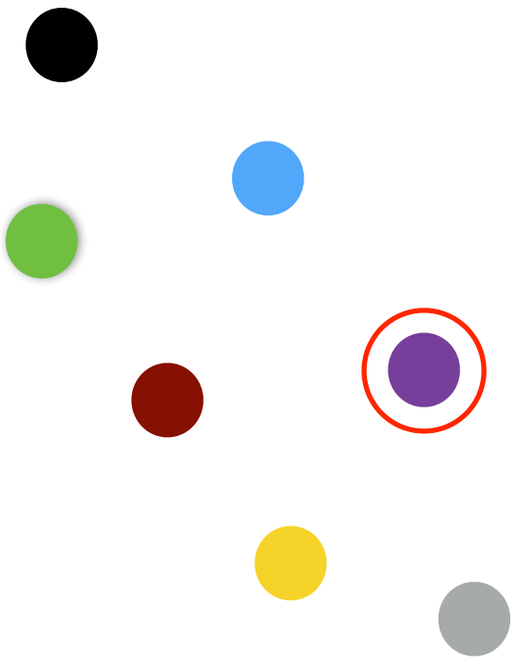


View I

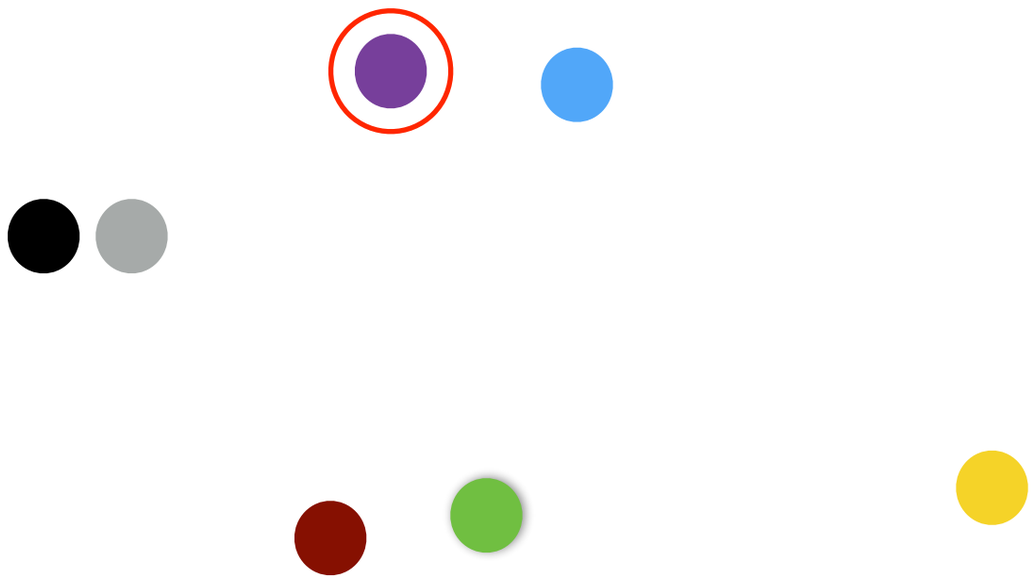


View II

Which Direction to Pick?

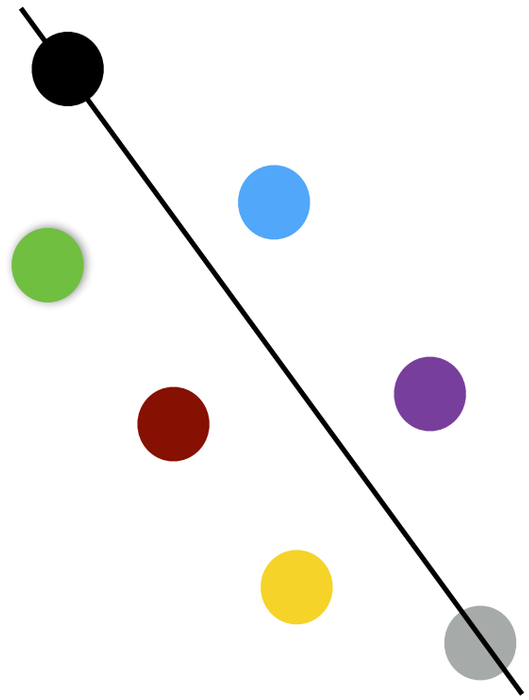


View I

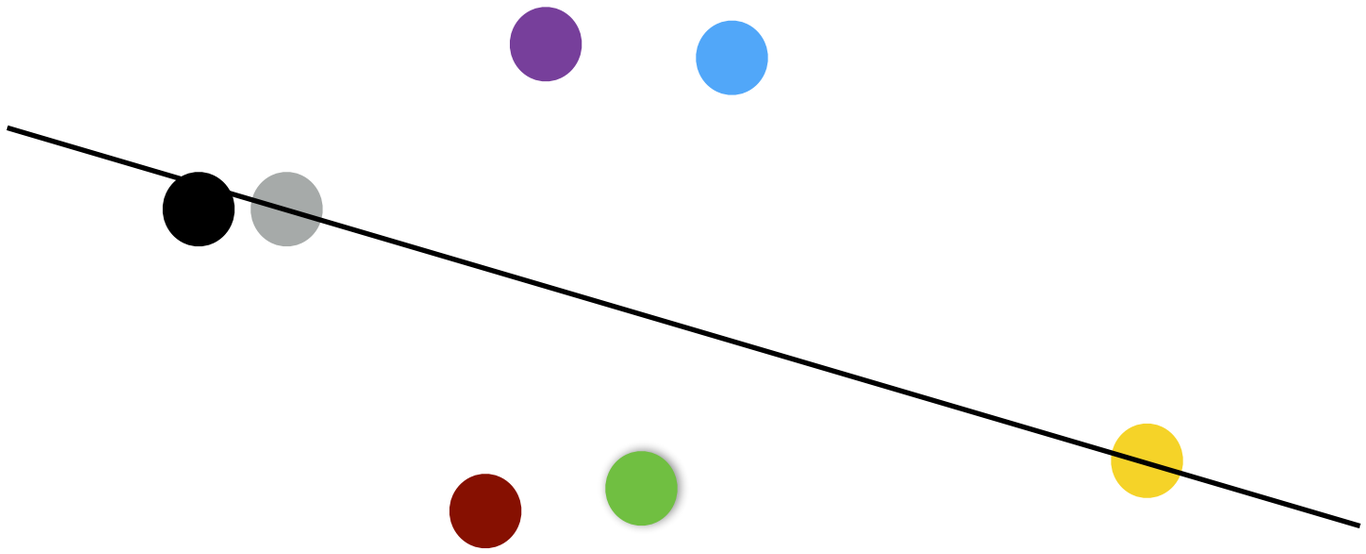


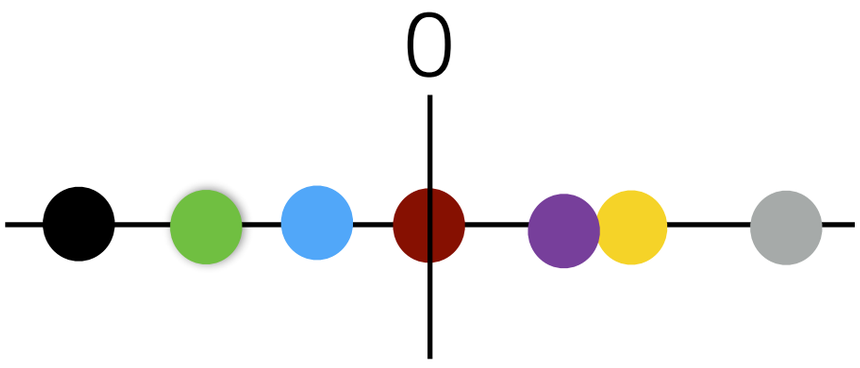
View II

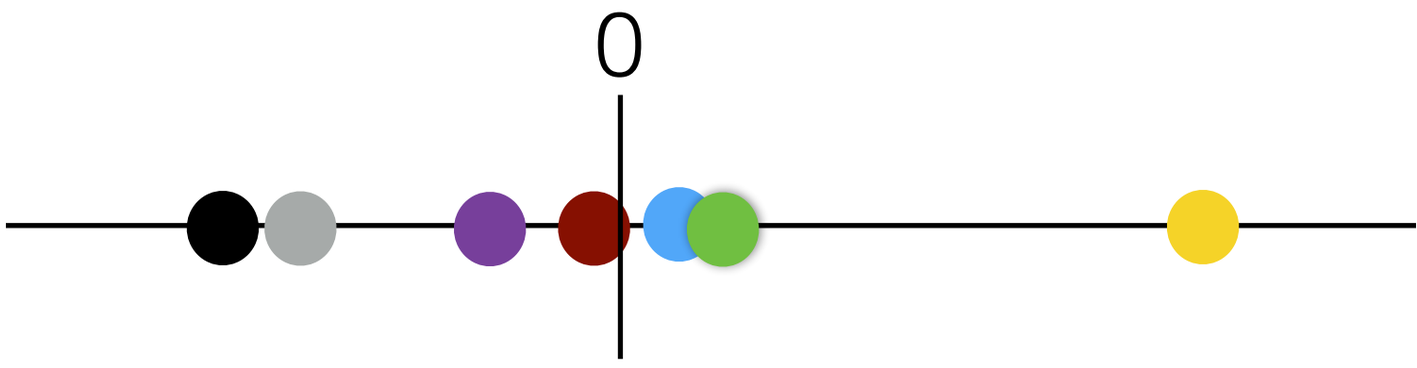
Which Direction to Pick?



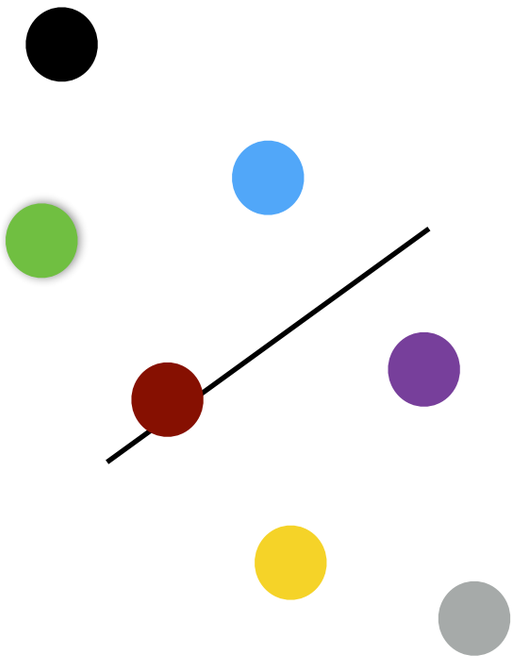
PCA direction

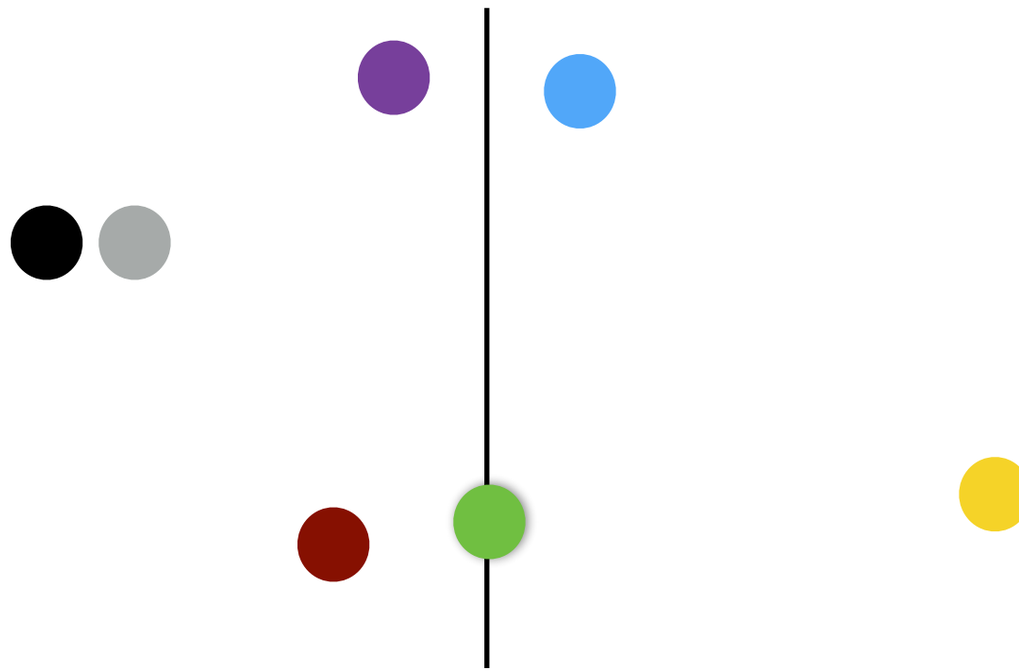






Which Direction to Pick?





Direction has large covariance

How do we pick the right direction to project to?

Maximizing Correlation Coefficient

- Say $\mathbf{w}_1$ and $\mathbf{v}_1$ are the directions we choose to project in views 1 and 2 respectively we want these directions to maximize,

$$\frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{y}_t[1]-\frac{1}{n}\sum_{t=1}^{n}\mathbf{y}_t[1]\right)\cdot\left(\mathbf{y}'_t[1]-\frac{1}{n}\sum_{t=1}^{n}\mathbf{y}'_t[1]\right)$$

where $\mathbf{y}_t[1]=\mathbf{w}_1^\top\mathbf{x}_t$ and $\mathbf{y}'_t[1]=\mathbf{v}_1^\top\mathbf{x}'_t$

What is the problem with the above?

How do we get the right direction? (single dimension K =1)

How do we get the right direction? (single dimension K =1)

Age

\+ Gender

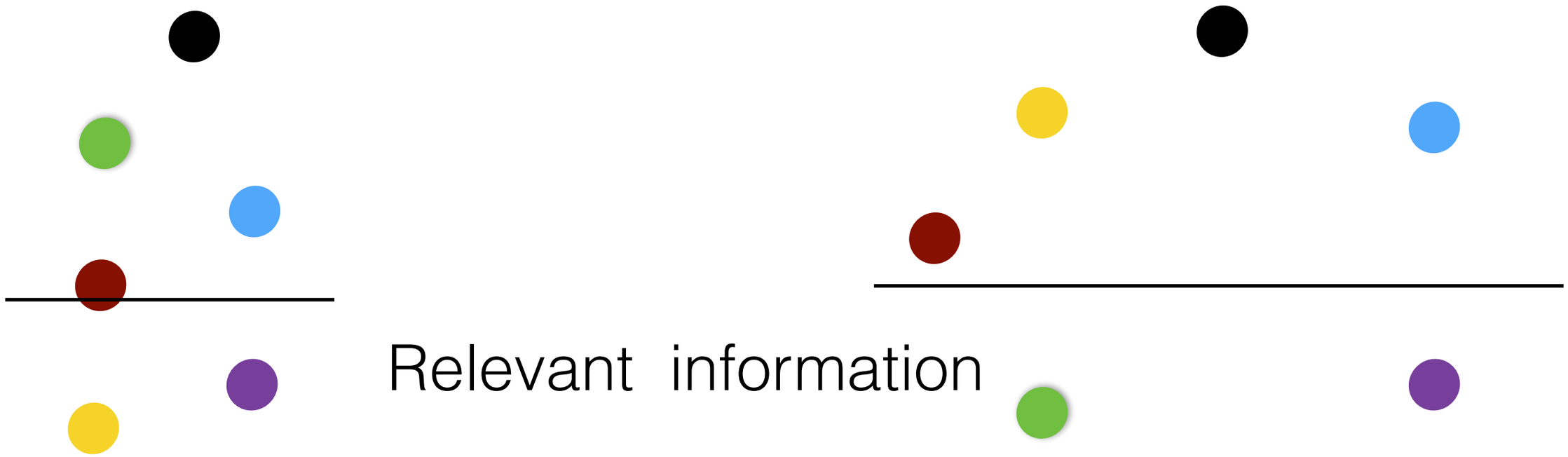
Candies per week

Favorite Cartoon

Why not Maximize Covariance

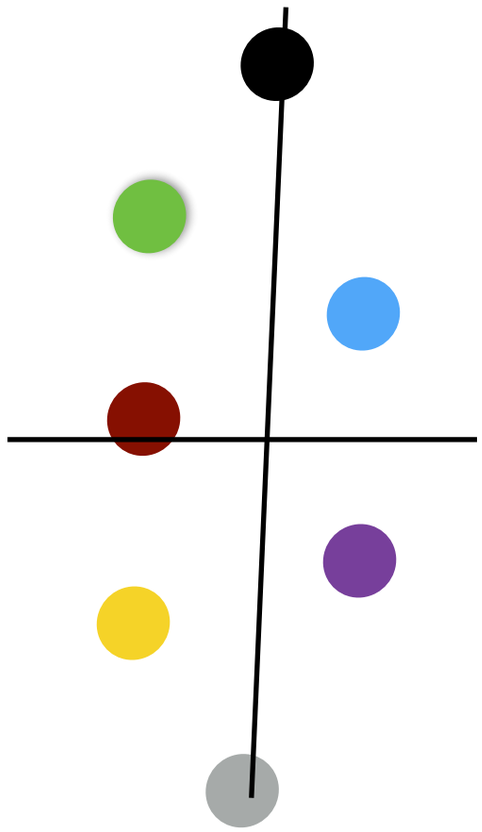
Why not Maximize Covariance

Why not Maximize Covariance

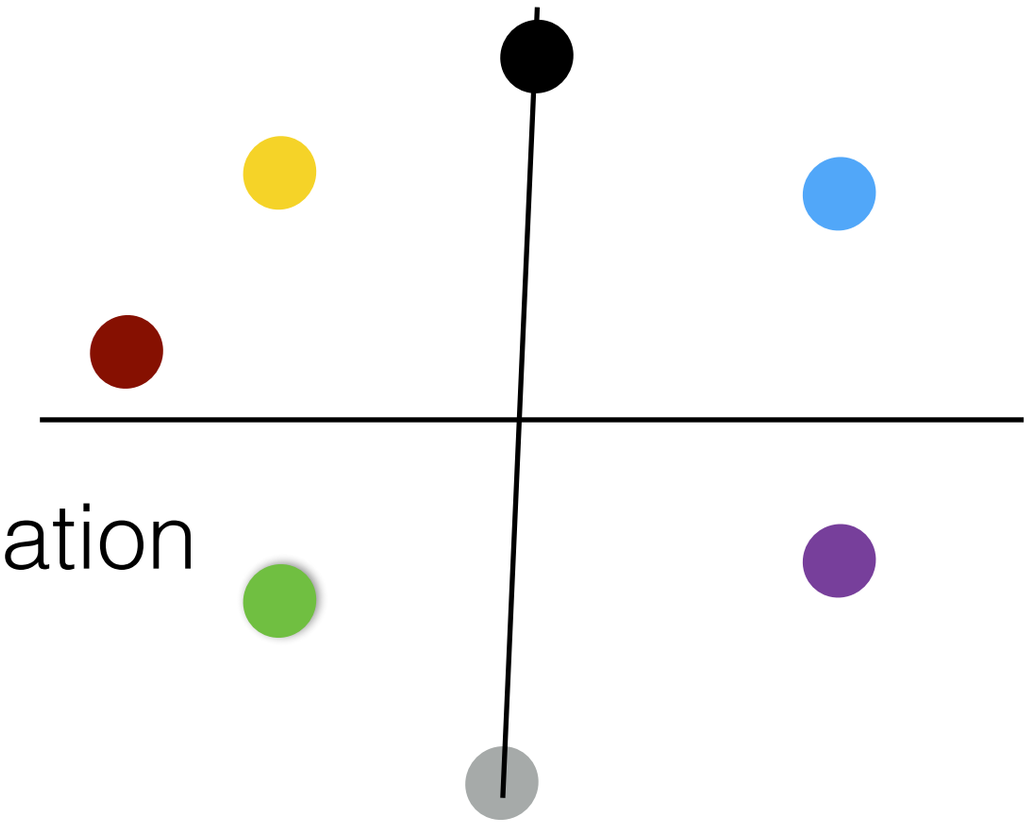


Relevant information

Why not Maximize Covariance



Relevant information



Say covariance in some coordinate just happens to be > 0

Scaling up this coordinate we can blow up covariance