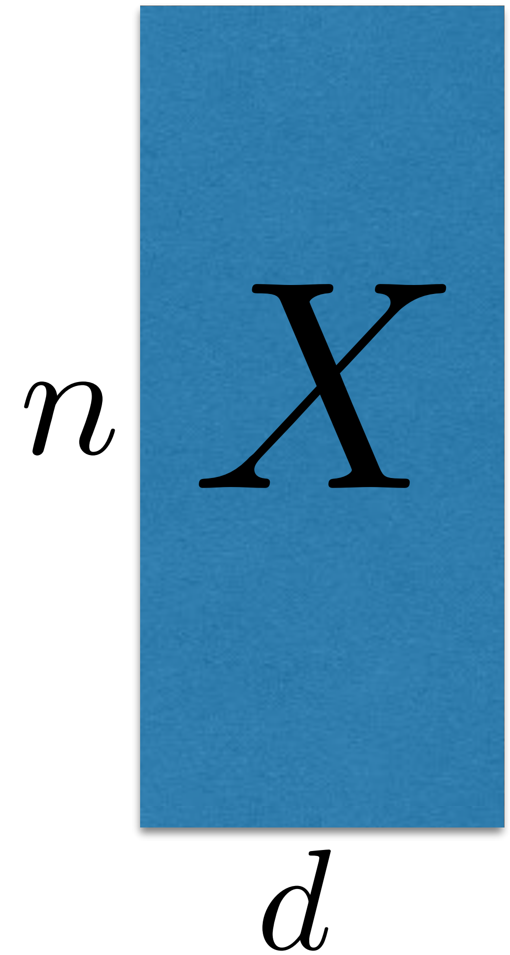
Machine Learning for Data Science (CS4786)
Lecture 11

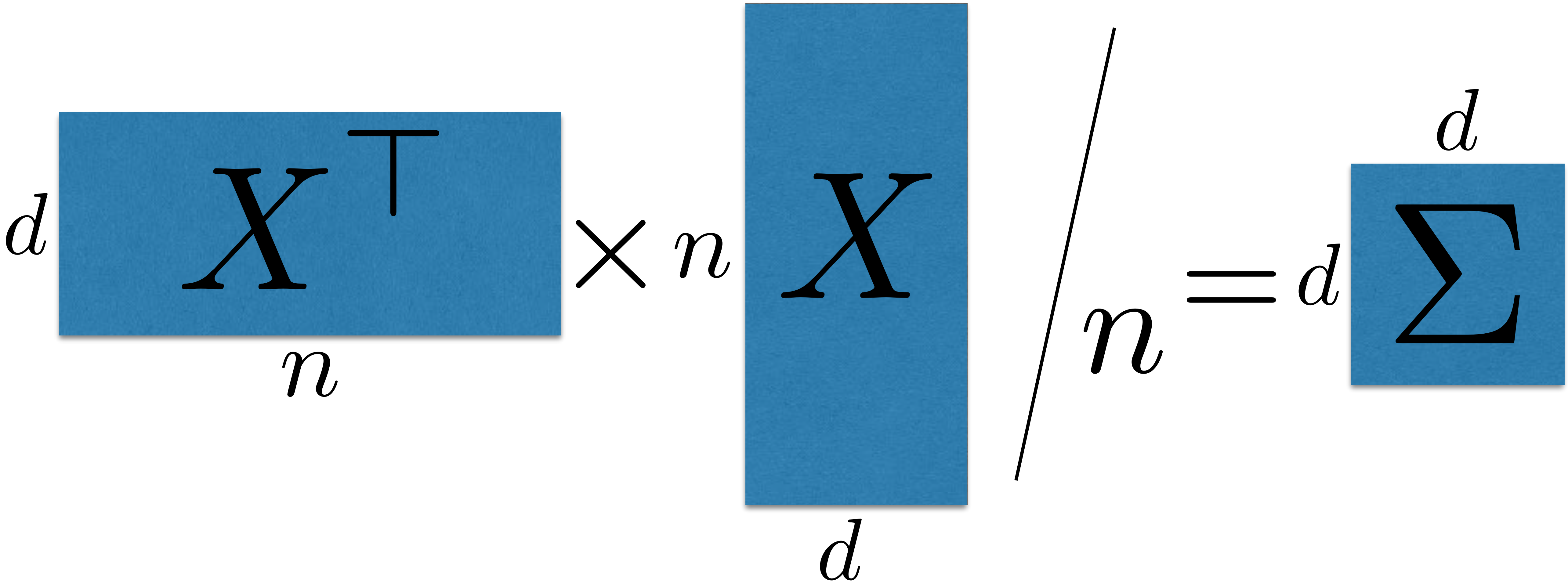
Random Projections & Canonical Correlation Analysis

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2017fa/

The Tall, THE FAT AND THE UGLY



The Tall, the Fat and the Ugly

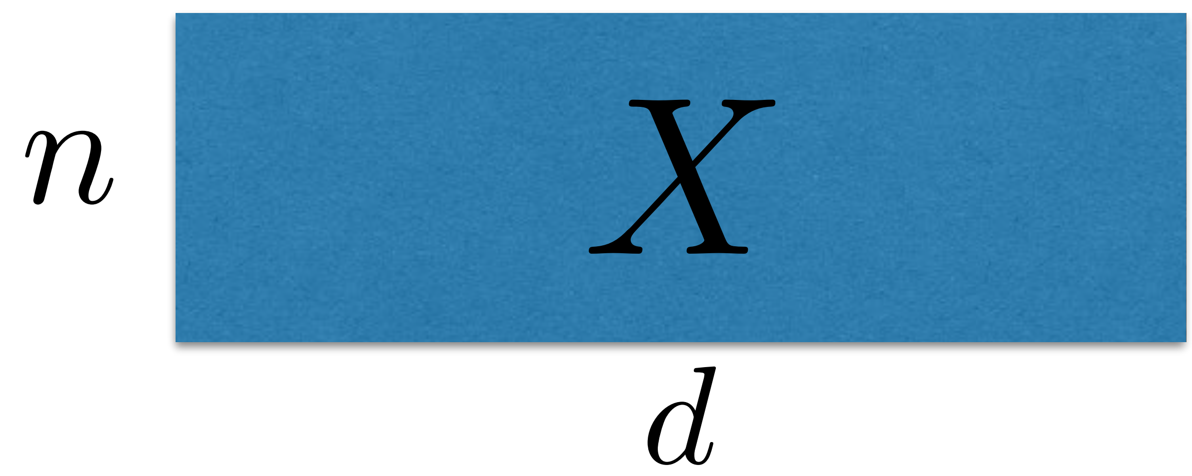


The Tall, the Fat and the Ugly

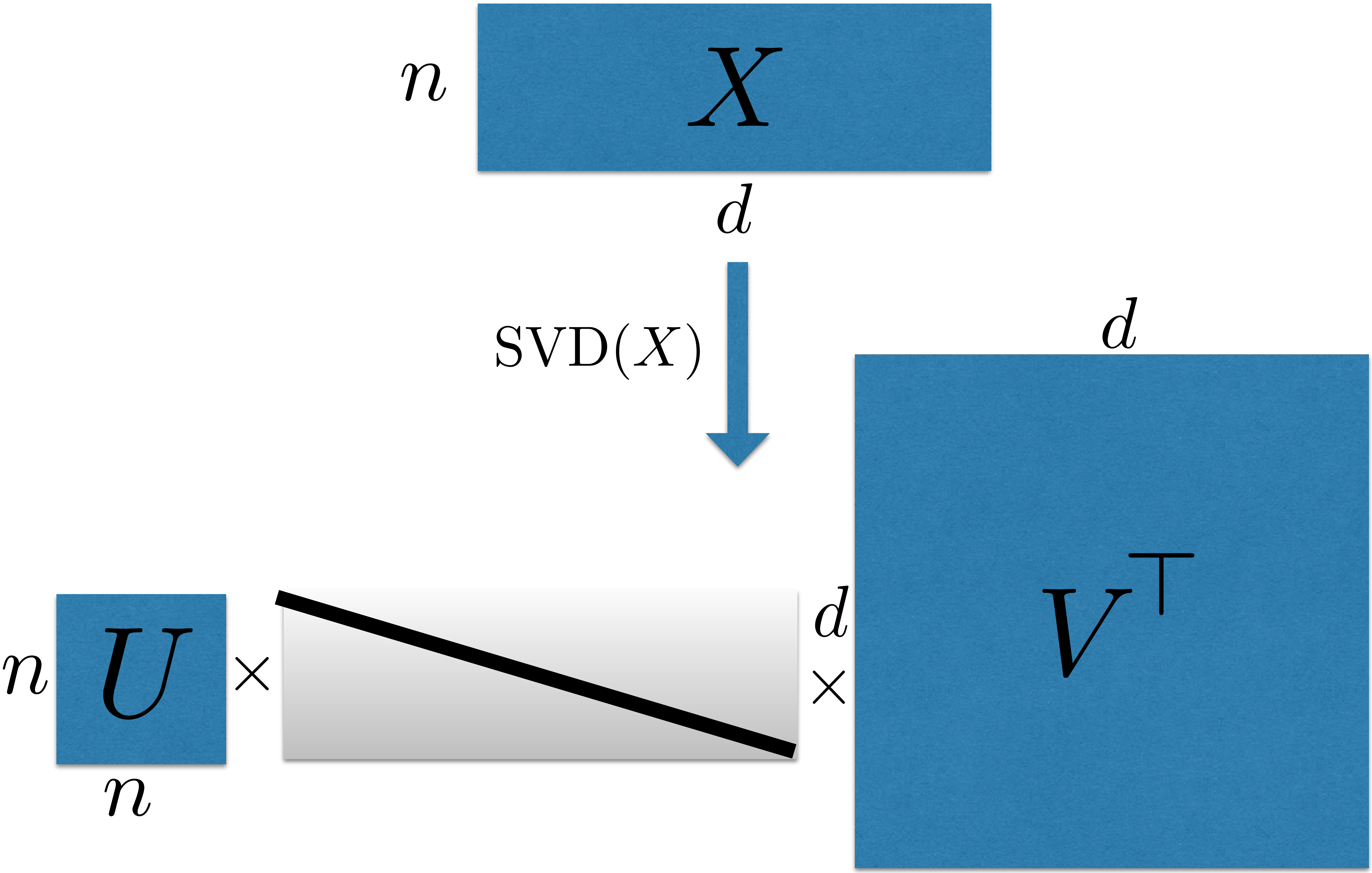
$$X^\top \times X \,/\, n = \Sigma$$

$$W = \mathrm{Eigs}(\Sigma, K)$$

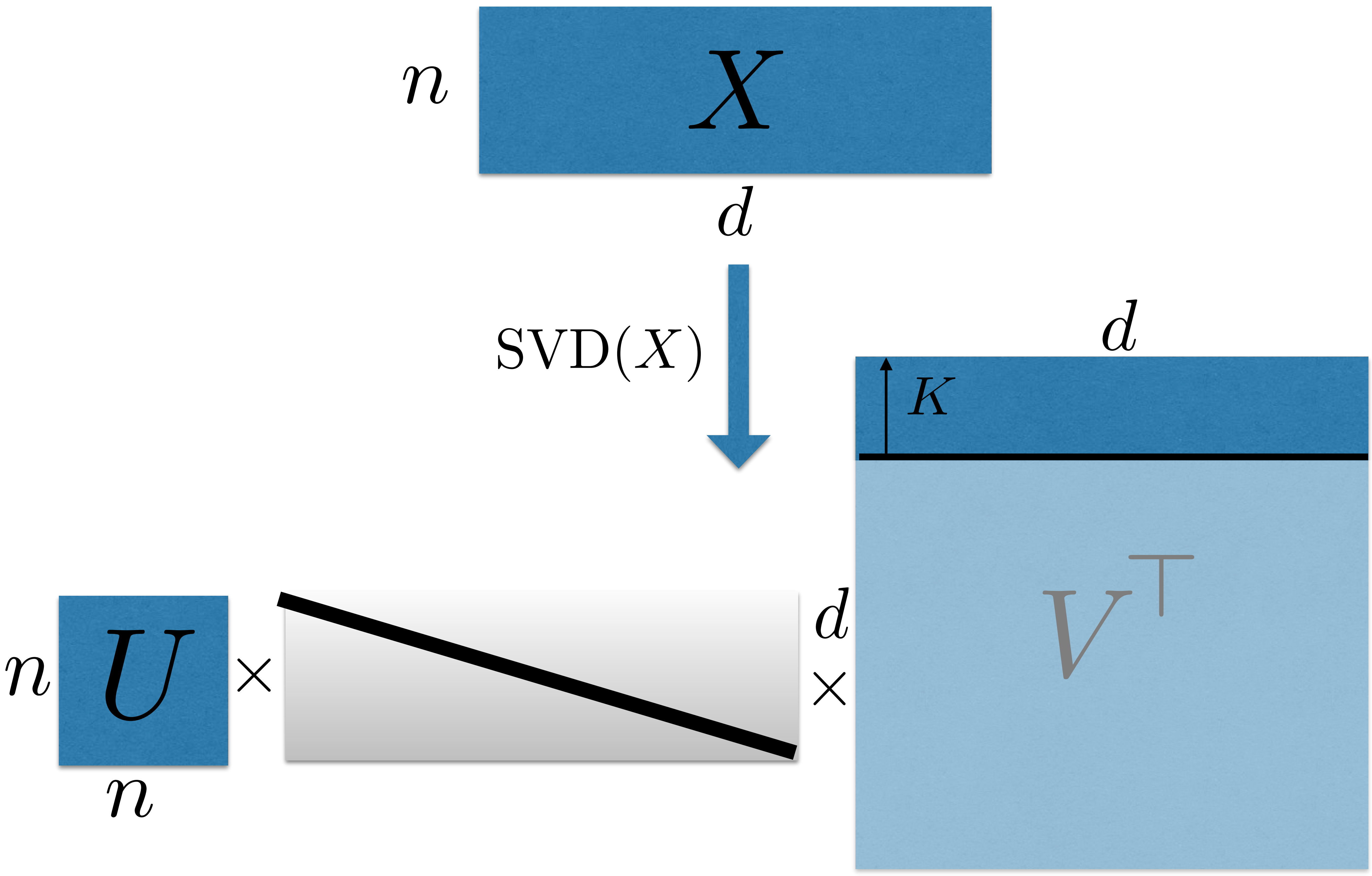
The Tall, **the Fat** and the Ugly



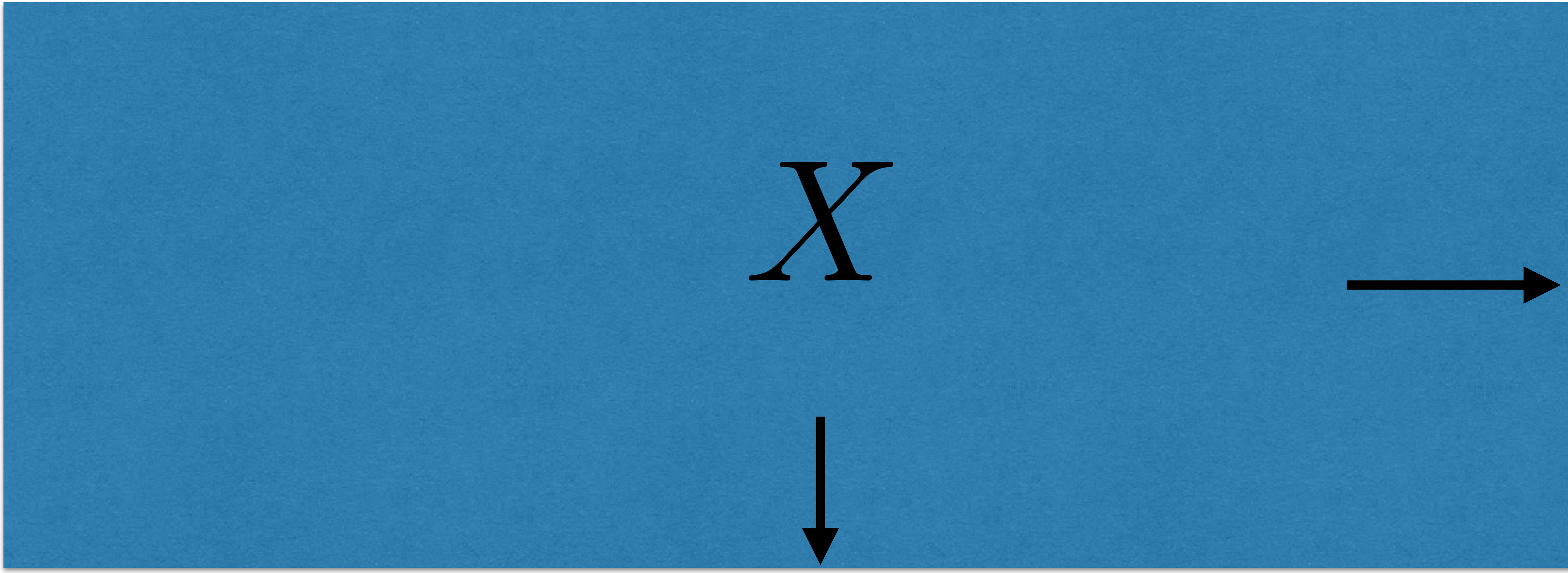
THE TALL, **the Fat** AND THE UGLY



THE TALL, **the Fat** AND THE UGLY



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



- 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

$$Y = X \times \left.\underbrace{\begin{bmatrix} +1 & \dots & -1 \\ -1 & \dots & +1 \\ +1 & \dots & -1 \\ & \vdots & \\ +1 & \dots & -1 \end{bmatrix}}_{K} d \right/ \sqrt{K}$$

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

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

Why should Random Projections even work?!

Say $K = 1$. Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}\, W$. Note that

Why should Random Projections even work?!

Say $K = 1$. Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = \tilde{\mathbf{x}} W$. Note that

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

Why should Random Projections even work?!

Say $K = 1$. Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}\, W$. Note that

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

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

WHY SHOULD RANDOM PROJECTIONS EVEN WORK?!

Say $K = 1$. Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}\, W$. Note that

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

Why should Random Projections even work?!

Say $K = 1$. Consider any vector $\tilde{\mathbf{x}} \in \mathbb{R}^d$ and let $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}\, W$. Note that

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

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

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

However $W^2[i,1] = 1/K = 1$ when $K = 1$

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

Why should Random Projections even work?!

Hence,

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

Why should Random Projections even work?!

Hence,

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

However $W[i,1]$ and $W[i',1]$ are independent and so

$$\mathbb{E}\left[W[i,1] \cdot W[i',1]\right] = \mathbb{E}\left[W[i,1]\right] \cdot \mathbb{E}\left[W[i',1]\right] = 0$$

Why should Random Projections even work?!

Hence,

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

However $W[i,1]$ and $W[i',1]$ are independent and so

$$\mathbb{E}\left[W[i,1] \cdot W[i',1]\right] = \mathbb{E}[W[i,1]] \cdot \mathbb{E}\left[W[i',1]\right] = 0$$

Using this we conclude that

$$\mathbb{E}\left[\tilde{\mathbf{y}}^2\right] = \sum_{i=1}^{d} \tilde{\mathbf{x}}^2[i] = \|\tilde{\mathbf{x}}\|^2$$

Why should Random Projections even work?!

Hence,

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

Why should Random Projections even work?!

Hence,

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

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

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

Why should Random Projections even work?!

Hence,

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

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

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

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

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

Why should Random Projections even work?!

Hence,

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

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

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

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 this in Matlab ...

Why should Random Projections even work?!

- Setting K large is like getting K samples.

Why should Random Projections even work?!

- Setting K large is like getting K samples.
- Specifically since we take W to be random signs normalized by $\sqrt{K}$, for each $j \in [K]$, for any $\tilde{\mathbf{x}}$ if $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}\, W$, then

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

Why should Random Projections even work?!

- Setting K large is like getting K samples.
- Specifically since we take W to be random signs normalized by $\sqrt{K}$, for each $j \in [K]$, for any $\tilde{\mathbf{x}}$ if $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}\, W$, then

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

Hence we can conclude that

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

Why should Random Projections even work?!

- Setting K large is like getting K samples.
- Specifically since we take W to be random signs normalized by $\sqrt{K}$, for each $j \in [K]$, for any $\tilde{\mathbf{x}}$ if $\tilde{\mathbf{y}} = \tilde{\mathbf{x}}\, W$, then

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

Hence we can conclude that

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

This is like taking an average of K independent measurements whose expectations are $\|\tilde{\mathbf{x}}\|_2^2$

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

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

Why should Random Projections even work?!

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

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

Lets try on Matlab …

Why should Random Projections even work?!

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

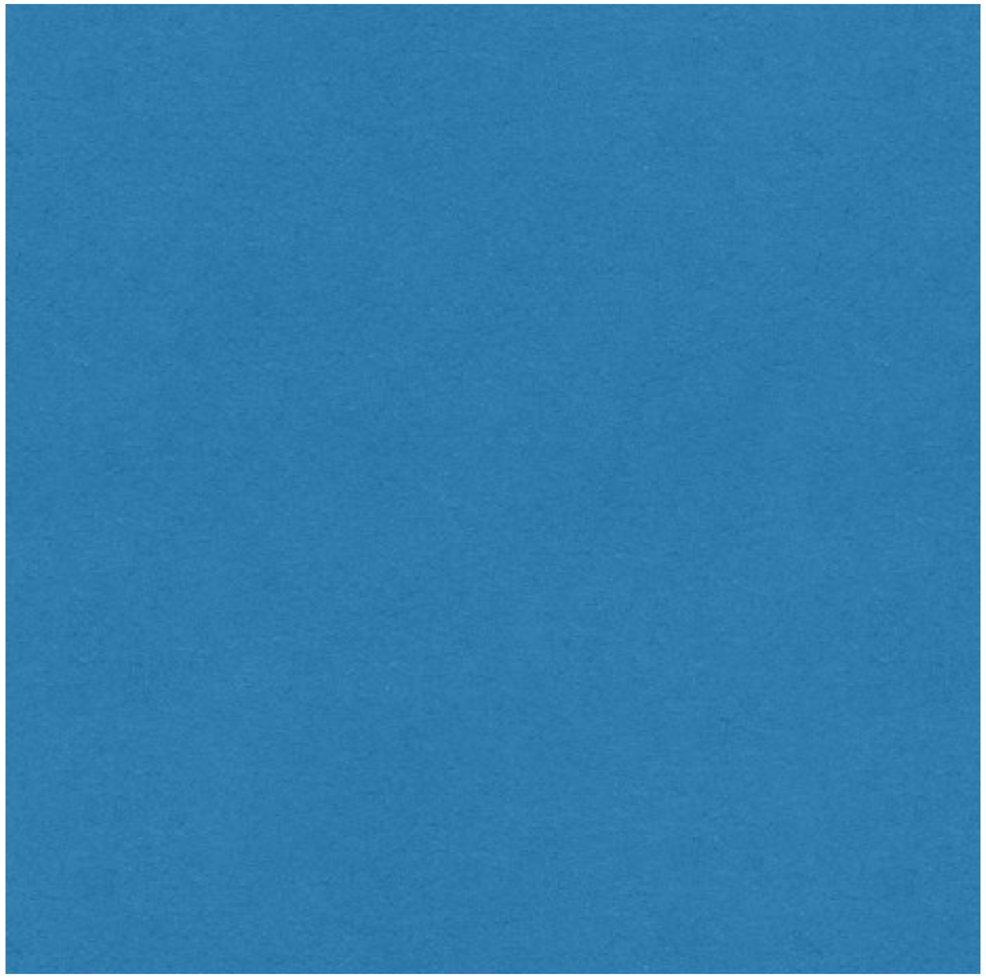
For any $\epsilon > 0$, if $K \approx \log(n/\delta)/\epsilon^2$, with probability $1 - \delta$ over draw of W, for all pairs of data points $i, j \in \{1, \ldots, n\}$,

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

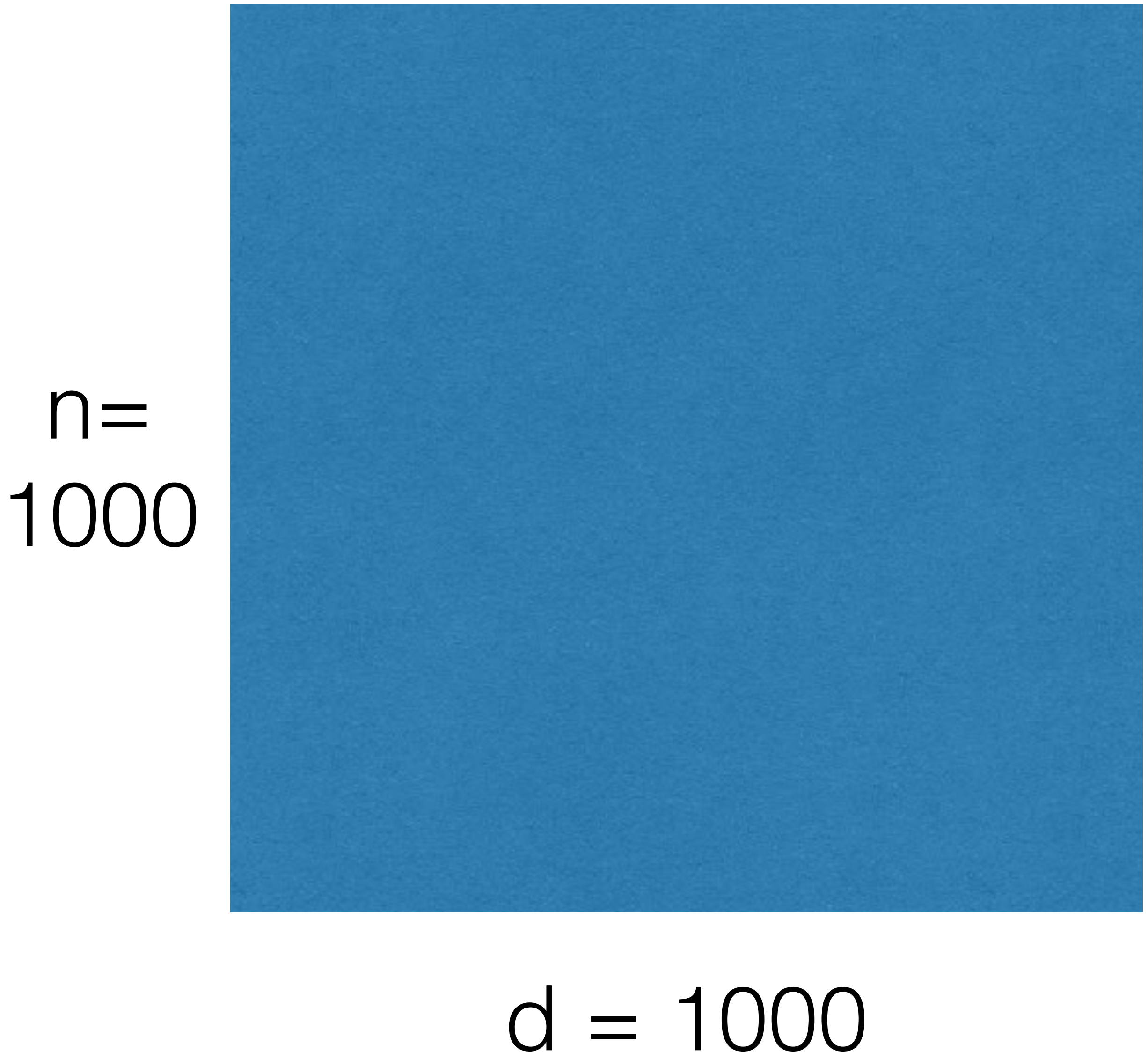
Lets try on Matlab ...

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

Why is this so Ridiculously Magical?



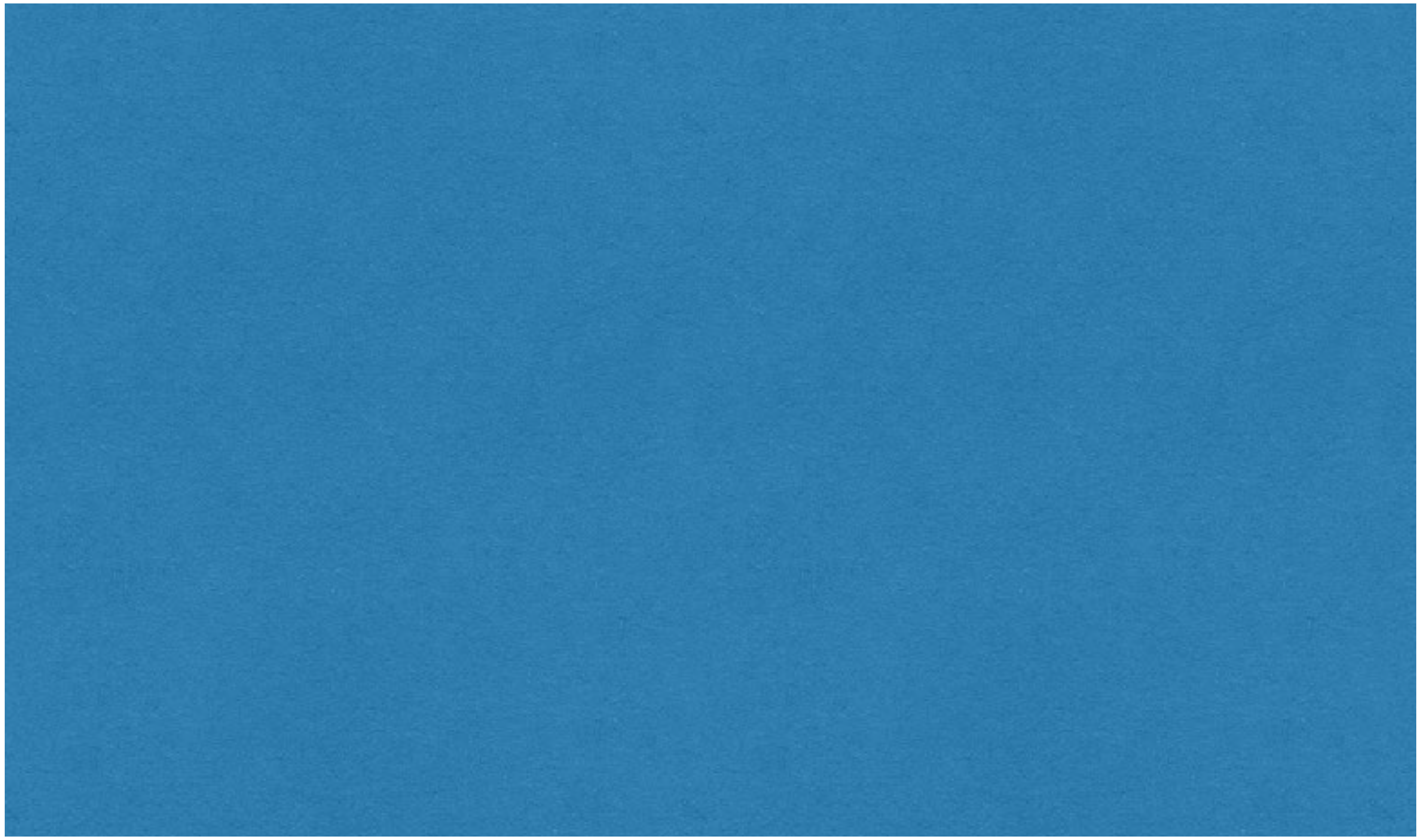
Why is this so Ridiculously Magical?



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

Why is this so Ridiculously Magical?

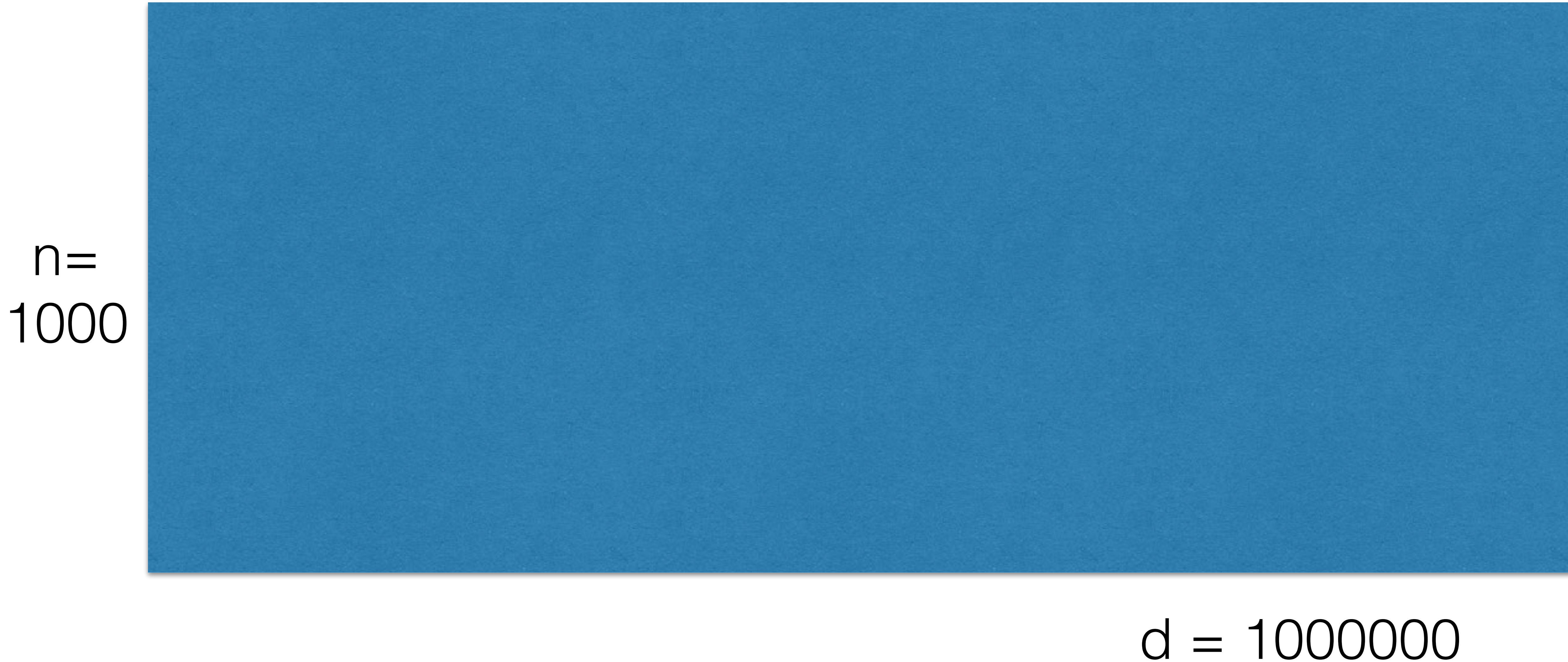
n=
1000



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?



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

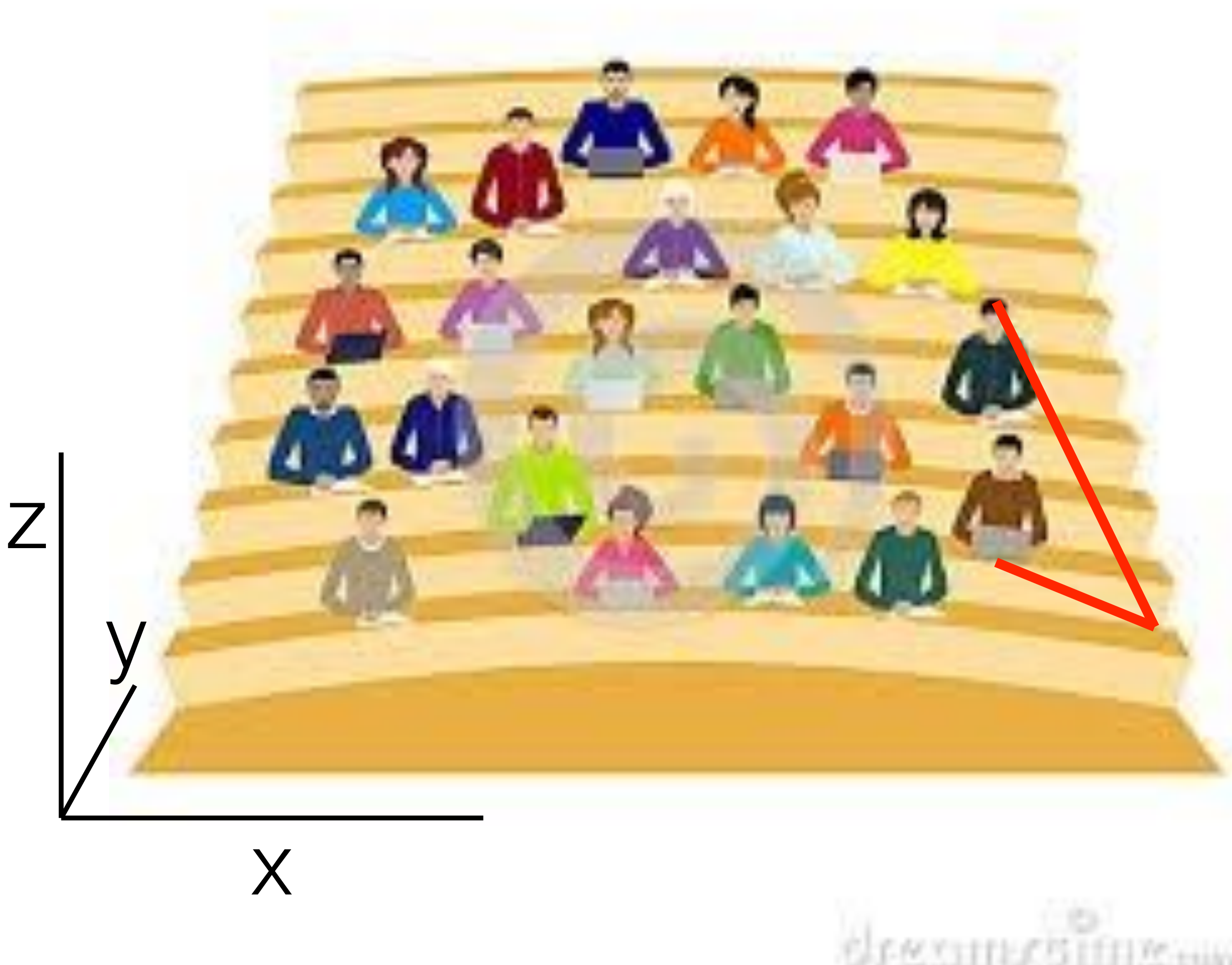
+

Age

Gender

Angle

Canonical Correlation Analysis



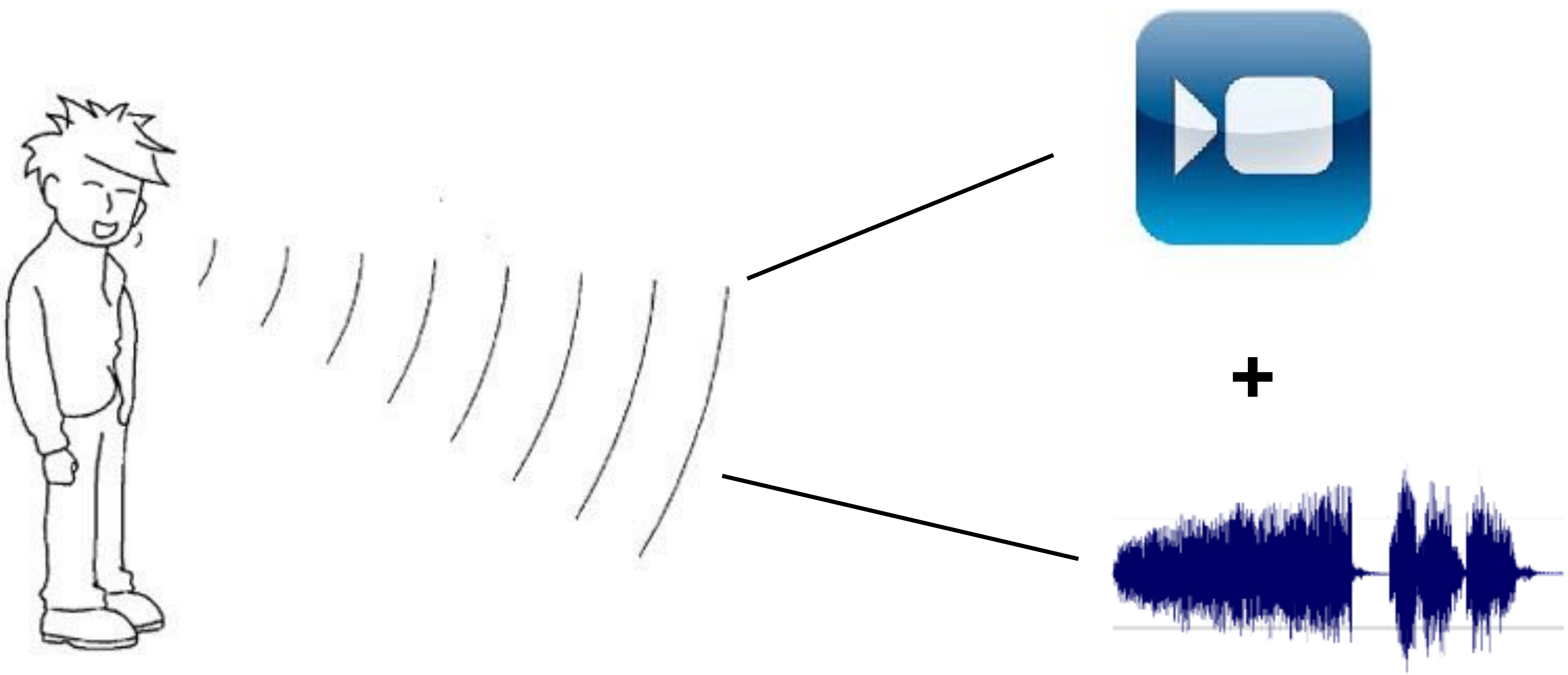
\+

Age

Gender

Angle

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)

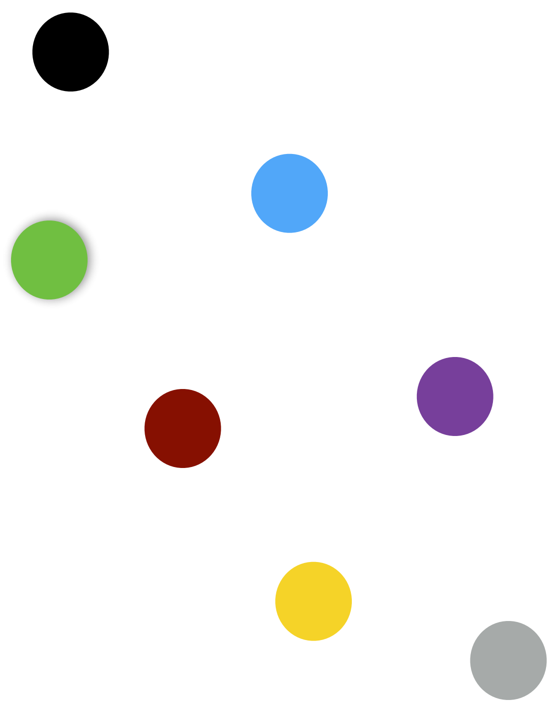
+

Age

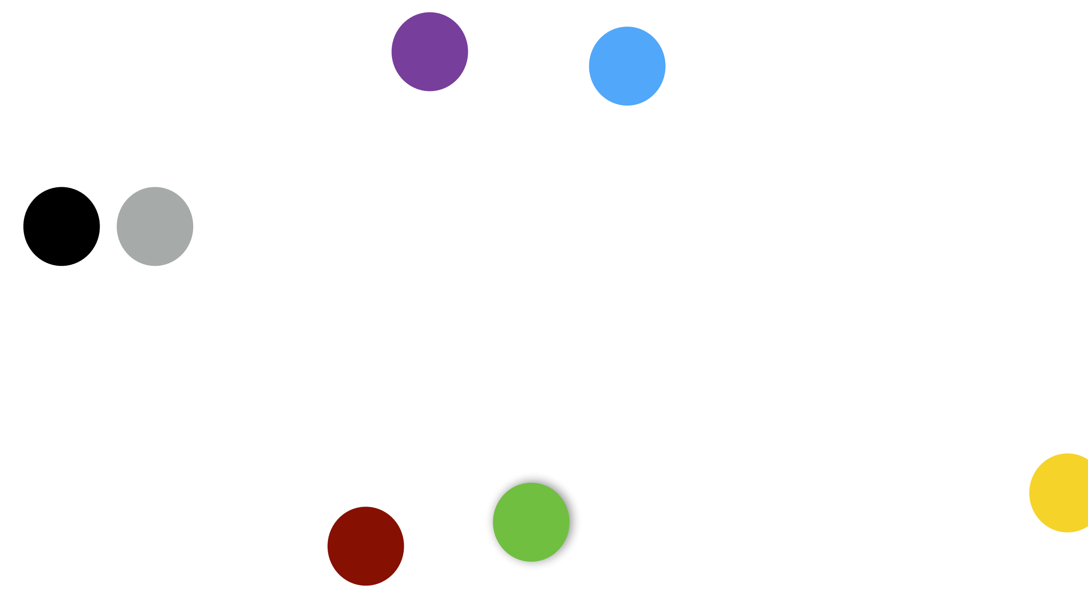
Gender

Angle

Which Direction to Pick?

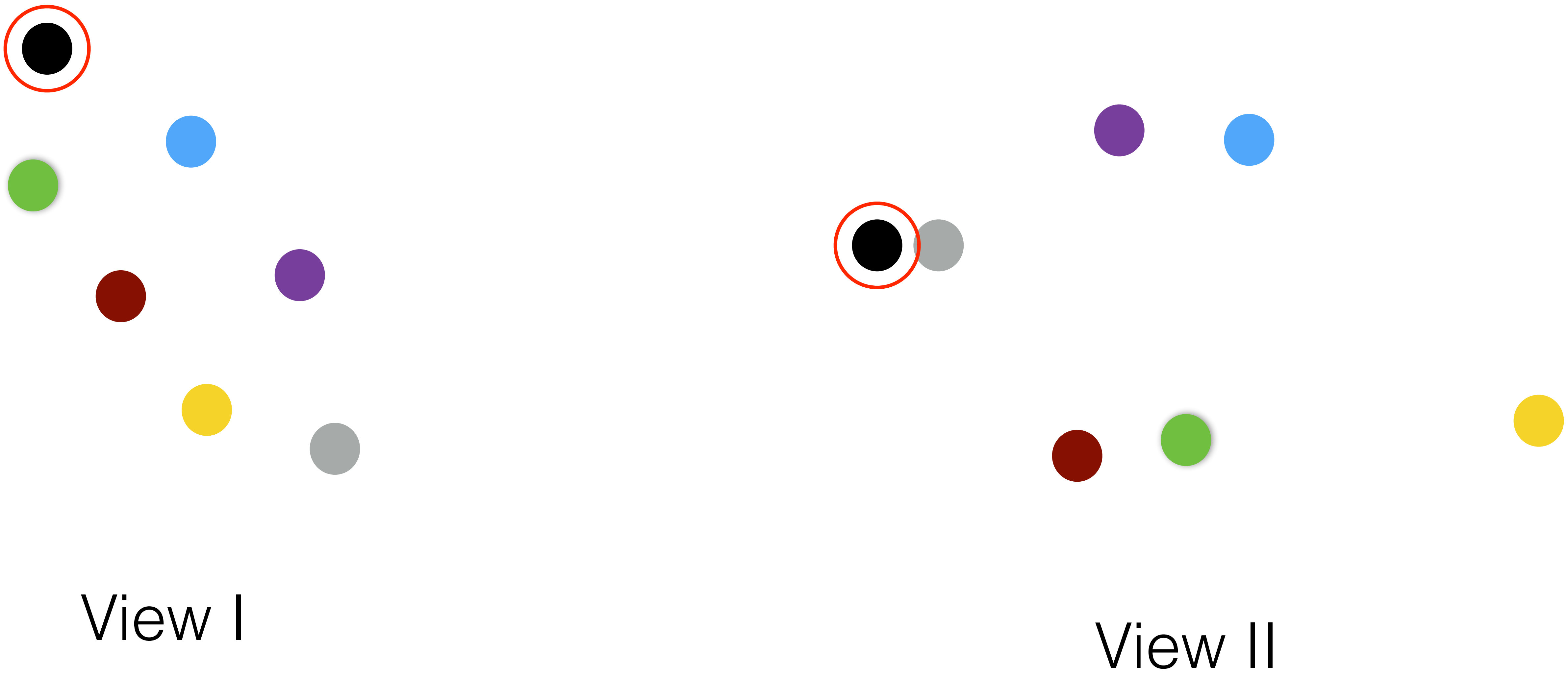


View I

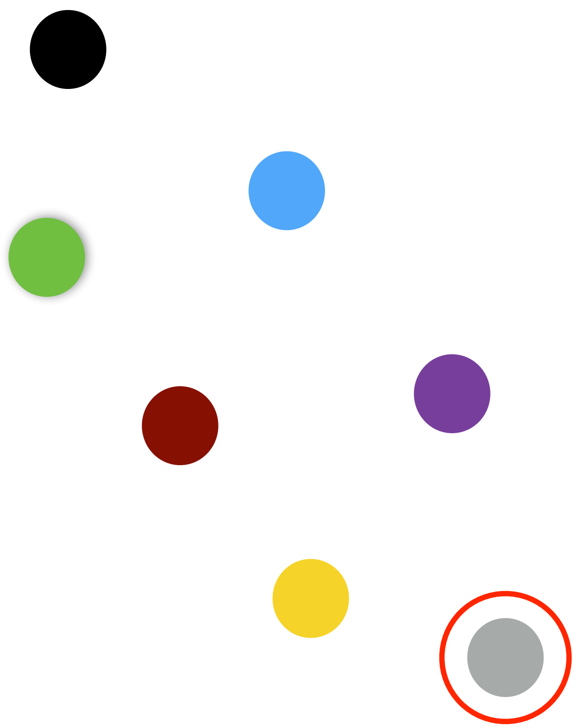


View II

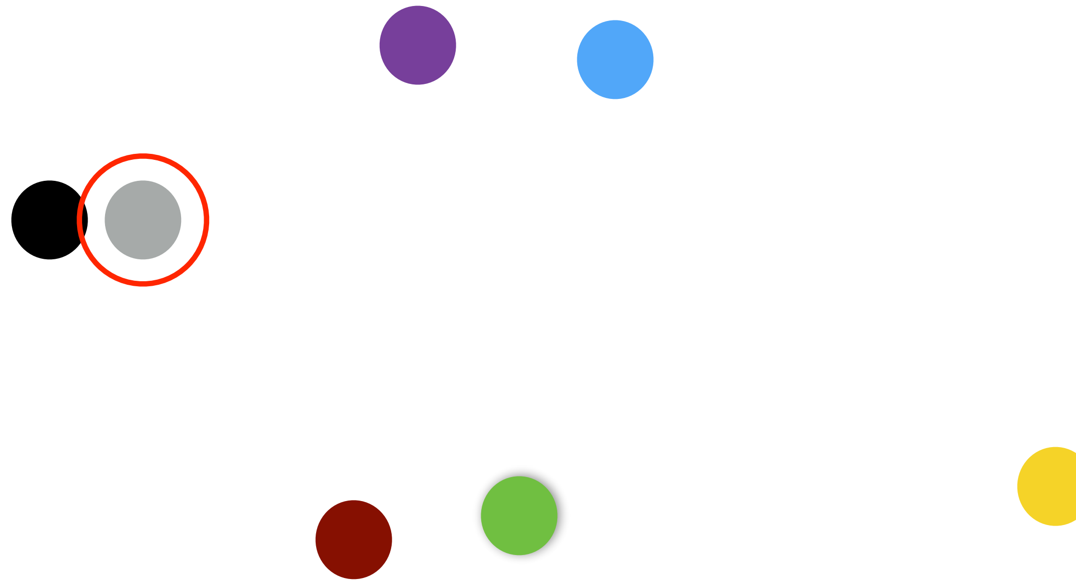
Which Direction to Pick?



Which Direction to Pick?

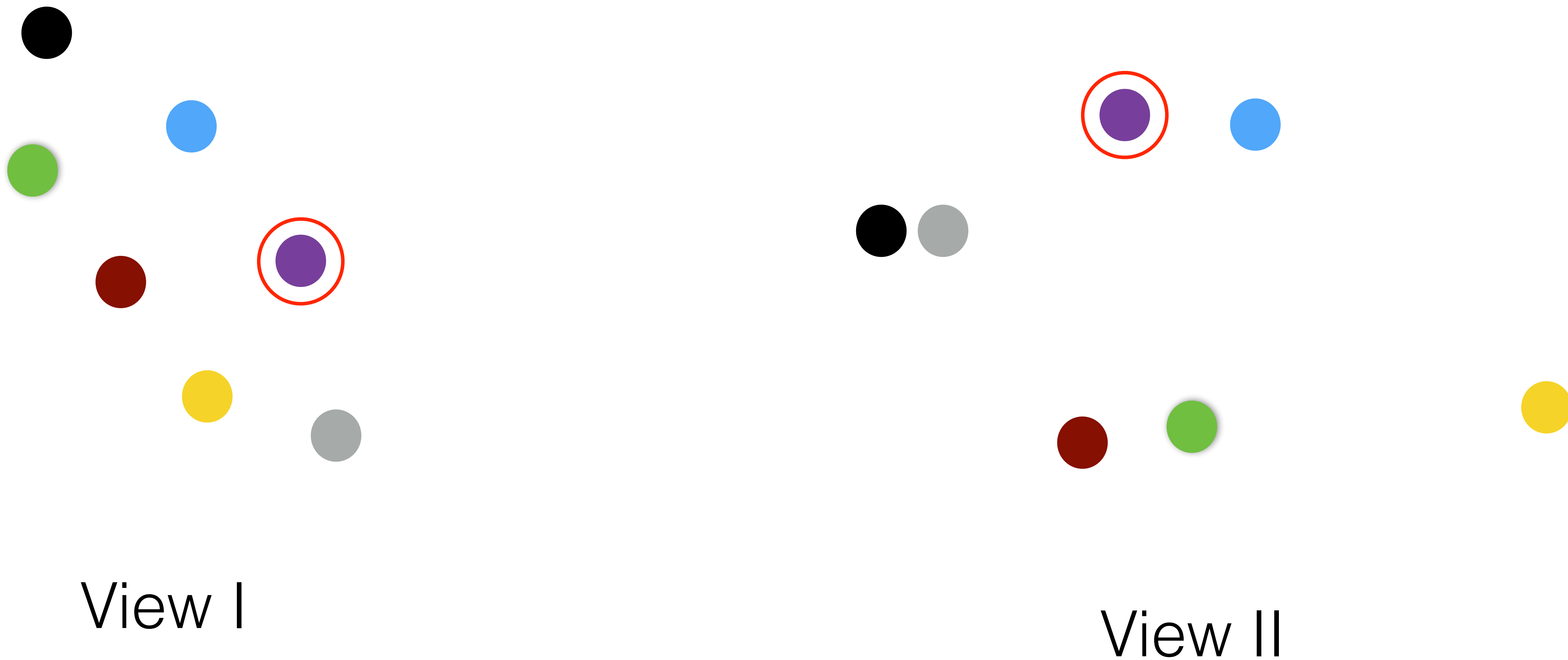


View I

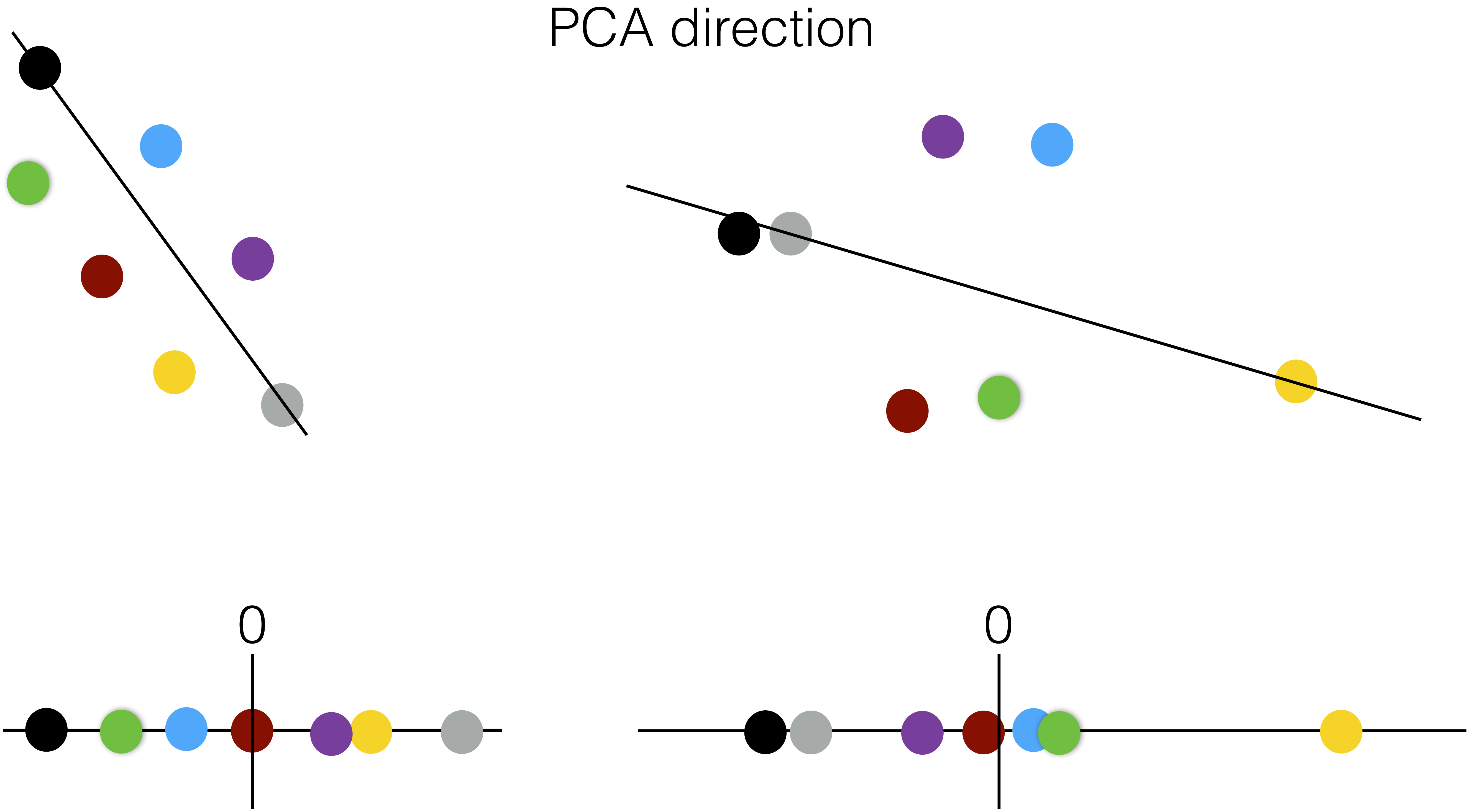


View II

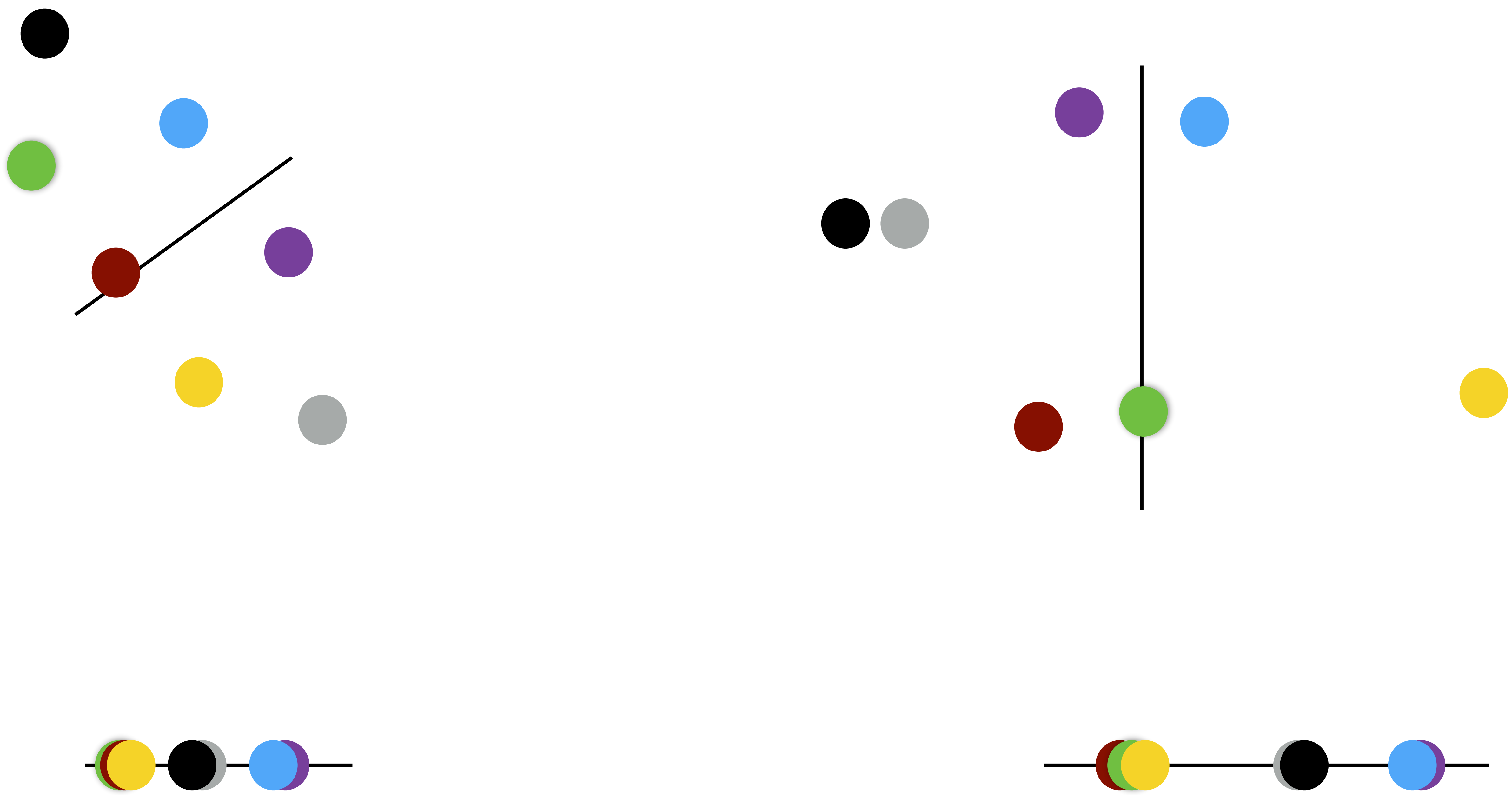
Which Direction to Pick?



Which Direction to Pick?



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

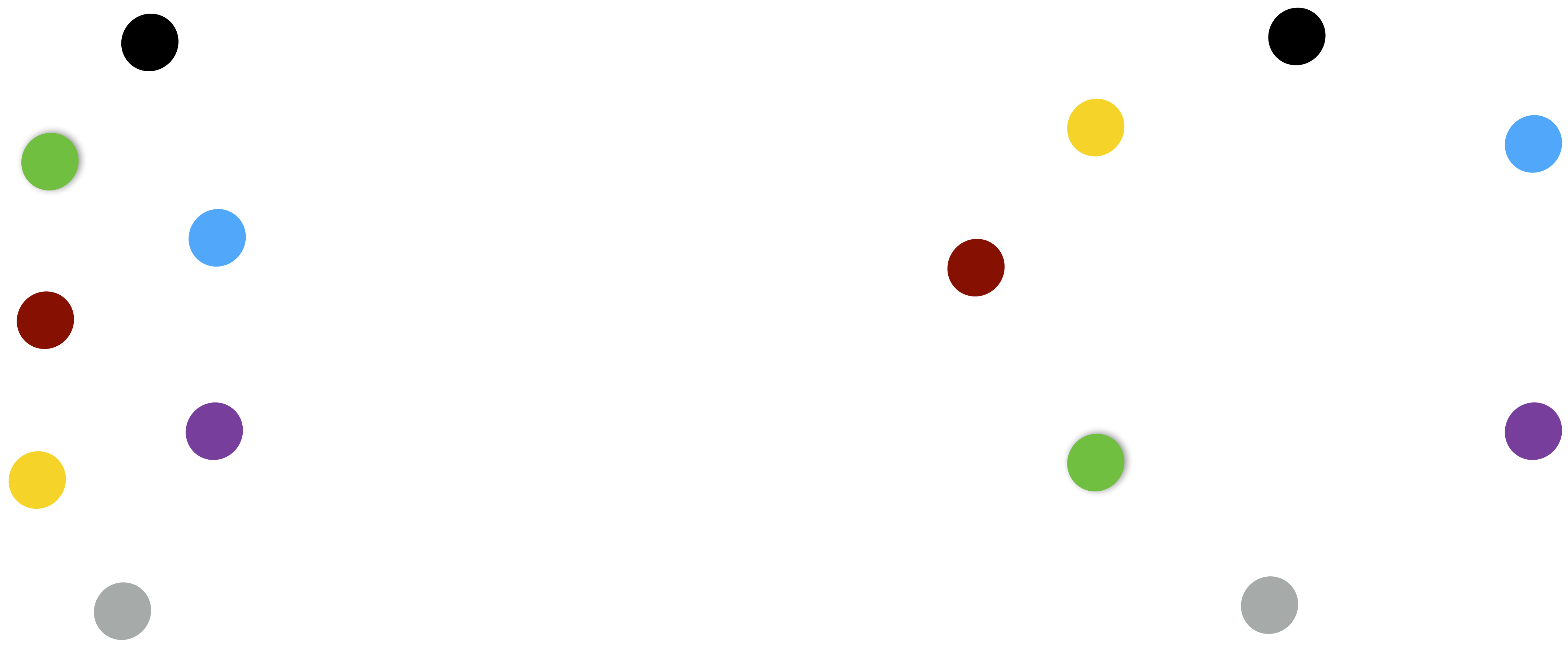
What is the problem with the above?

Why not Maximize Covariance

$$\text{Say } \frac{1}{n}\sum_{t=1}^{n} \mathbf{x}_t[2] \cdot \mathbf{x}'_t[2] > 0$$

Scaling up this coordinate we can blow up covariance

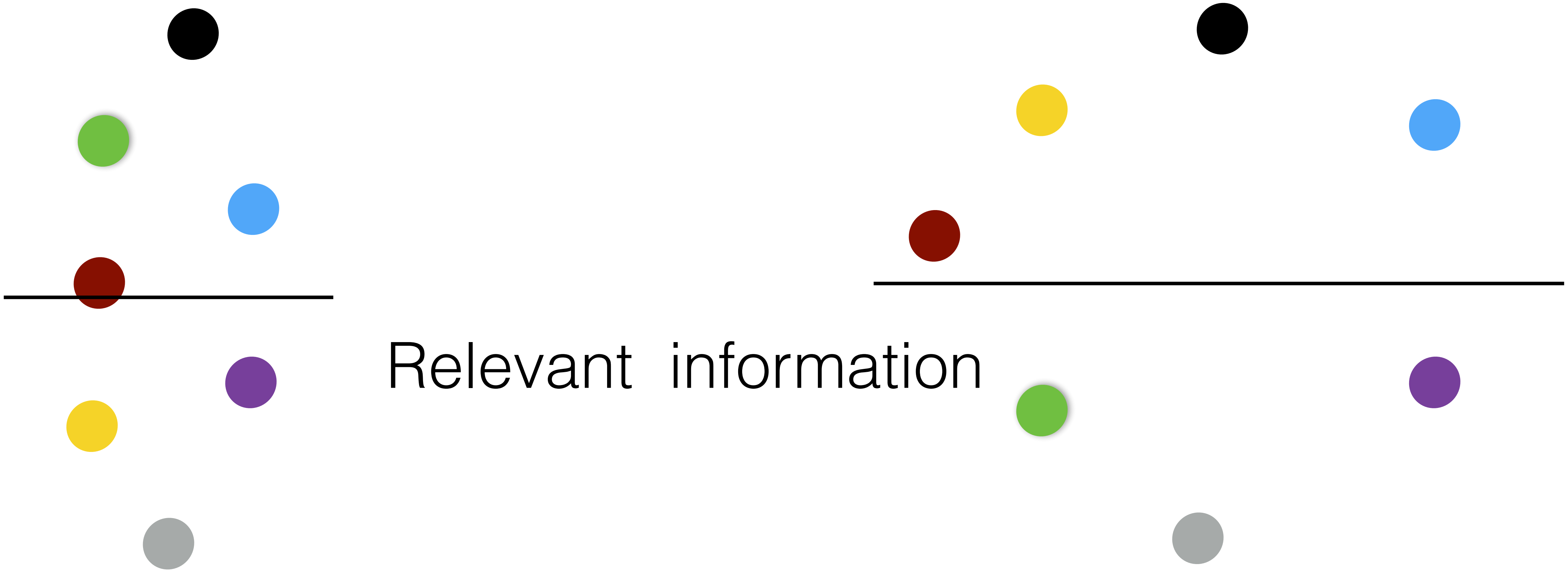
Why not Maximize Covariance



$$\text{Say } \frac{1}{n}\sum_{t=1}^{n} \mathbf{x}_t[2] \cdot \mathbf{x}'_t[2] > 0$$

Scaling up this coordinate we can blow up covariance

Why not Maximize Covariance



$$\text{Say } \frac{1}{n}\sum_{t=1}^{n} \mathbf{x}_t[2] \cdot \mathbf{x}'_t[2] > 0$$

Scaling up this coordinate we can blow up covariance

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,

$$\frac{\frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{y}_t[1] - \frac{1}{n}\sum_{t=1}^{n}\mathbf{y}_t[1]\right)\cdot\left(\mathbf{y}'_t[1] - \frac{1}{n}\sum_{t=1}^{n}\mathbf{y}'_t[1]\right)}{\sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{y}_t[1] - \frac{1}{n}\sum_{t=1}^{n}\mathbf{y}_t[1]\right)^2}\sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{y}'_t[1] - \frac{1}{n}\sum_{t=1}^{n}\mathbf{y}'_t[1]\right)}}$$

Basic Idea of CCA

- Normalize variance in chosen direction to be constant (say 1)
- Then maximize covariance
- This is same as maximizing "correlation coefficient"

Covariance Vs Correlation

- $\text{Covariance}(A, B) = \mathbb{E}[(A - \mathbb{E}[A]) \cdot (B - \mathbb{E}[B])]$

  Depends on the scale of A and B. If B is rescaled, covariance shifts.

- $\text{Corelation}(A, B) = \frac{\mathbb{E}[(A - \mathbb{E}[A]) \cdot (B - \mathbb{E}[B])]}{\sqrt{\text{Var}(A)}\sqrt{\text{Var}(B)}}$

  Scale free.

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

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,

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

s.t. $\frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{y}_t[1]-\frac{1}{n}\sum_{t=1}^{n}\mathbf{y}_t[1]\right)^2 = \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{y}'_t[1]-\frac{1}{n}\sum_{t=1}^{n}\mathbf{y}'_t[1]\right) = 1$

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

Canonical Correlation Analysis

- Hence we want to solve for projection vectors $\mathbf{w}_1$ and $\mathbf{v}_1$ that

$$\text{maximize } \frac{1}{n}\sum_{t=1}^{n} \mathbf{w}_1^\top(\mathbf{x}_t - \mu) \cdot \mathbf{v}_1^\top(\mathbf{x}'_t - \mu')$$

$$\text{subject to } \frac{1}{n}\sum_{t=1}^{n}(\mathbf{w}_1^\top(\mathbf{x}_t - \mu))^2 = \frac{1}{n}\sum_{t=1}^{n}(\mathbf{v}_1^\top(\mathbf{x}'_t - \mu'))^2 = 1$$

where $\mu = \frac{1}{n}\sum_{t=1}^{n}\mathbf{x}_t$ and $\mu' = \frac{1}{n}\sum_{t=1}^{n}\mathbf{x}'_t$

Canonical Correlation Analysis

- Hence we want to solve for projection vectors $\mathbf{w}_1$ and $\mathbf{v}_1$ that

$$\begin{aligned}&\text{maximize } \mathbf{w}_1^\top \Sigma_{1,2} \mathbf{v}_1 \\ &\text{subject to } \mathbf{w}_1^\top \Sigma_{1,1} \mathbf{w}_1 = \mathbf{v}_1^\top \Sigma_{2,2} \mathbf{v}_1 = 1\end{aligned}$$

CANONICAL CORRELATION ANALYSIS

- Hence we want to solve for projection vectors $\mathbf{w}_1$ and $\mathbf{v}_1$ that

$$\text{maximize } \mathbf{w}_1^\top \Sigma_{1,2} \mathbf{v}_1$$
$$\text{subject to } \mathbf{w}_1^\top \Sigma_{1,1} \mathbf{w}_1 = \mathbf{v}_1^\top \Sigma_{2,2} \mathbf{v}_1 = 1$$

$$\Sigma = \begin{matrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{matrix} = \mathrm{cov}\left(X \; X' \right)$$

Solution

Solution

$$W_1 = \text{eigs}\left(\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}, K\right)$$

$$W_2 = \text{eigs}\left(\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}, K\right)$$

CCA Algorithm

CCA Algorithm

1. $$X = \Big({}^{n}\underbrace{X_1}_{d_1},\ \underbrace{X_2}_{d_2} \Big)$$

CCA Algorithm

1. $$X = \left({}^{n}\underbrace{X_1}_{d_1}, \underbrace{X_2}_{d_2} \right)$$

2. $$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix} = \mathrm{cov}\left(X \right)$$

CCA Algorithm

1. $X = \left({}^{n} \underbrace{X_1}_{d_1}, \underbrace{X_2}_{d_2} \right)$

2. $\Sigma = \begin{matrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{matrix} = \mathrm{cov}\left(X \right)$

3. $W_1 = \mathrm{eigs}\left(\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}, K \right)$

CCA Algorithm

1. $X = \left({}^{n} X_1 , X_2 \right)$ with widths d_1, d_2

2. $\Sigma = \begin{matrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{matrix} = \mathrm{cov}\left(X \right)$

3. $W_1 = \mathrm{eigs}\left(\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}, K \right)$

4. $Y_1 = X_1 - \mu_1 \times W_1$

CCA Demo