Machine Learning for Data Science (CS4786)
Lecture 8

Mixture Models, Dimensionality Reduction

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

TOWARDS HARD GAUSSIAN MIXTURE MODEL

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions π^0 randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point
  $$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \; (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})^\top \left(\hat{\Sigma}_j^{m-1}\right)^{-1} (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1}) - \log(\pi_j^{m-1})$$
  2. For each $j \in [K]$, set new representative as
  $$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top \qquad \pi_j^m = \frac{|C_j^m|}{n}$$
  3. $m \leftarrow m + 1$

Towards Hard Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions π^0 randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \ (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})^\top \left(\hat{\Sigma}_j^{m-1}\right)^{-1} (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1}) - \log(\pi_j^{m-1})$$

  2. For each $j \in [K]$, set new representative as

$$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top \qquad \pi_j^m = \frac{|C_j^m|}{n}$$

  3. $m \leftarrow m + 1$

Towards Hard Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions π^0 randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \; (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1})^\top \left(\hat{\Sigma}_j^{m-1}\right)^{-1} (\mathbf{x}_t - \hat{\mathbf{r}}_j^{m-1}) - \log(\pi_j^{m-1})$$

$$d(\mathbf{x}_t, C_j)$$

  2. For each $j \in [K]$, set new representative as

$$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top \qquad \pi_j^m = \frac{|C_j^m|}{n}$$

  3. $m \leftarrow m + 1$

General Hard Mixture Model

- For all $j \in [K]$, initialize π^0 and parameters $\theta_1, \ldots, \theta_K$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point
  $$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\operatorname{argmin}} \; d(\mathbf{x}_t, \theta_j) - \log(\pi_j^{m-1})$$
  2. For each $j \in [K]$, set new representative as
  $$\text{compute } \theta_j \text{ for cluster } C_j \quad \& \quad \pi_j^m = \frac{|C_j^m|}{n}$$
  3. $m \leftarrow m + 1$

Multivariate Gaussian

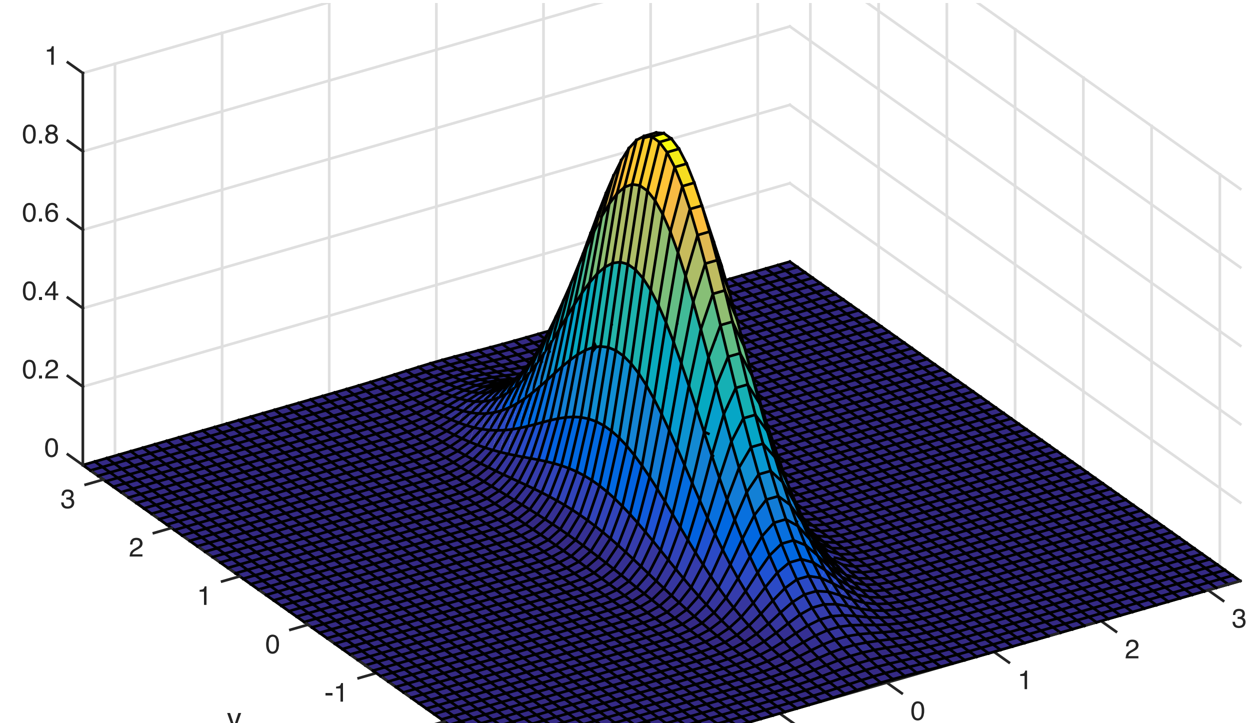
- Two parameters:
  - Mean $\mu \in \mathbb{R}^d$
  - Covariance matrix Σ of size dxd

$$p(x;\mu,\Sigma) = (2\pi)^{d/2}\mathrm{det}(\Sigma)^{-1/2}\exp\left(-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right)$$

Multivariate Gaussian

- Two parameters:
  - Mean $\mu \in \mathbb{R}^d$
  - Covariance matrix Σ of size dxd

$$p(x;\mu,\Sigma) = (2\pi)^{d/2}\det(\Sigma)^{-1/2}\exp\left(-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right)$$



Hard Gaussian Mixture Model

- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$, ellipsoids $\hat{\Sigma}_j^0$ and initial proportions π^0 randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point
  $$\hat{c}^m(\mathbf{x}_t) = \underset{j \in [K]}{\arg\max}\ p(\mathbf{x}_t, \hat{\mathbf{r}}_j^{m-1}, \hat{\Sigma}_j^{m-1}) \times \pi^m(j)$$
  2. For each $j \in [K]$, set new representative as
  $$\hat{\mathbf{r}}_j^m = \frac{1}{|\hat{C}_j^m|} \sum_{\mathbf{x}_t \in \hat{C}_j^m} \mathbf{x}_t \qquad \hat{\Sigma}_j^m = \frac{1}{|C_j|} \sum_{t \in C_j} (\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top \qquad \pi_j^m = \frac{|C_j^m|}{n}$$
  3. $m \leftarrow m + 1$

General Hard Mixture Model

- For all $j \in [K]$, initialize π^0 and parameters $\theta_1, \ldots, \theta_K$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point

$$\hat{c}^m(\mathbf{x}_t) = \operatorname*{arg\,max}_{j \in [K]} \; p(\mathbf{x}_t, \theta_j) \times \pi_j^{m-1}$$

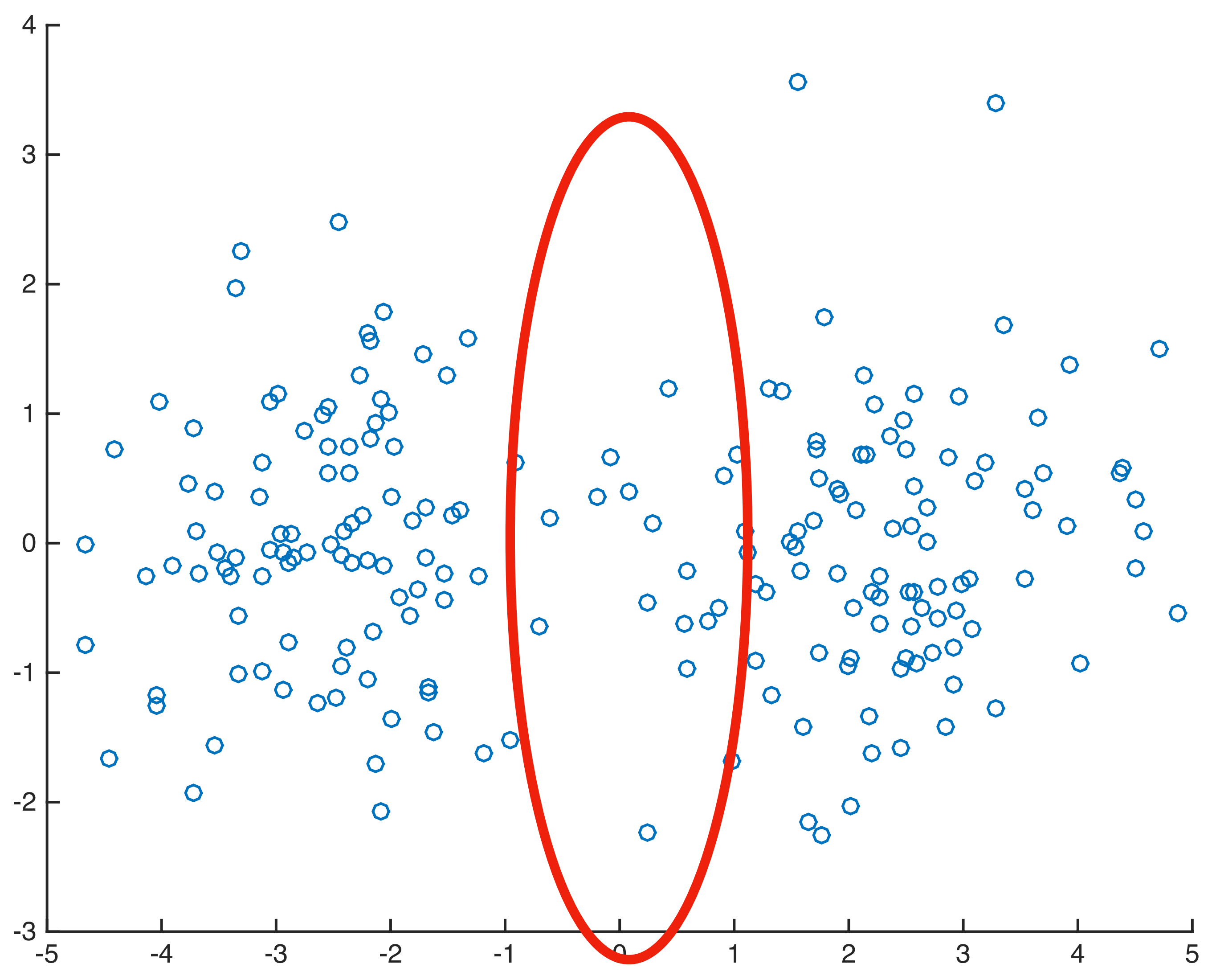
  2. For each $j \in [K]$, set new representative as

$$\text{compute } \theta_j \text{ for cluster } C_j \quad \& \quad \pi_j^m = \frac{|C_j^m|}{n}$$

  3. $m \leftarrow m + 1$

Demo

Pitfall of Hard Assignment



(Soft) Gaussian Mixture Model

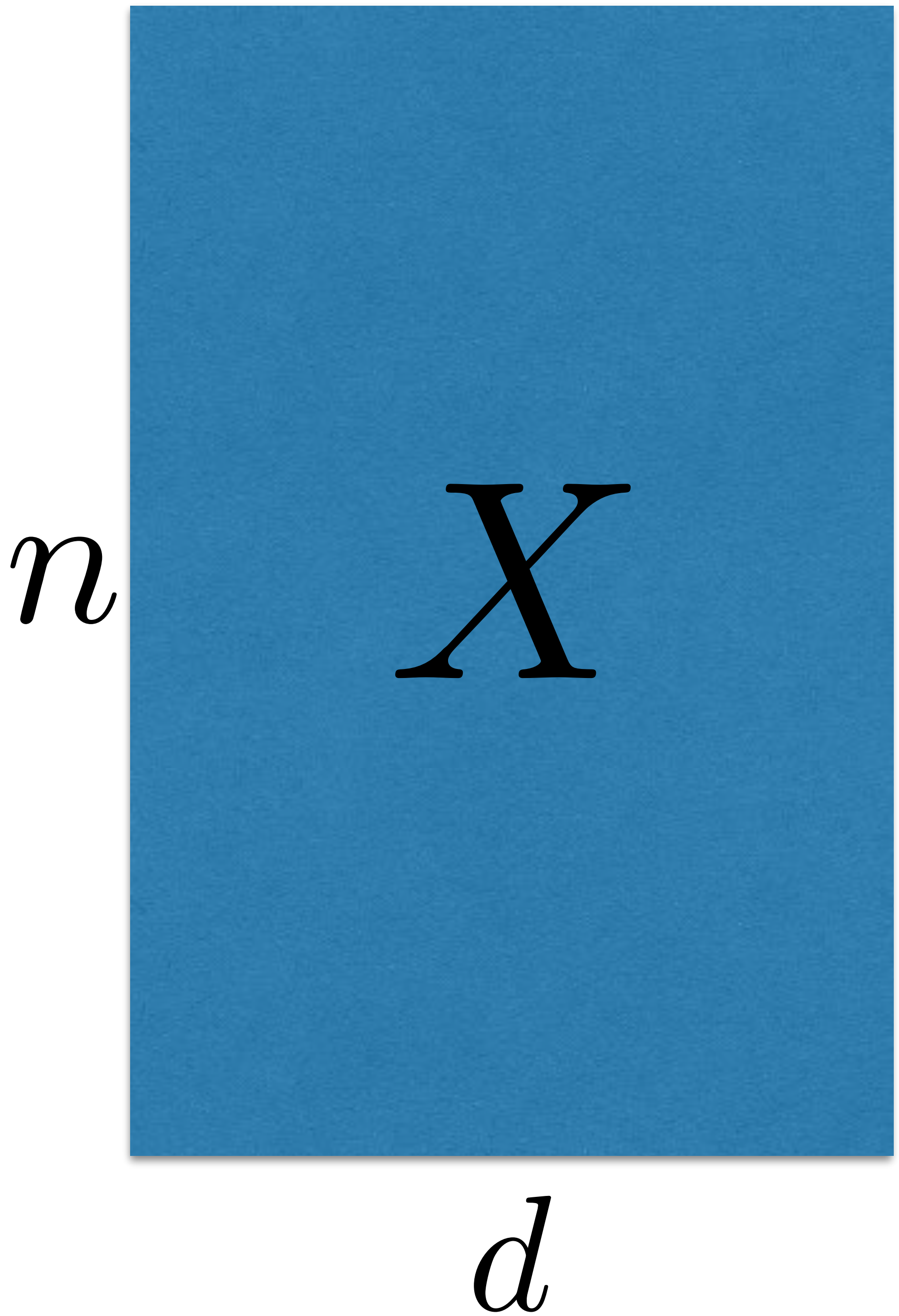
- For all $j \in [K]$, initialize cluster centroids $\hat{\mathbf{r}}_j^0$ and ellipsoids $\hat{\Sigma}_j^0$ randomly and set $m = 1$
- Repeat until convergence (or until patience runs out)
  1. For each $t \in \{1, \ldots, n\}$, set cluster identity of the point
  $$Q_t^m(j) \propto p(\mathbf{x}_t, \hat{\mathbf{r}}_j^{m-1}, \hat{\Sigma}_j^{m-1}) \times \pi^{m-1}(j)$$
  2. For each $j \in [K]$, set new representative as
  $$\hat{\mathbf{r}}_j^m = \frac{\sum_{t=1}^n Q_t(j)\mathbf{x}_t}{\sum_{t=1}^n Q_t(j)} \qquad \hat{\Sigma}_j^m = \frac{\sum_{t=1}^n Q_t(j)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)(\mathbf{x}_t - \hat{\mathbf{r}}_j^m)^\top}{\sum_{t=1}^n Q_t(j)}$$
  $$\pi_j^m = \frac{\sum_{t=1}^n Q_t(j)}{n}$$
  3. $m \leftarrow m + 1$

How to choose K

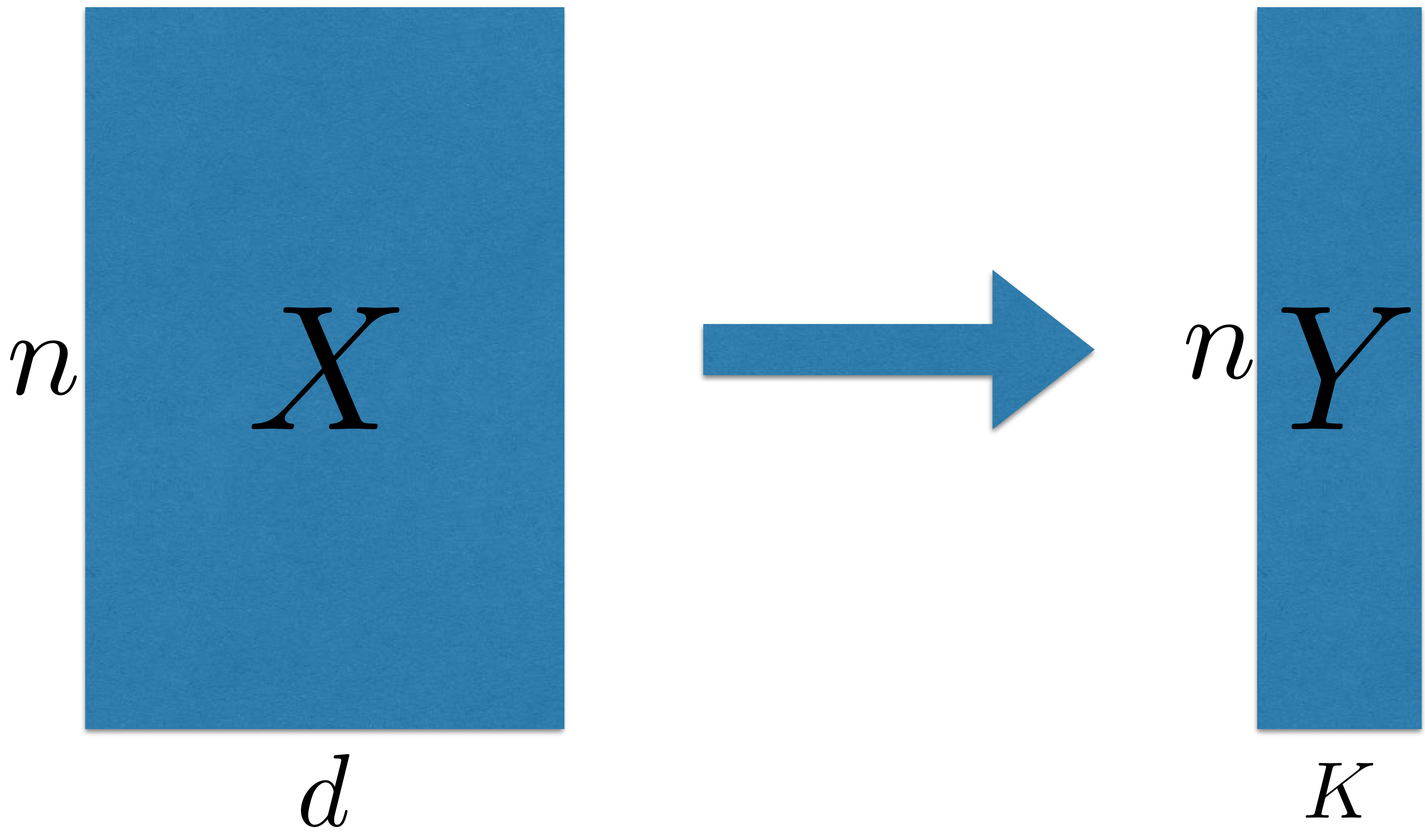
- Elbow method:
  - plot Objective versus K, typically it monotonically decreases.
  - Pick point where there is a kink (explanation in variance is not as much)
  - Intuition: look at rate of change
- Add to objective penalty + p(K) and minimize, where p increases with K
  - intuition we prefer smaller clusters
  - Use prior knowledge to pick p
  - (AIC, BIC etc can been seen to be specific cases)

Dimensionality Reduction

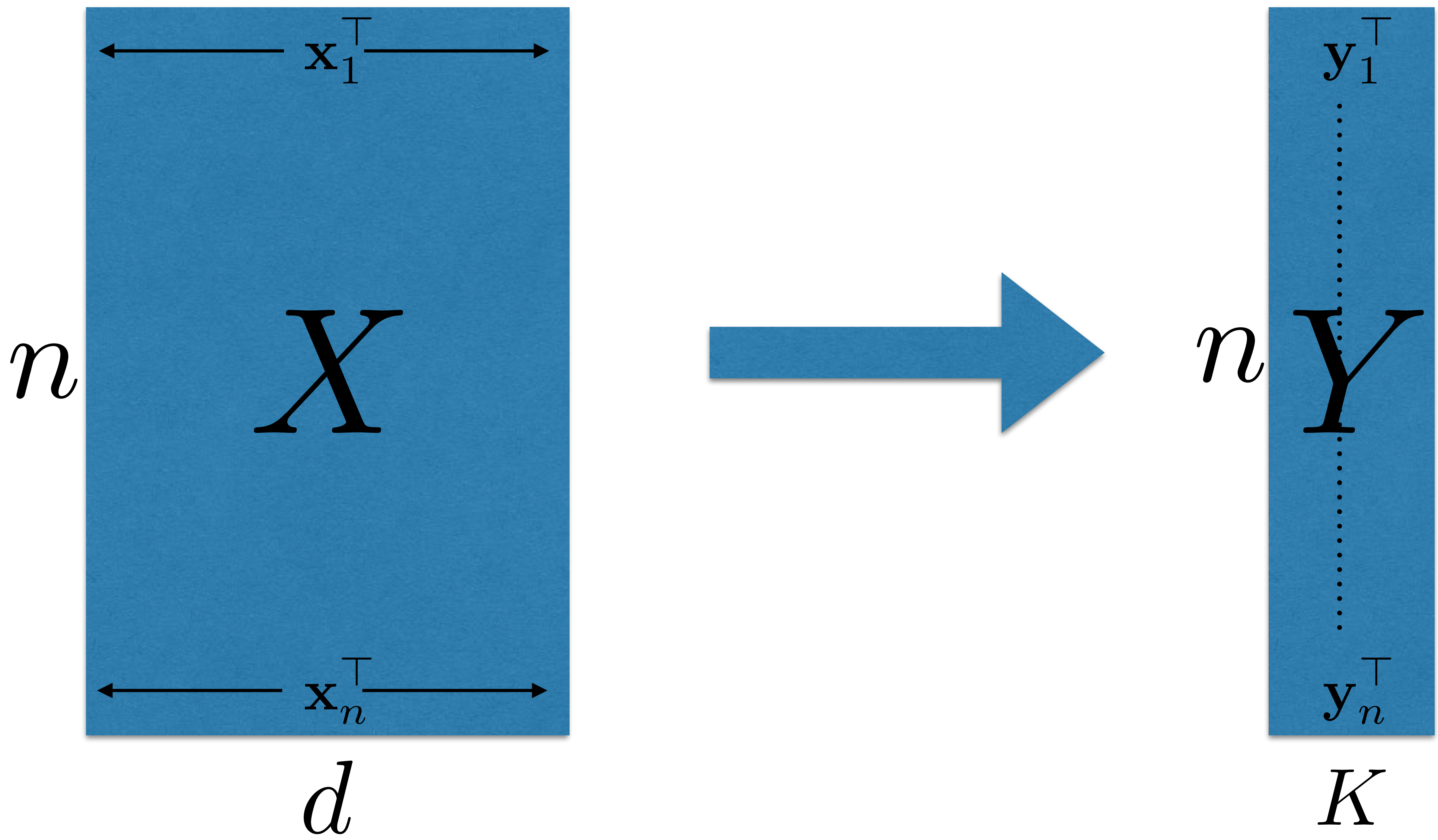
Dimensionality Reduction



Dimensionality Reduction



Dimensionality Reduction



Dimensionality Reduction

- You are provided with n data points each in $\mathbb{R}^d$
- Goal: Compress data into n, points in $\mathbb{R}^K$ where $K << d$
  - Retain as much information about the original data set
  - Retain desired properties of the original data set

WHY DIMENSIONALITY REDUCTION?

- For computational ease
  - As input to supervised learning algorithm
  - Before clustering to remove redundant information and noise
- Data compression & Noise reduction
- Data visualization

Dimensionality Reduction

Given feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^K$ where $K << d$

Dimensionality Reduction

Desired properties:

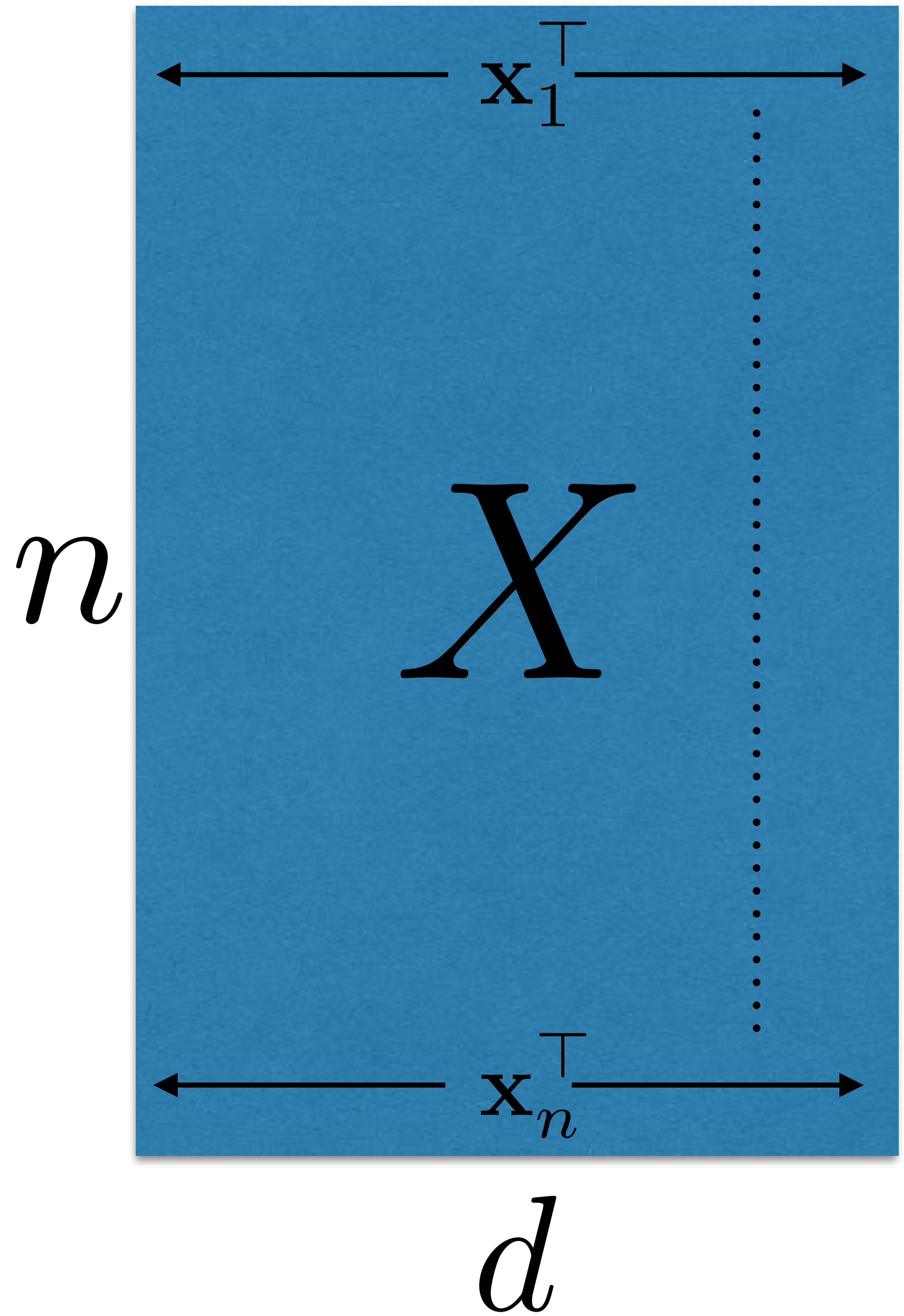
1. Original data can be (approximately) reconstructed
2. Preserve distances between data points
3. “Relevant” information is preserved
4. Noise is reduced

Dim Reduction: Linear Transformation

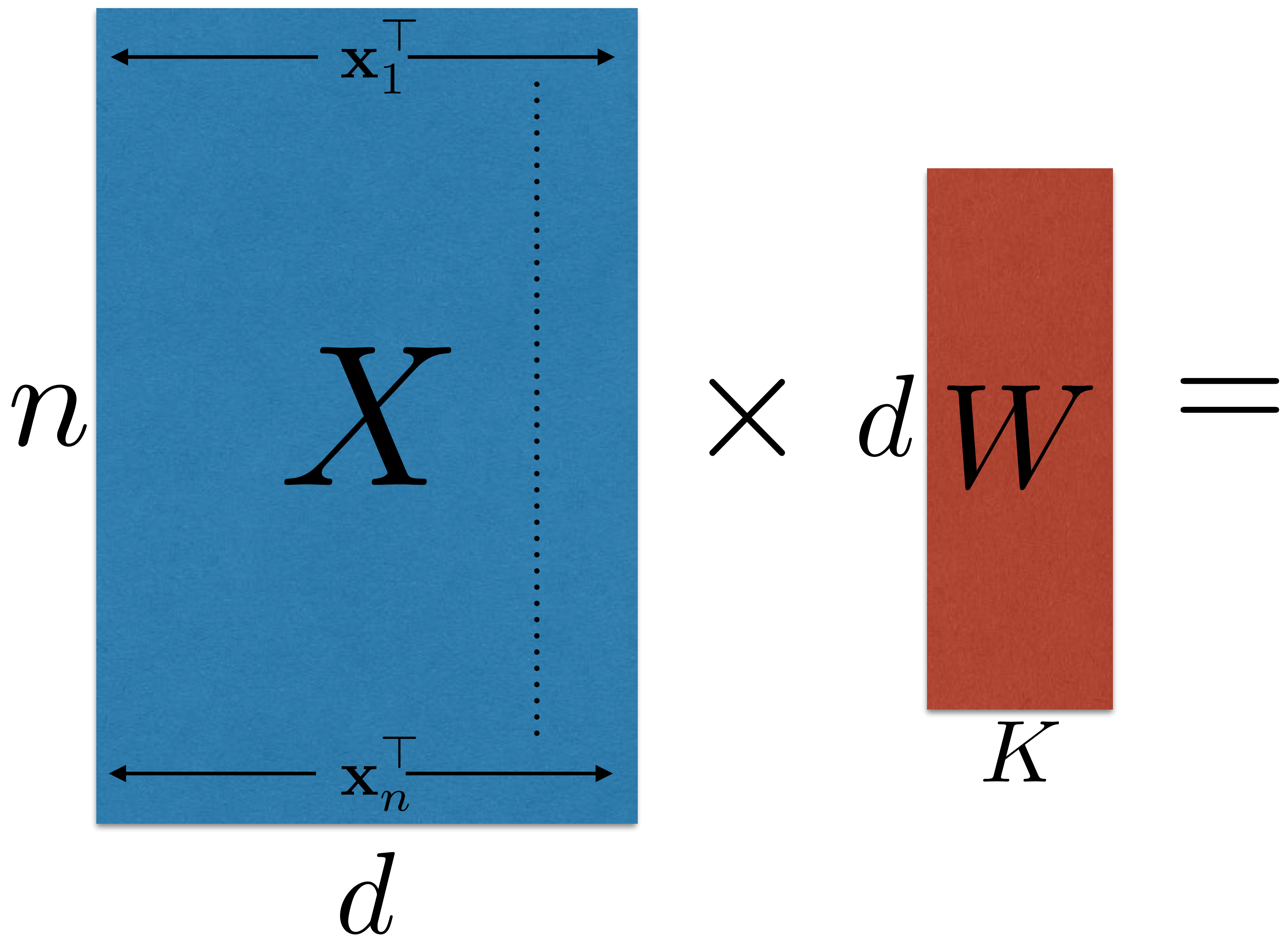
- Pick a low dimensional subspace
- Project linearly to this subspace
- Subspace retains as much information

Dim Reduction: Linear Transformation

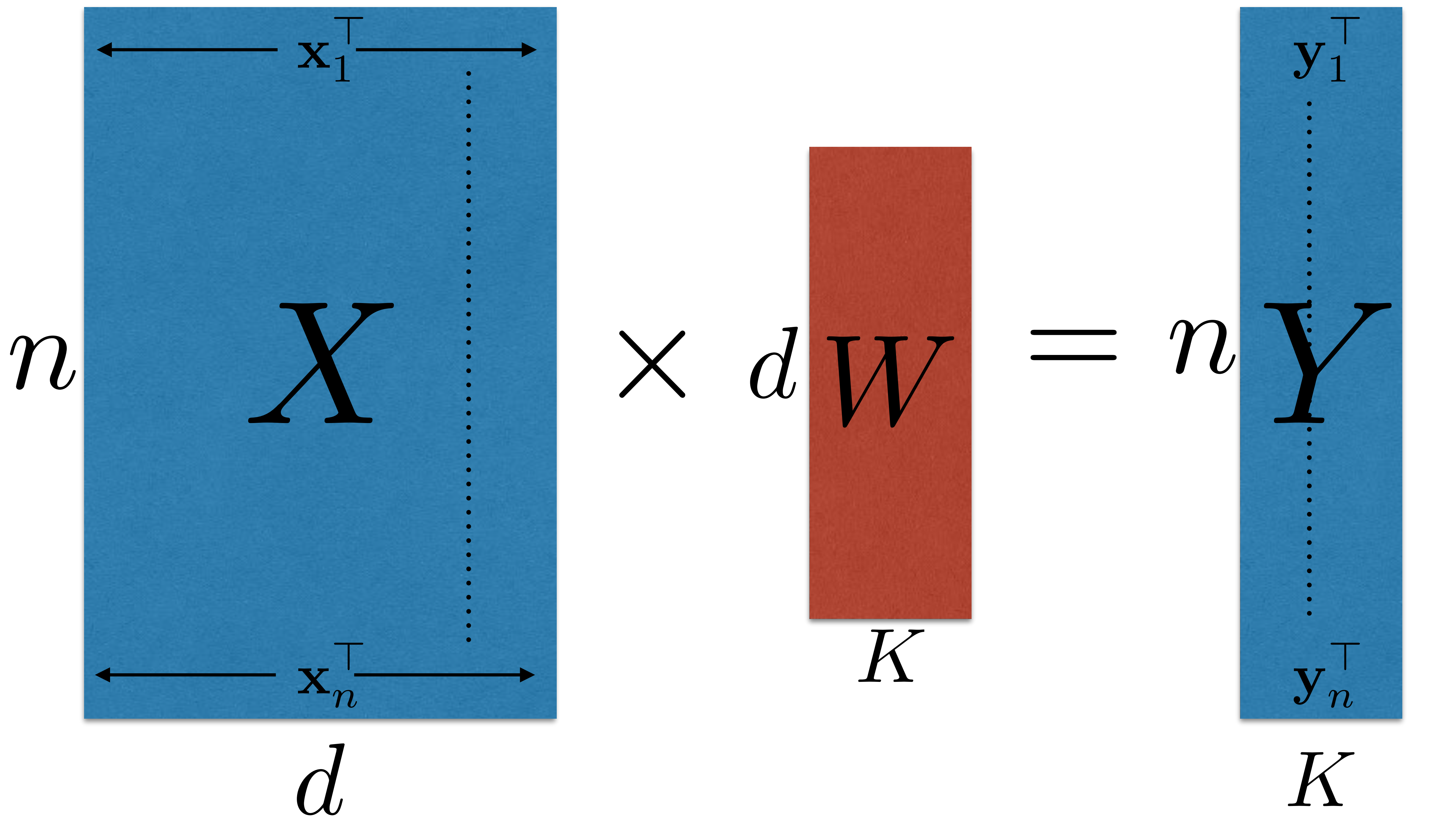
Dim Reduction: Linear Transformation



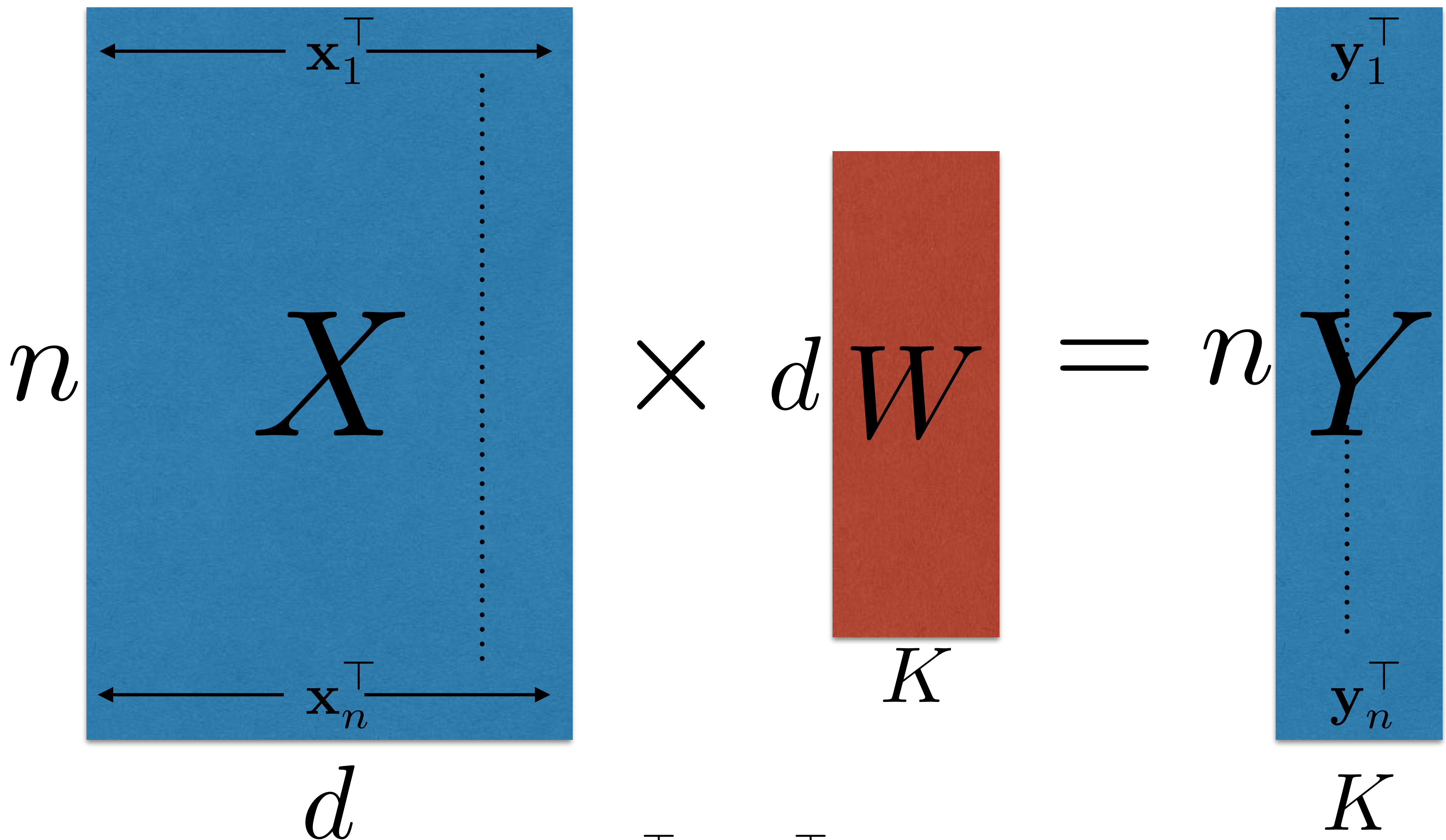
Dim Reduction: Linear Transformation



Dim Reduction: Linear Transformation



Dim Reduction: Linear Transformation

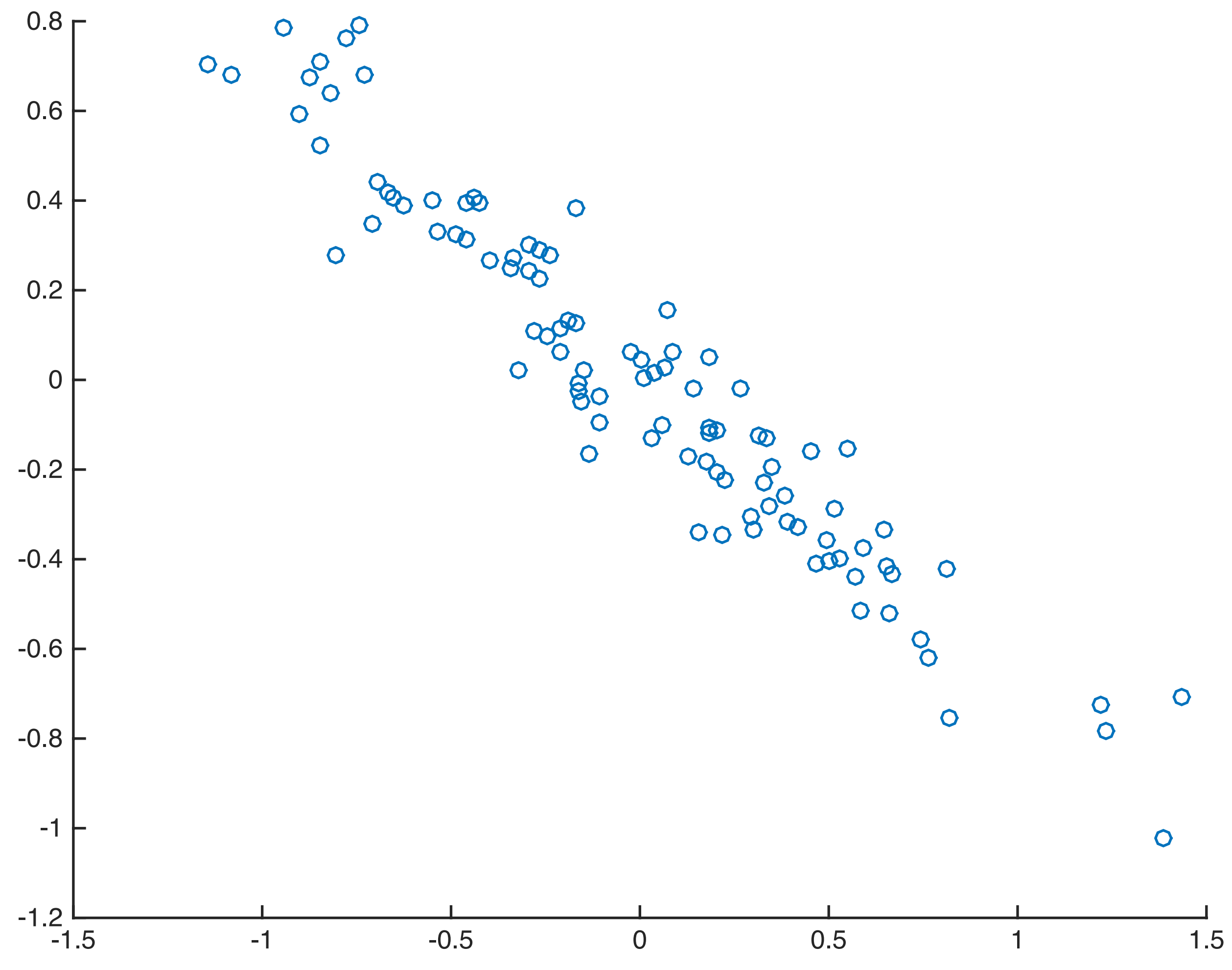


$$\mathbf{y}_i^\top = \mathbf{x}_i^\top W$$

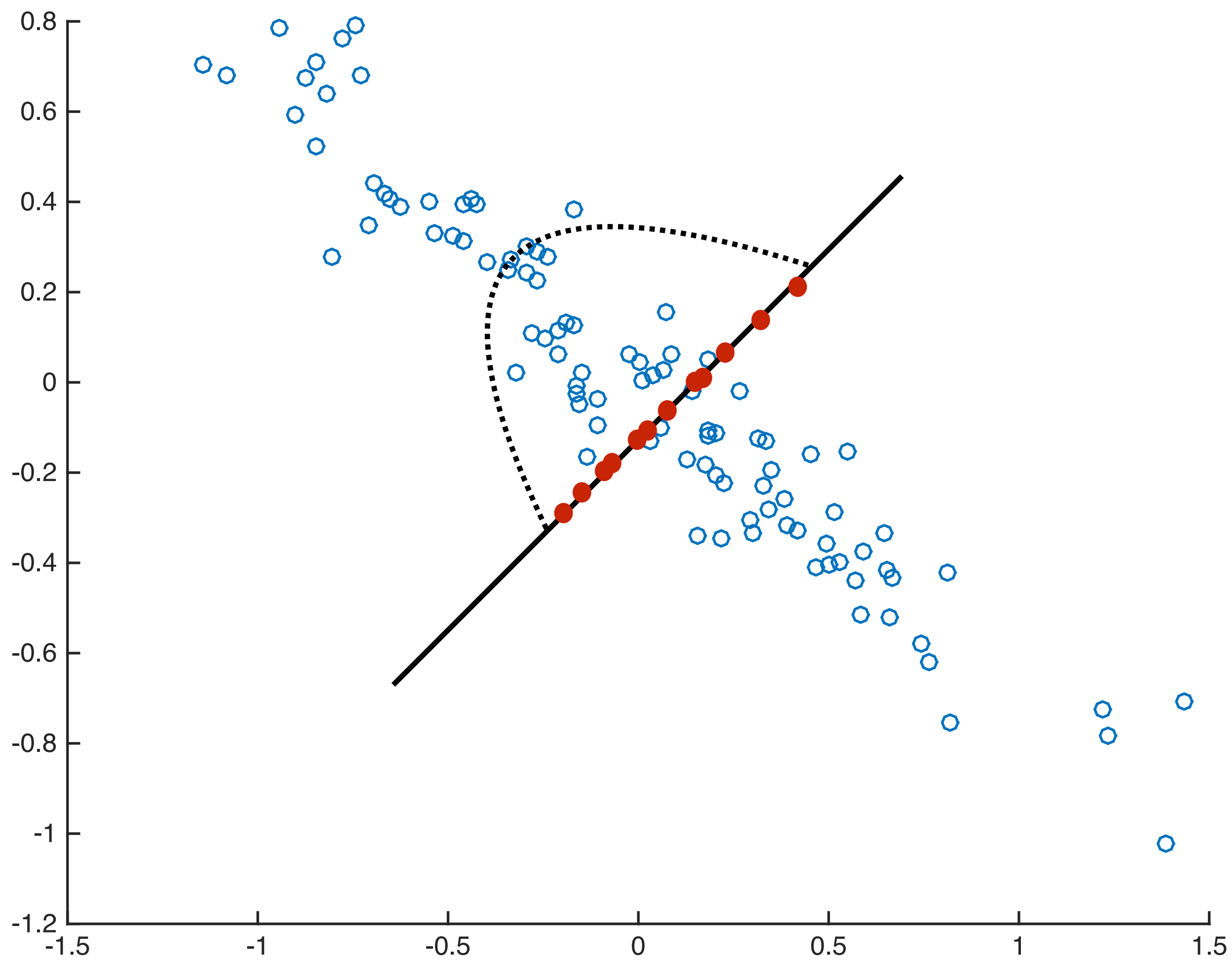
Example:
Students in classroom

Example:
Students in classroom

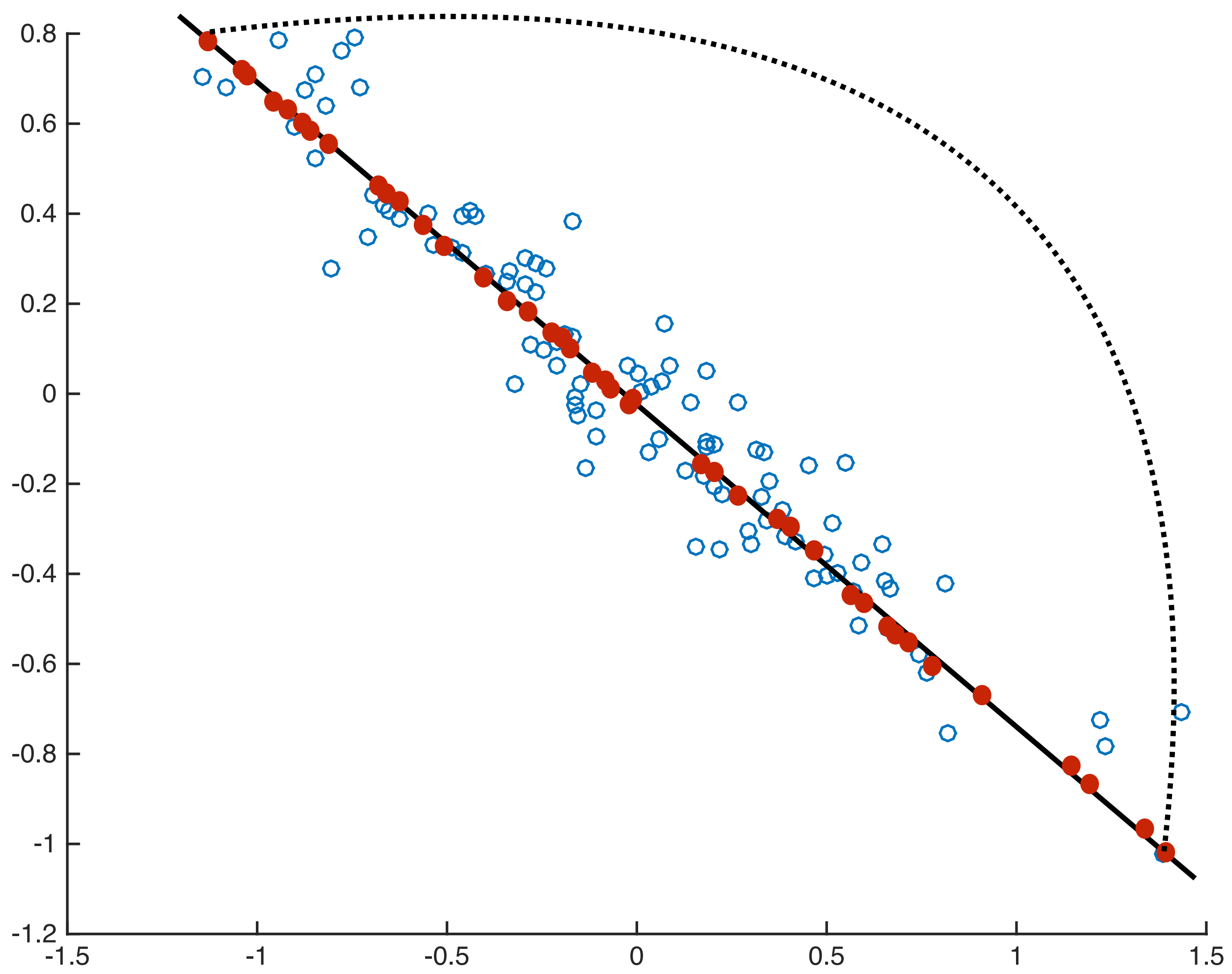
PCA: Variance Maximization



PCA: Variance Maximization



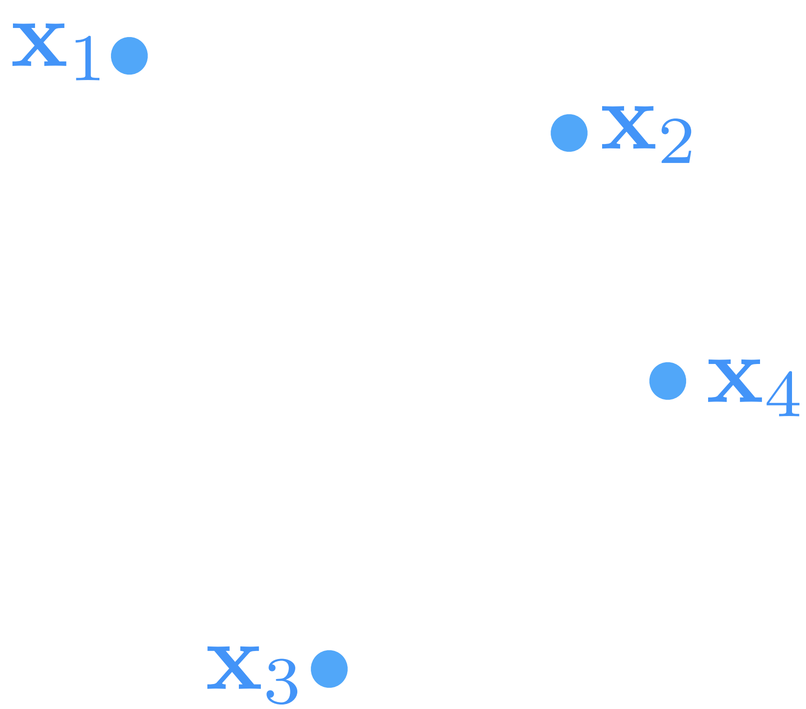
PCA: Variance Maximization



Prelude: reducing to 1 dimension

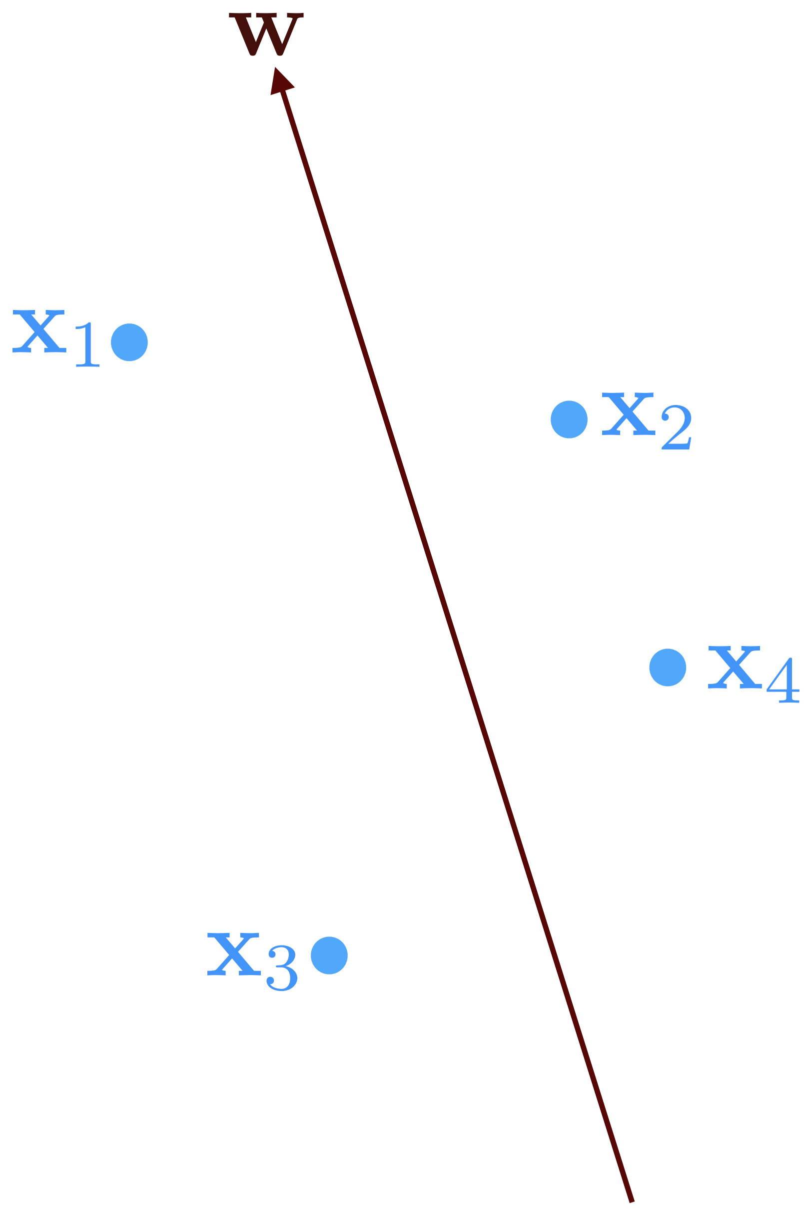
Dim Reduction: Linear Transformation

Prelude: reducing to 1 dimension

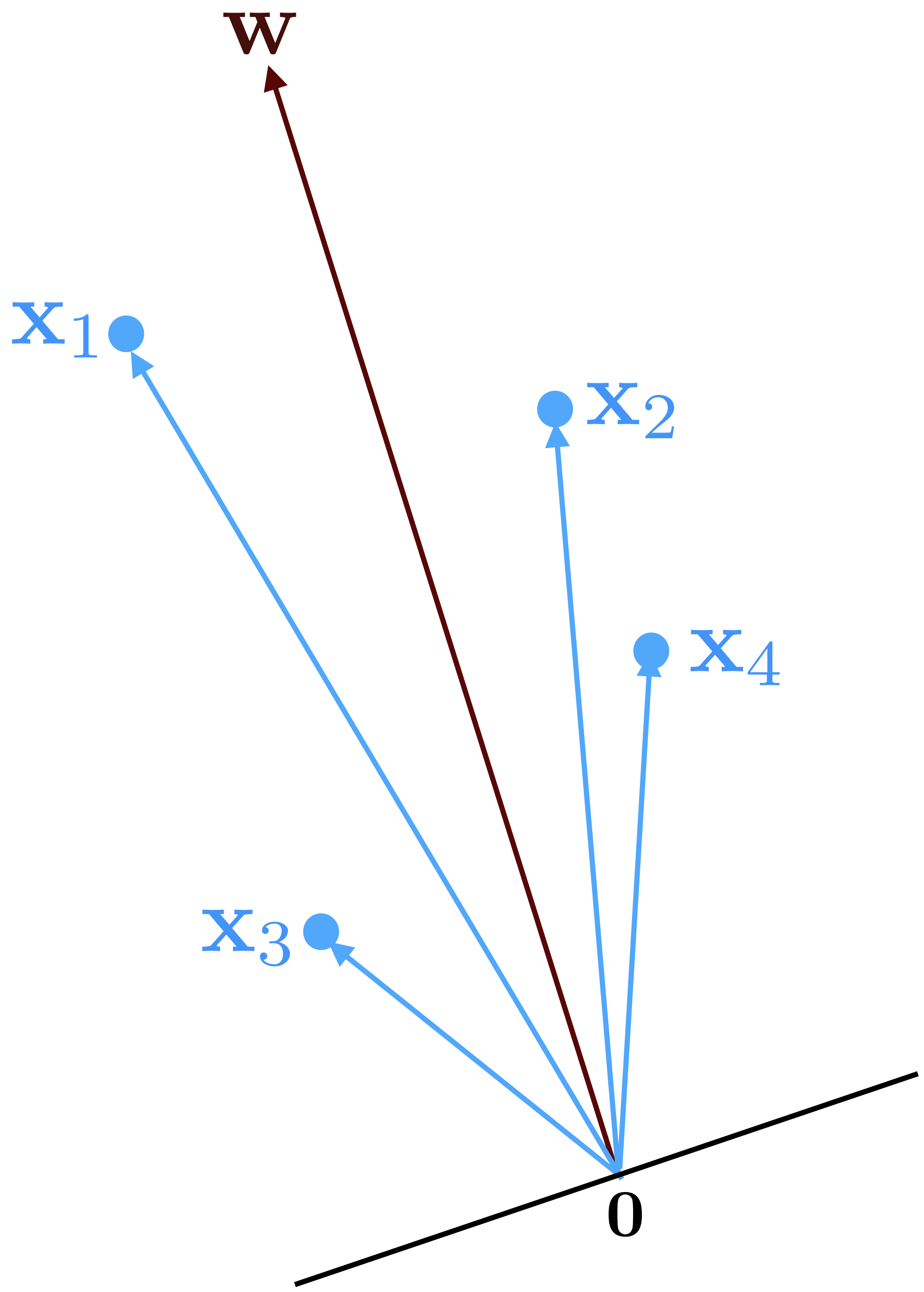


Dim Reduction: Linear Transformation

Prelude: reducing to 1 dimension



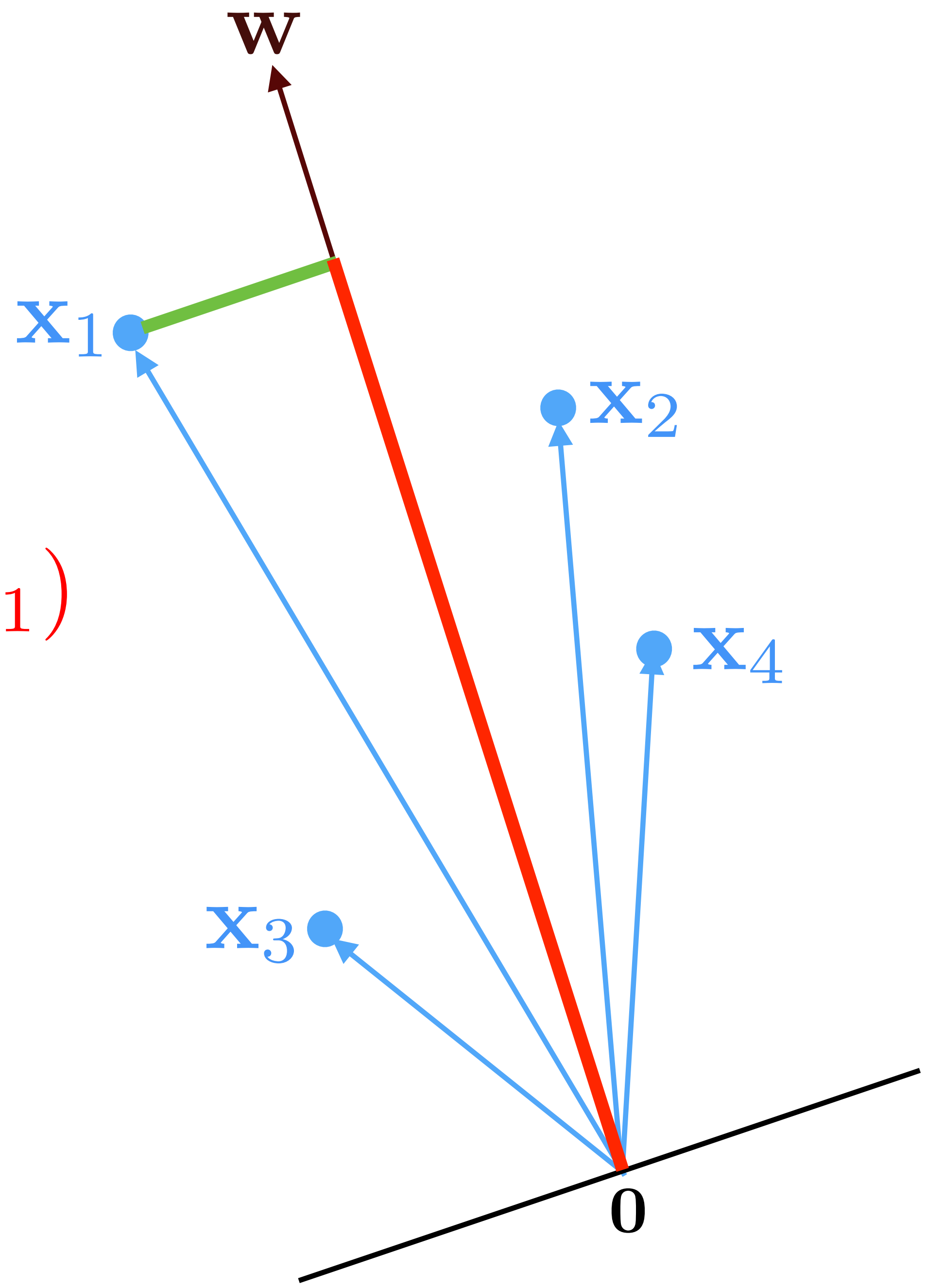
Prelude: reducing to 1 dimension



DIM REDUCTION: LINEAR TRANSFORMATION

Prelude: reducing to 1 dimension

$$\mathbf{y}_1 = \mathbf{w}^\top \mathbf{x}_1 = \|\mathbf{x}_1\| \cos\left(\angle \mathbf{w}\mathbf{x}_1\right)$$



PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{w}^\top\mathbf{x}_t - \frac{1}{n}\sum_{t=1}^{n}\mathbf{w}^\top\mathbf{x}_t\right)^2$$

PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{w}^\top\mathbf{x}_t - \frac{1}{n}\sum_{t=1}^{n}\mathbf{w}^\top\mathbf{x}_t\right)^2$$

$$= \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{w}^\top(\mathbf{x}_t - \mu)\right)^2$$

PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

$$
\begin{aligned}
\mathbf{w}_1 &= \arg \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{w}^\top \mathbf{x}_t - \frac{1}{n}\sum_{t=1}^{n}\mathbf{w}^\top \mathbf{x}_t\right)^2 \\
&= \arg \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{w}^\top (\mathbf{x}_t - \mu)\right)^2 \\
&= \arg \max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\mathbf{w}^\top (\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top \mathbf{w}
\end{aligned}
$$

PCA: Variance Maximization

- Pick directions along which data varies the most
- First principal component:

$$
\begin{aligned}
\mathbf{w}_1 &= \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{w}^\top \mathbf{x}_t - \frac{1}{n}\sum_{t=1}^{n}\mathbf{w}^\top \mathbf{x}_t\right)^2 \\
&= \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\left(\mathbf{w}^\top(\mathbf{x}_t - \mu)\right)^2 \\
&= \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \frac{1}{n}\sum_{t=1}^{n}\mathbf{w}^\top(\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top \mathbf{w} \\
&= \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top \Sigma \mathbf{w}
\end{aligned}
$$

Σ is the covariance matrix

Review

- Review covariance
- Review Eigen vectors

Covariance Matrix

- Its a $d \times d$ matrix, $\Sigma[i,j]$ measures "covariance" of features i and j

$$\Sigma[i,j] = \frac{1}{n}\sum_{t=1}^{n}(\mathbf{x}_t[i] - \mu[i])(\mathbf{x}_t[j] - \mu[j])$$

PCA: Variance Maximization

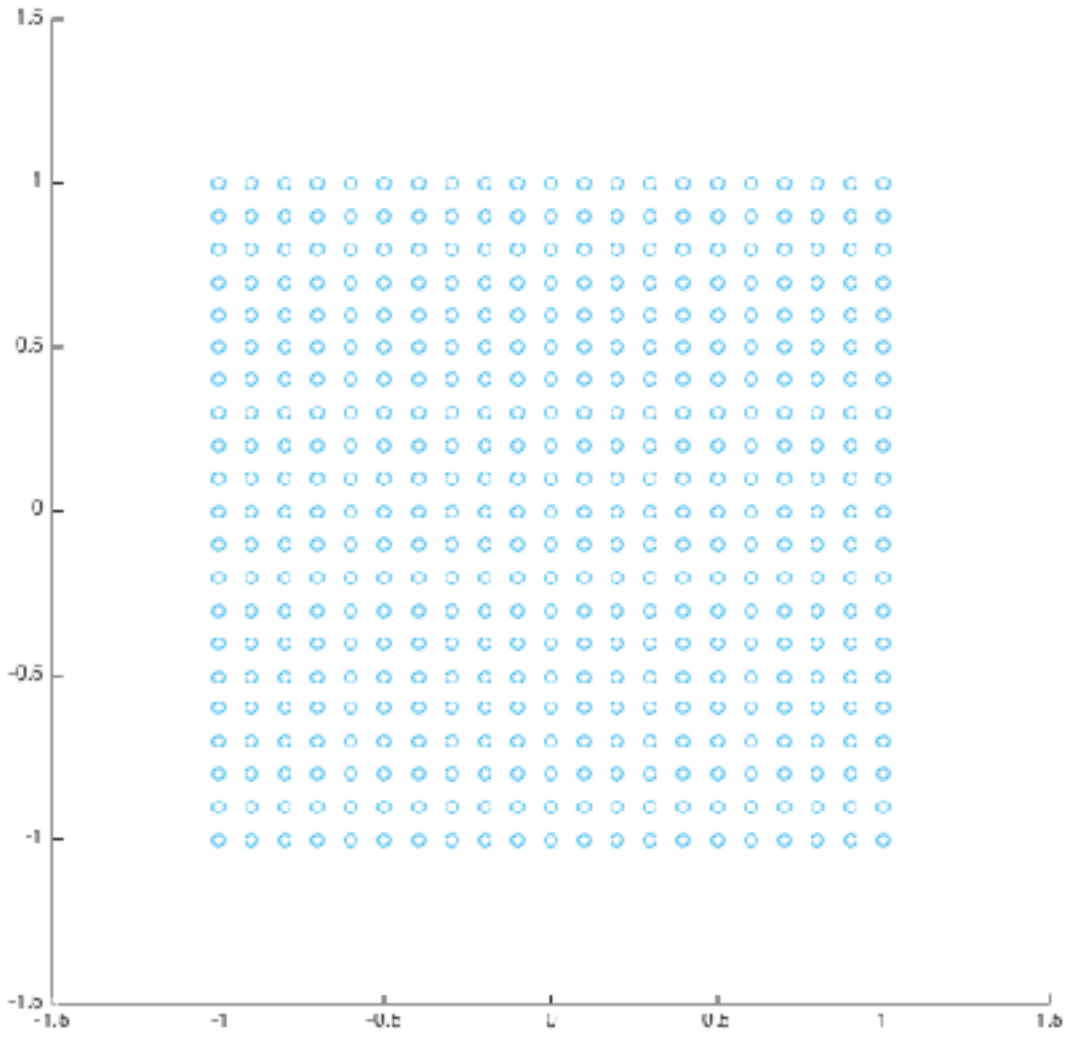
Covariance matrix:

$$\Sigma = \frac{1}{n}\sum_{t=1}^{n}(\mathbf{x}_t - \mu)(\mathbf{x}_t - \mu)^\top$$

- Its a $d \times d$ matrix, $\Sigma[i,j]$ measures "covariance" of features i and j

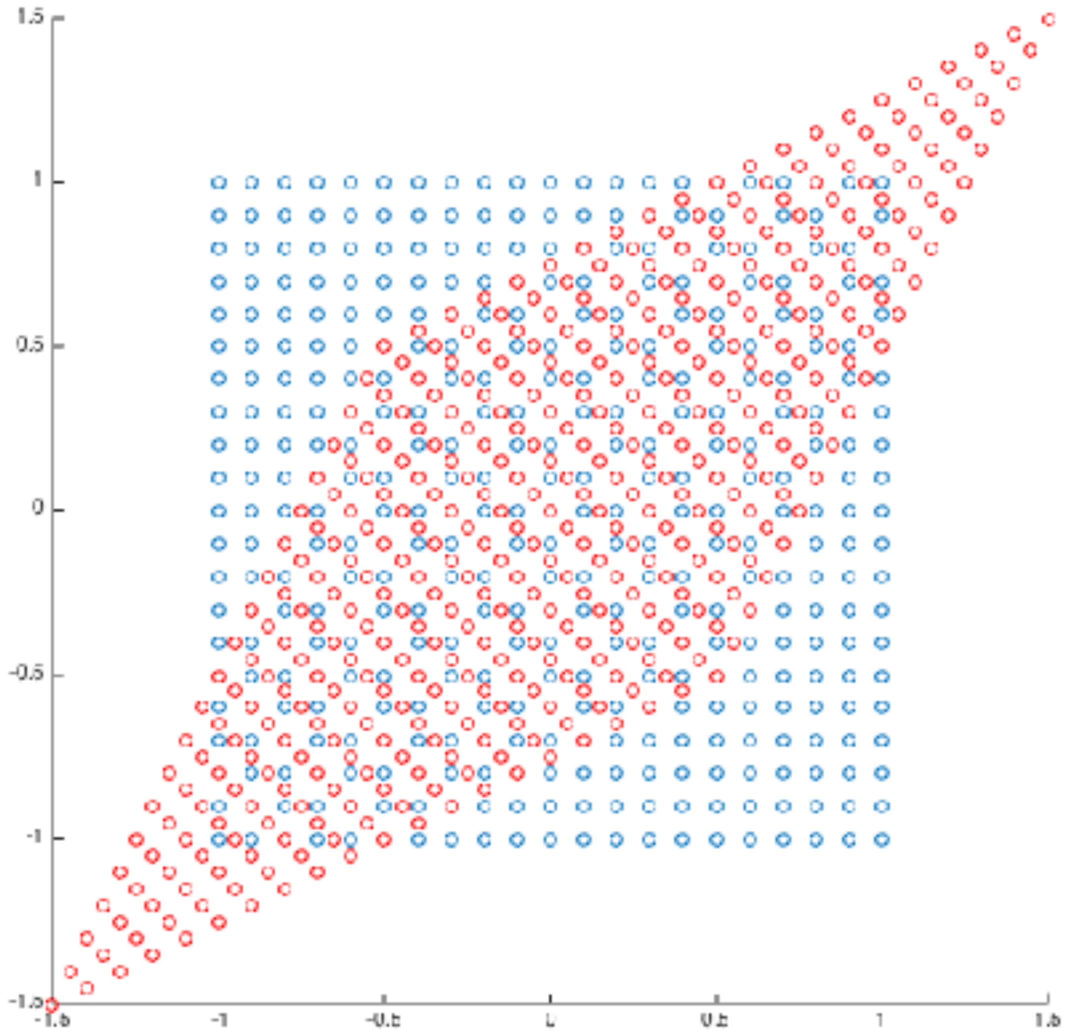
$$\Sigma[i,j] = \frac{1}{n}\sum_{t=1}^{n}(\mathbf{x}_t[i] - \mu[i])(\mathbf{x}_t[j] - \mu[j])$$

What are Eigen Vectors?



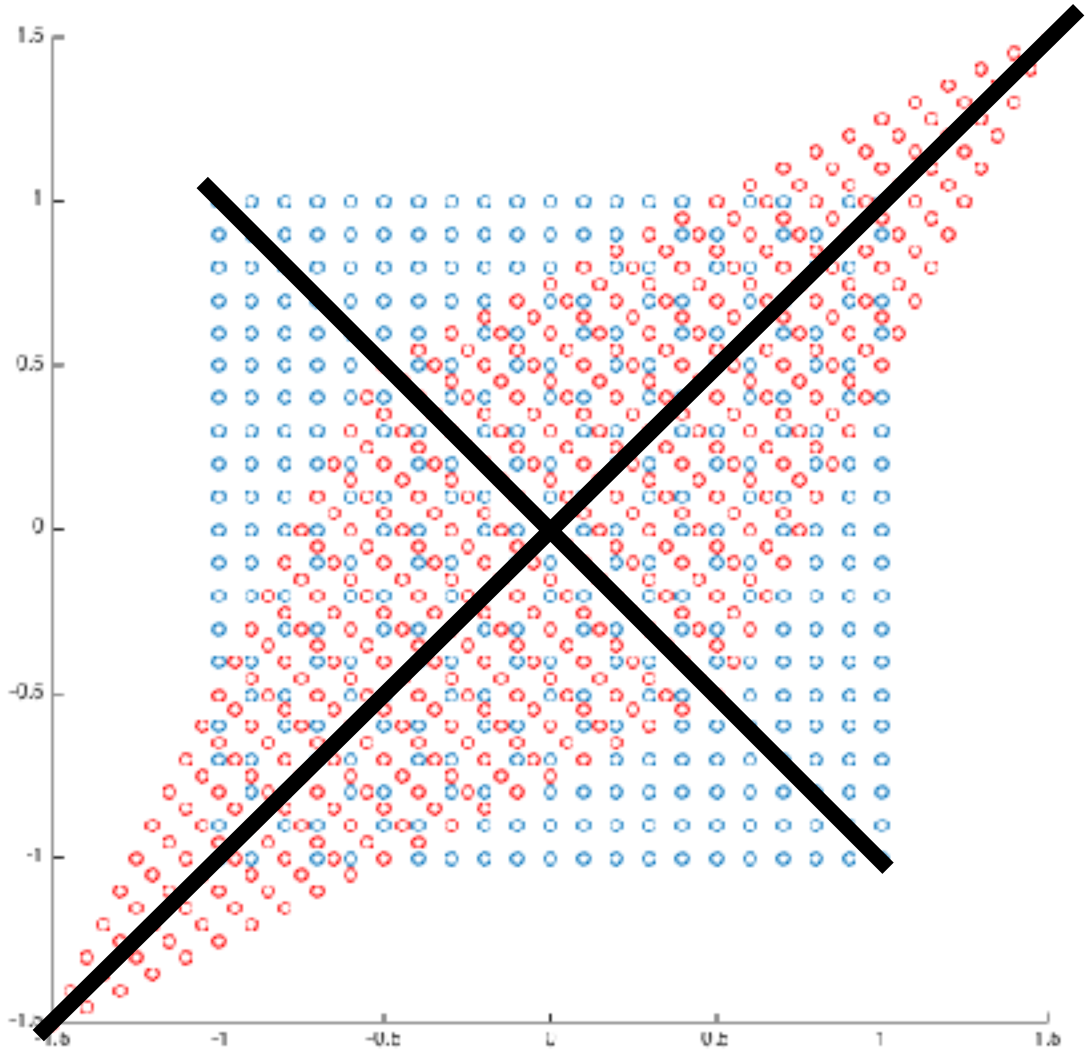
What are Eigen Vectors?

$$x \mapsto Ax$$



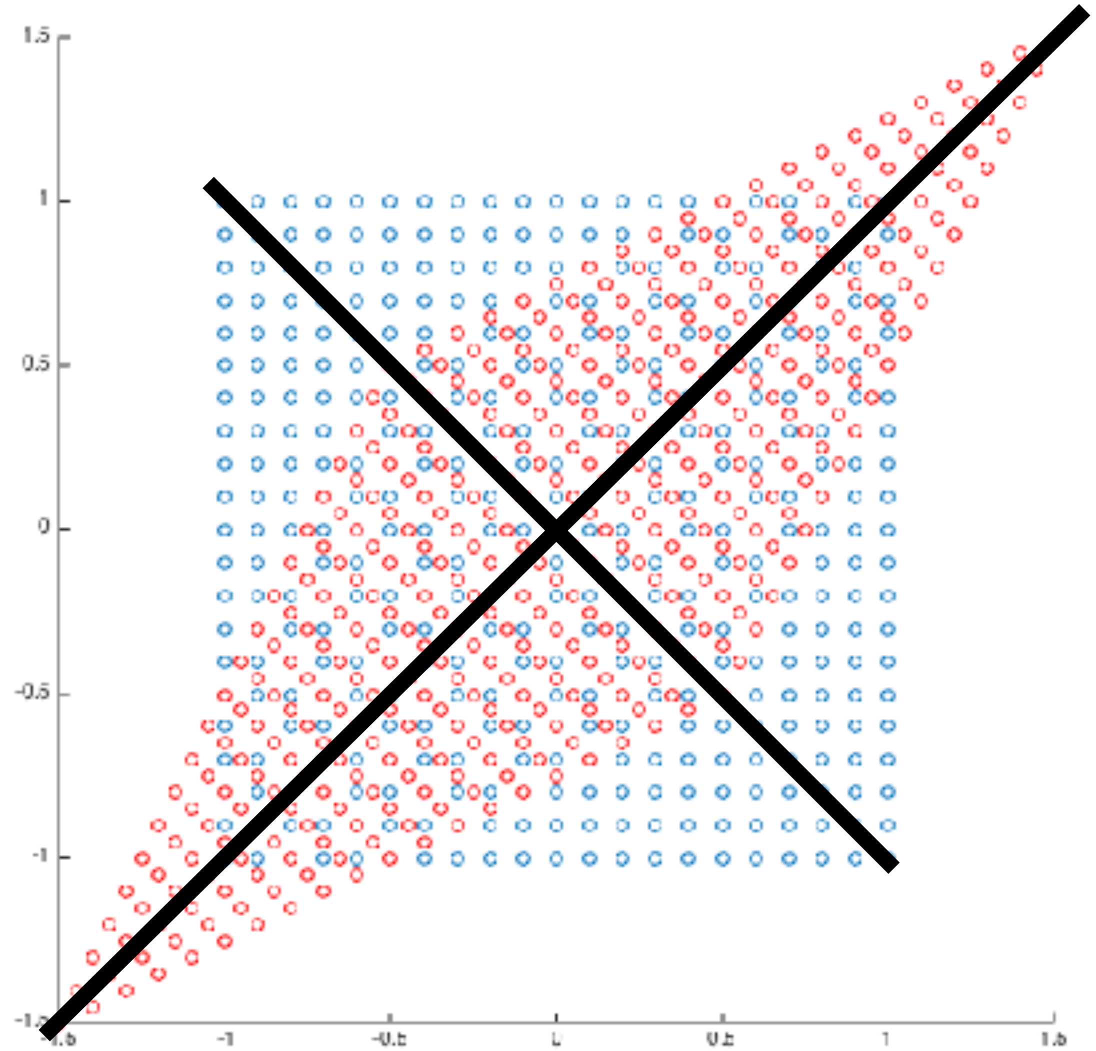
What are Eigen Vectors?

$$x \mapsto Ax$$



What are Eigen Vectors?

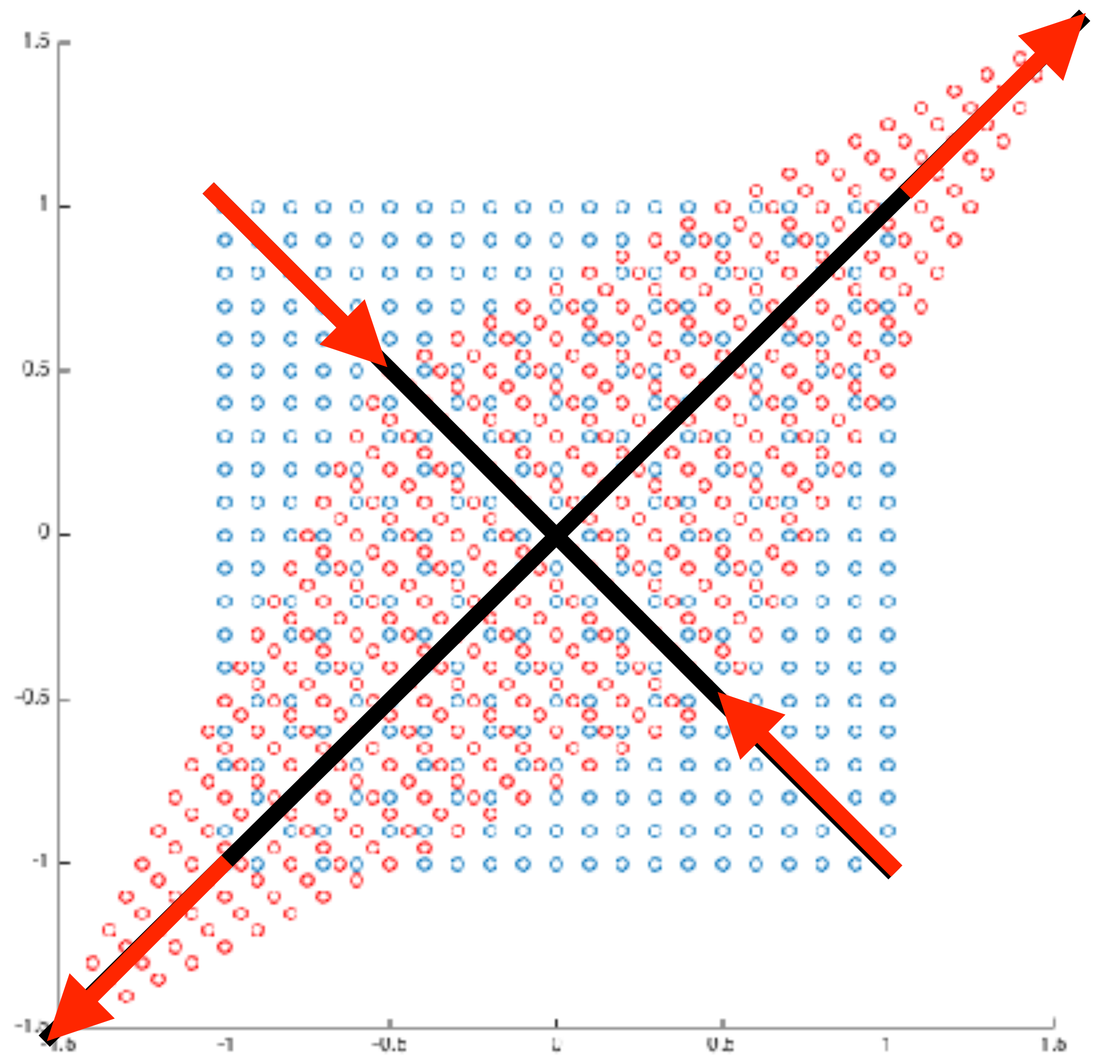
$$x \mapsto Ax$$



$$A\mathbf{x} = \lambda\mathbf{x}$$

What are Eigen Vectors?

$$x \mapsto Ax$$



$$A\mathbf{x} = \lambda\mathbf{x}$$