Machine Learning for Data Science (CS4786) Lecture 2

Dimensionality Reduction
&
Principal Component Analysis

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2016sp/

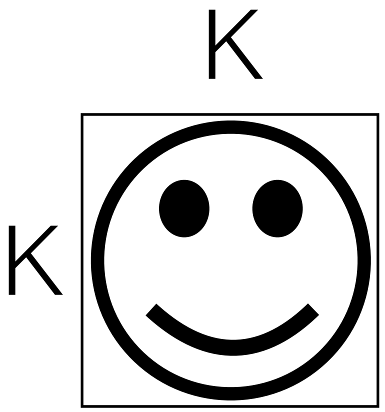
Announcements

- Diagnostic assignment due on 4th Feb (Thursday) beginning of class
- Course webpage is the official source of all class related information
- You will be added to CMS once you return Assignment 0 with your net-id on it.

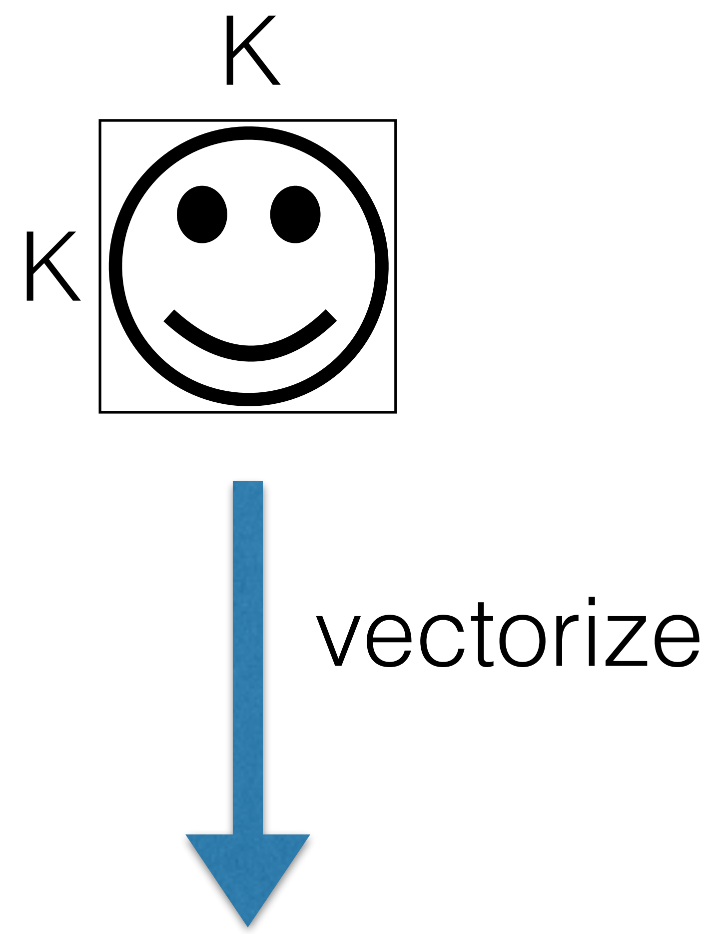
Representing Data as Feature Vectors

- How do we represent data?
- Each data-point often represented as vector referred to as feature vector
- Eg. text document represented by vector in which each coordinate represents a word and value represents number of times the word occurred in the document
- Eg. Image represented as a vector where each coordinate represents a pixel and value represents the grayscale value of that pixel

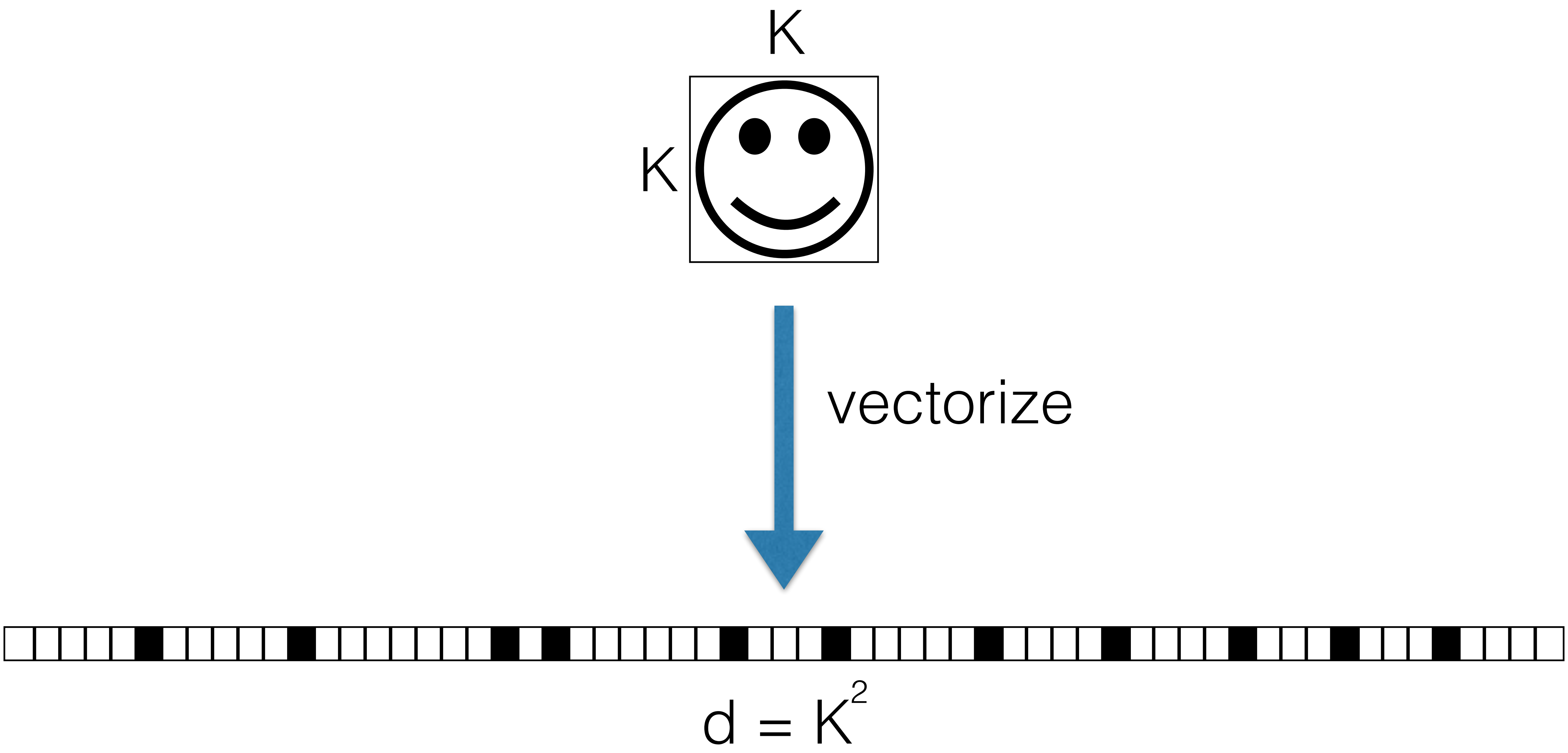
Example: Images



Example: Images



EXAMPLE: IMAGES



Example: Text (Bag of Words)

Documents:

car
engine
hood
tires
truck
trunk

car
emissions
hood
make
model
trunk

Chomsky
corpus
noun
parsing
tagging
wonderful

Example: Text (Bag of Words)

Documents:

car
engine
hood
tires
truck
trunk

car
emissions
hood
make
model
trunk

Chomsky
corpus
noun
parsing
tagging
wonderful

car	Chomsky	corpus	emissions	engine	hood	make	model	noun	parsing	tagging	tires	truck	trunk	wonderful
1	0	0	0	1	1	0	0	0	0	0	1	1	1	0
1	0	0	1	0	1	1	1	0	0	0	0	0	1	0
0	1	1	0	0	0	0	0	1	1	1	0	0	0	1

Dimensionality Reduction

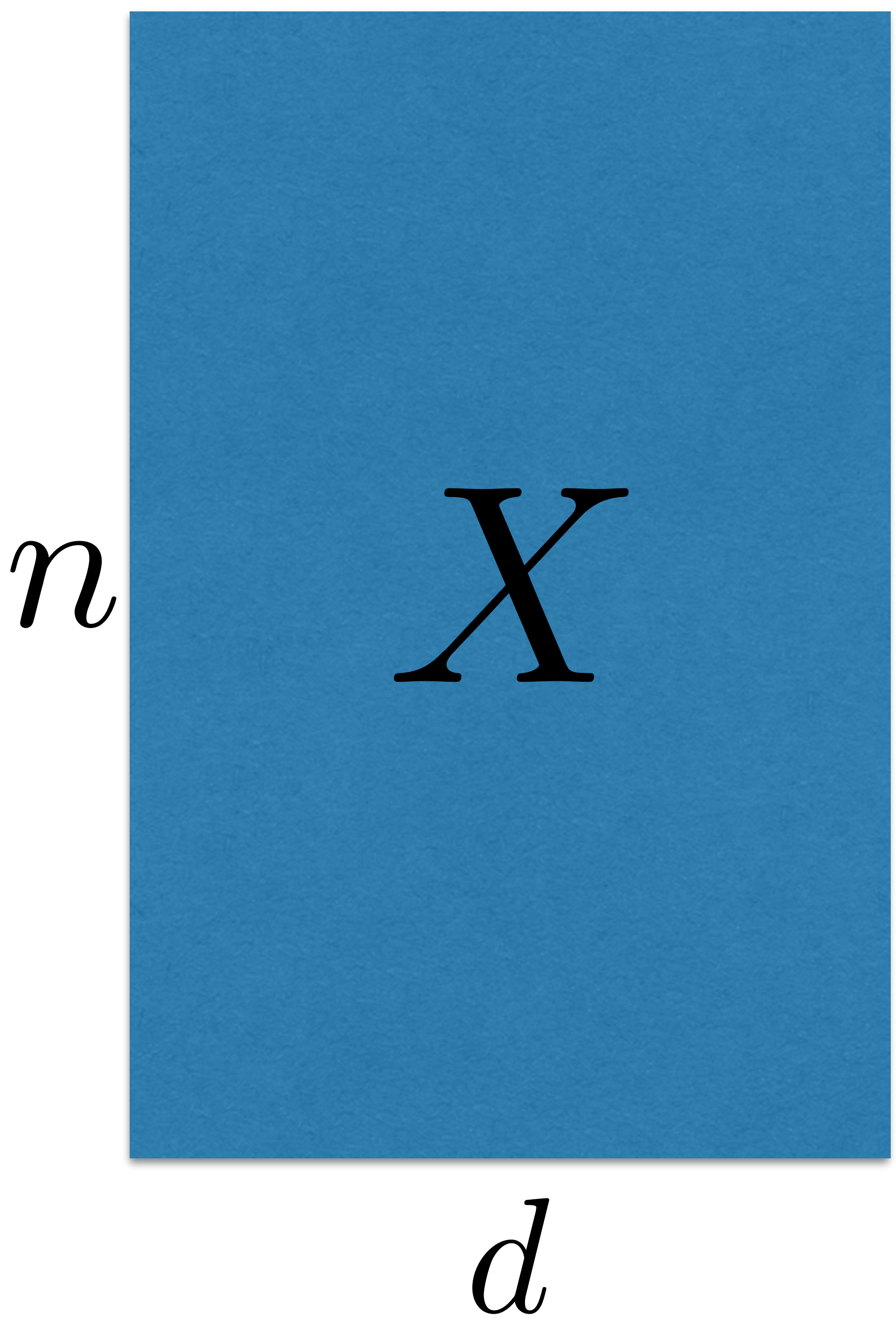
- You are provided with n data points each in $\mathbb{R}^d$
- Goal: Compress data into n, points in $\mathbb{R}^K$ where $K << d$
  - Retain as much information about the original data set
  - Retain desired properties of the original data set

Dimensionality Reduction

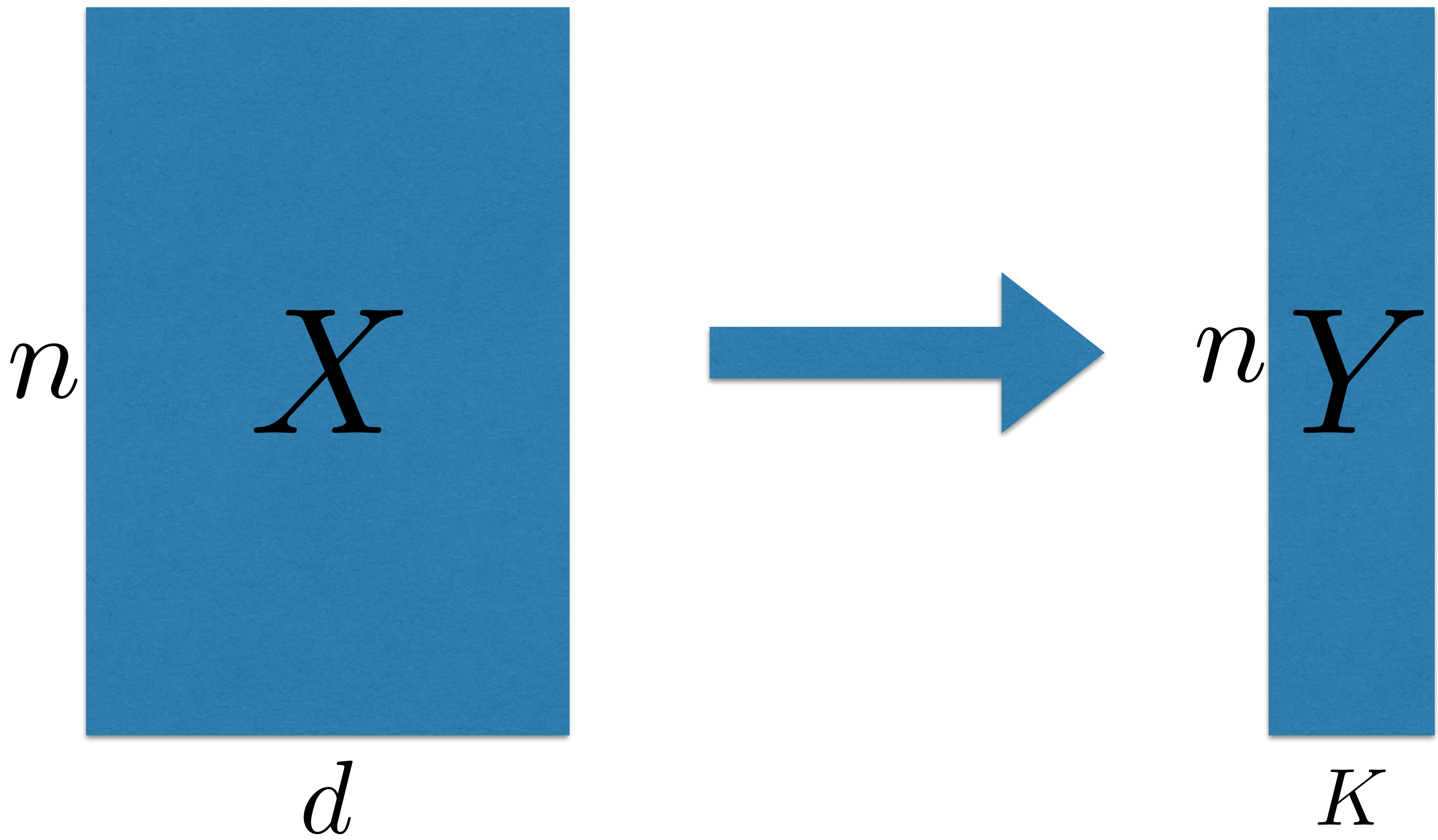
Given feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$, compress the data points into low dimensional representation $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^K$ where $K << d$

Dimensionality Reduction

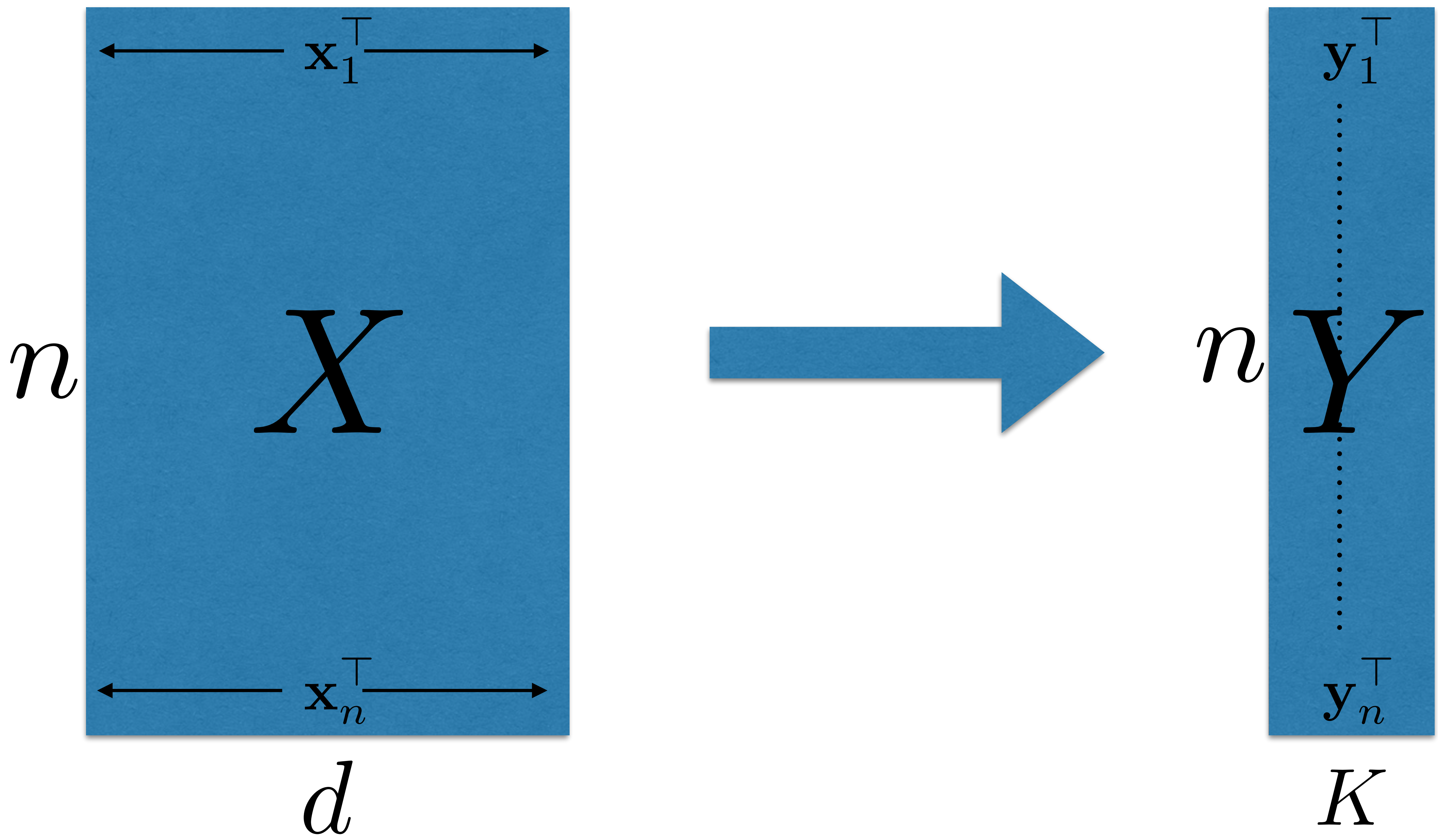
Dimensionality Reduction



Dimensionality Reduction



Dimensionality Reduction



Dimensionality Reduction

Desired properties:

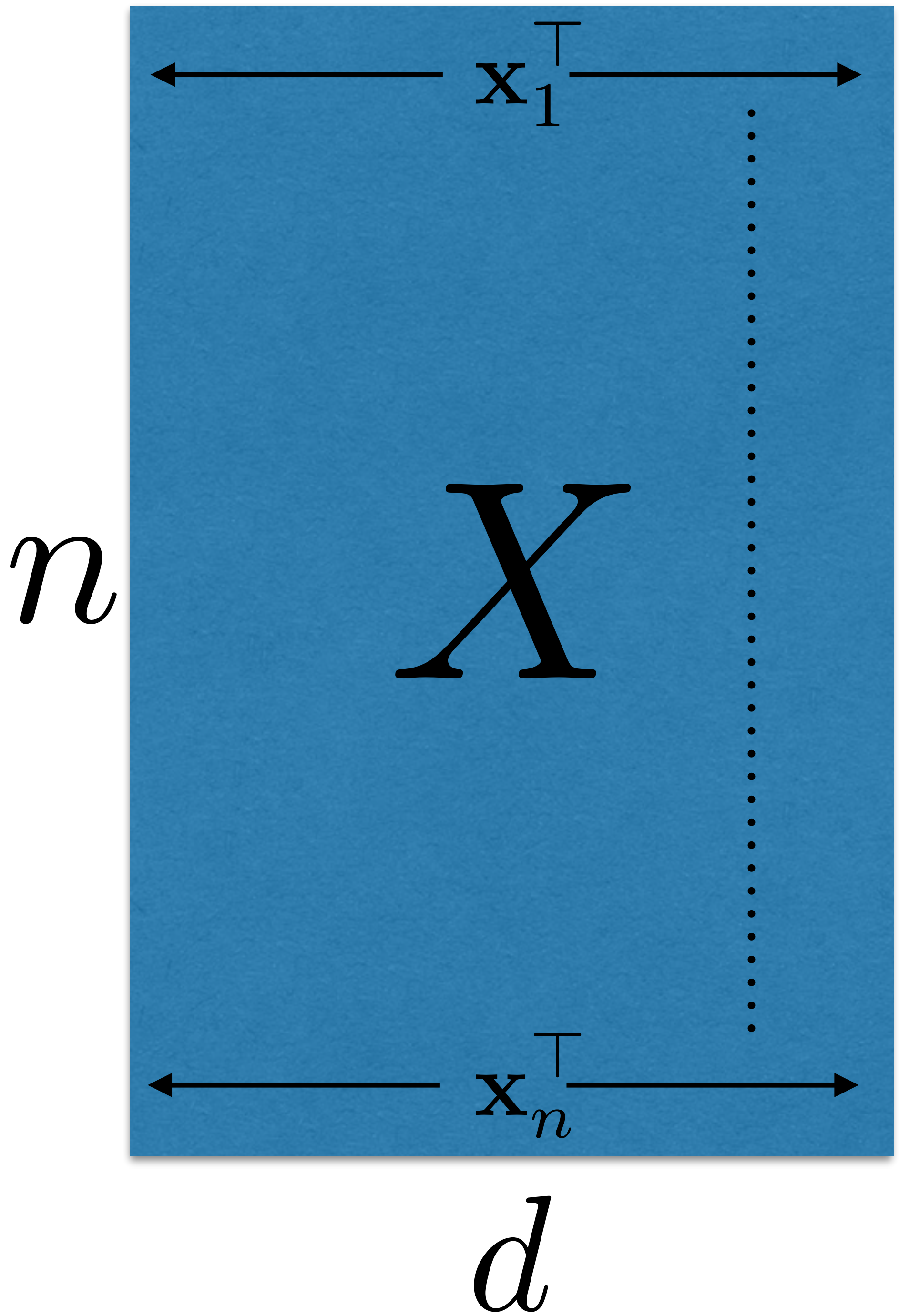
1. Original data can be (approximately) reconstructed
2. Preserve distances between data points
3. "Relevant" information is preserved
4. Noise is reduced

Dim Reduction: Linear Transformation

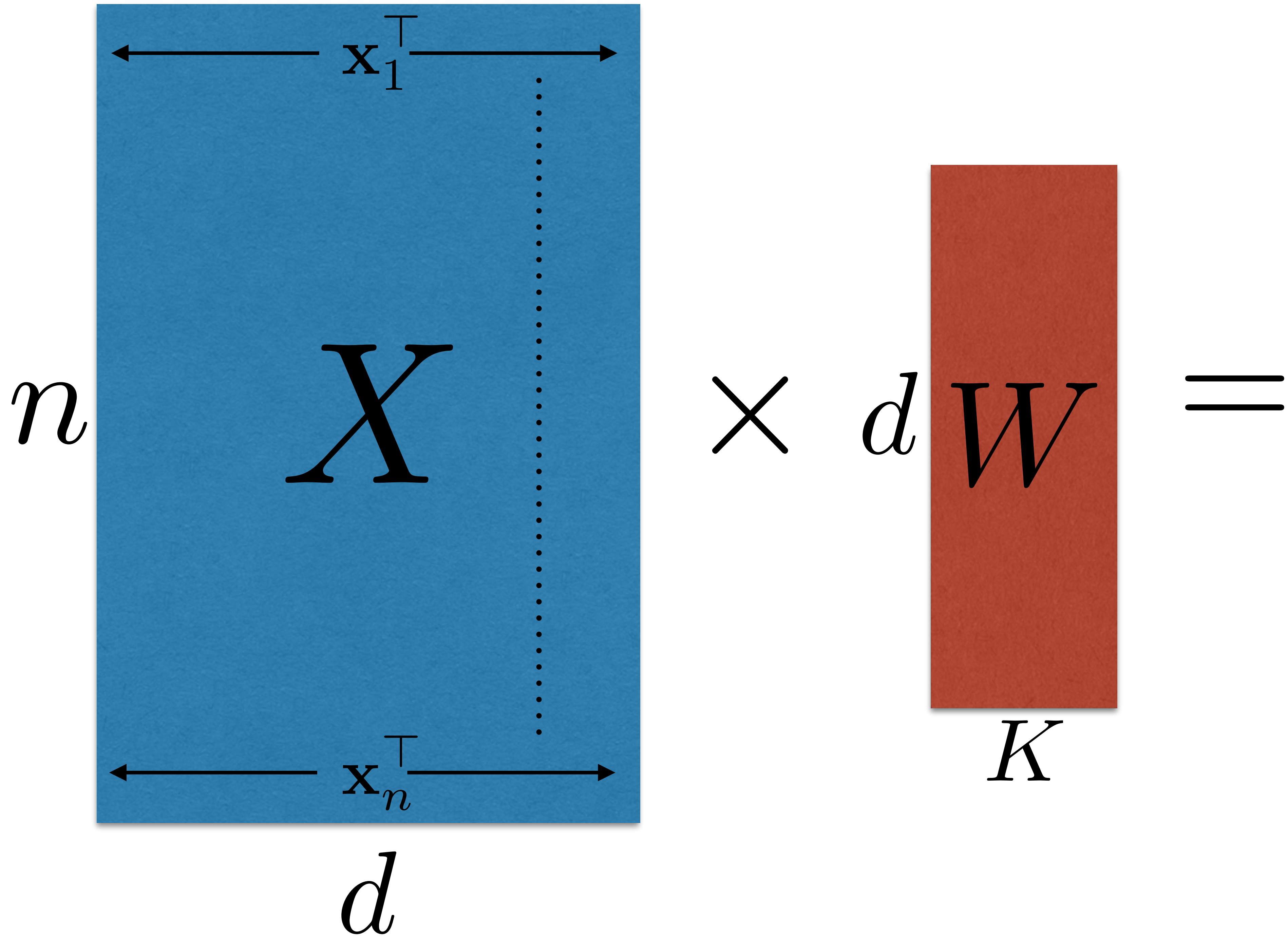
- Pick a low dimensional subspace
- Project linearly to this subspace
- Subspace retains as much information

Dim Reduction: Linear Transformation

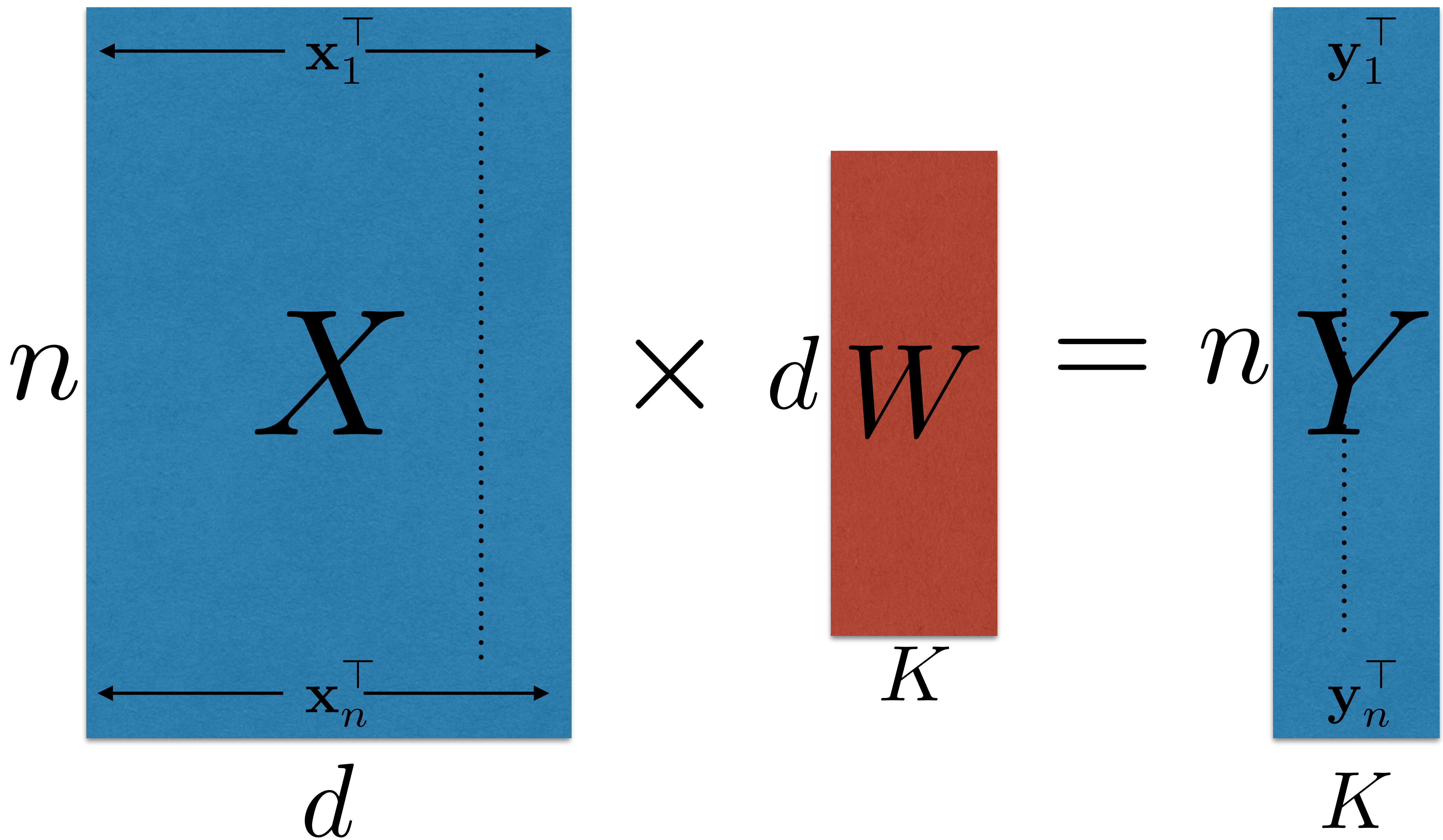
Dim Reduction: Linear Transformation



Dim Reduction: Linear Transformation



Dim Reduction: Linear Transformation



DIM REDUCTION: LINEAR TRANSFORMATION

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1.1 & 2 & 3 & 4 \\ 3 & 2 & 3 & 4 \\ -1 & 2 & 3 & 4 \\ -0.2 & 2 & 3 & 4 \\ -2 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ -0.1 & 2 & 3 & 4 \\ 0.5 & 2 & 3 & 4 \end{bmatrix}$$

Dim Reduction: Linear Transformation

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1.1 & 2 & 3 & 4 \\ 3 & 2 & 3 & 4 \\ -1 & 2 & 3 & 4 \\ -0.2 & 2 & 3 & 4 \\ -2 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ -0.1 & 2 & 3 & 4 \\ 0.5 & 2 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.1 \\ 3 \\ -1 \\ -0.2 \\ -2 \\ 1.4 \\ 1.4 \\ -0.1 \\ 0.5 \end{bmatrix} \times [1, 0, 0, 0] + \begin{bmatrix} 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \end{bmatrix}$$

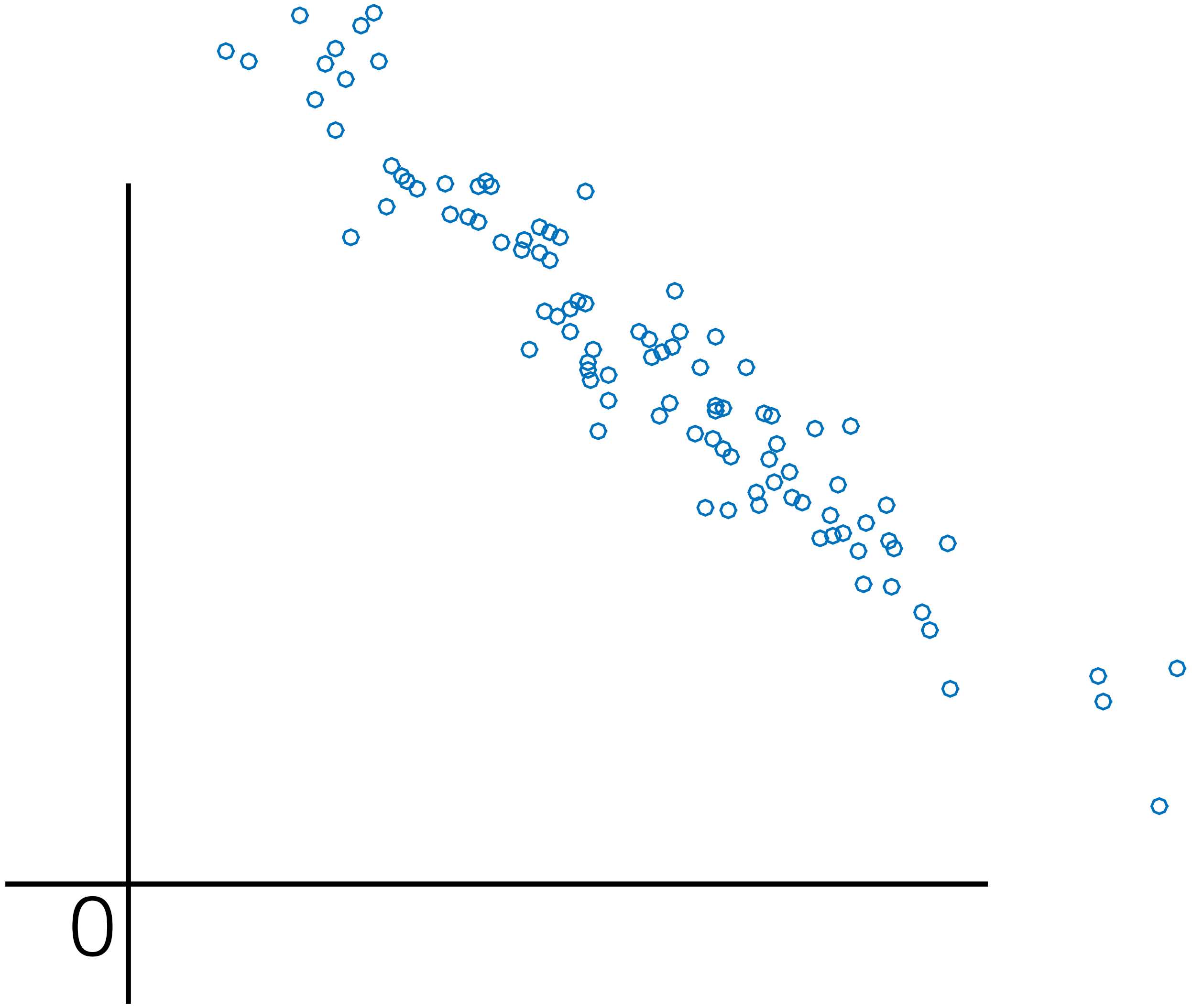
Dim Reduction: Linear Transformation

$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1.1 & 2 & 3 & 4 \\ 3 & 2 & 3 & 4 \\ -1 & 2 & 3 & 4 \\ -0.2 & 2 & 3 & 4 \\ -2 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ -0.1 & 2 & 3 & 4 \\ 0.5 & 2 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.1 \\ 3 \\ -1 \\ -0.2 \\ -2 \\ 1.4 \\ 1.4 \\ -0.1 \\ 0.5 \end{bmatrix} \times \underset{\mathrm{W}}{[1, 0, 0, 0]} + \begin{bmatrix} 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \end{bmatrix}$$

Dim Reduction: Linear Transformation

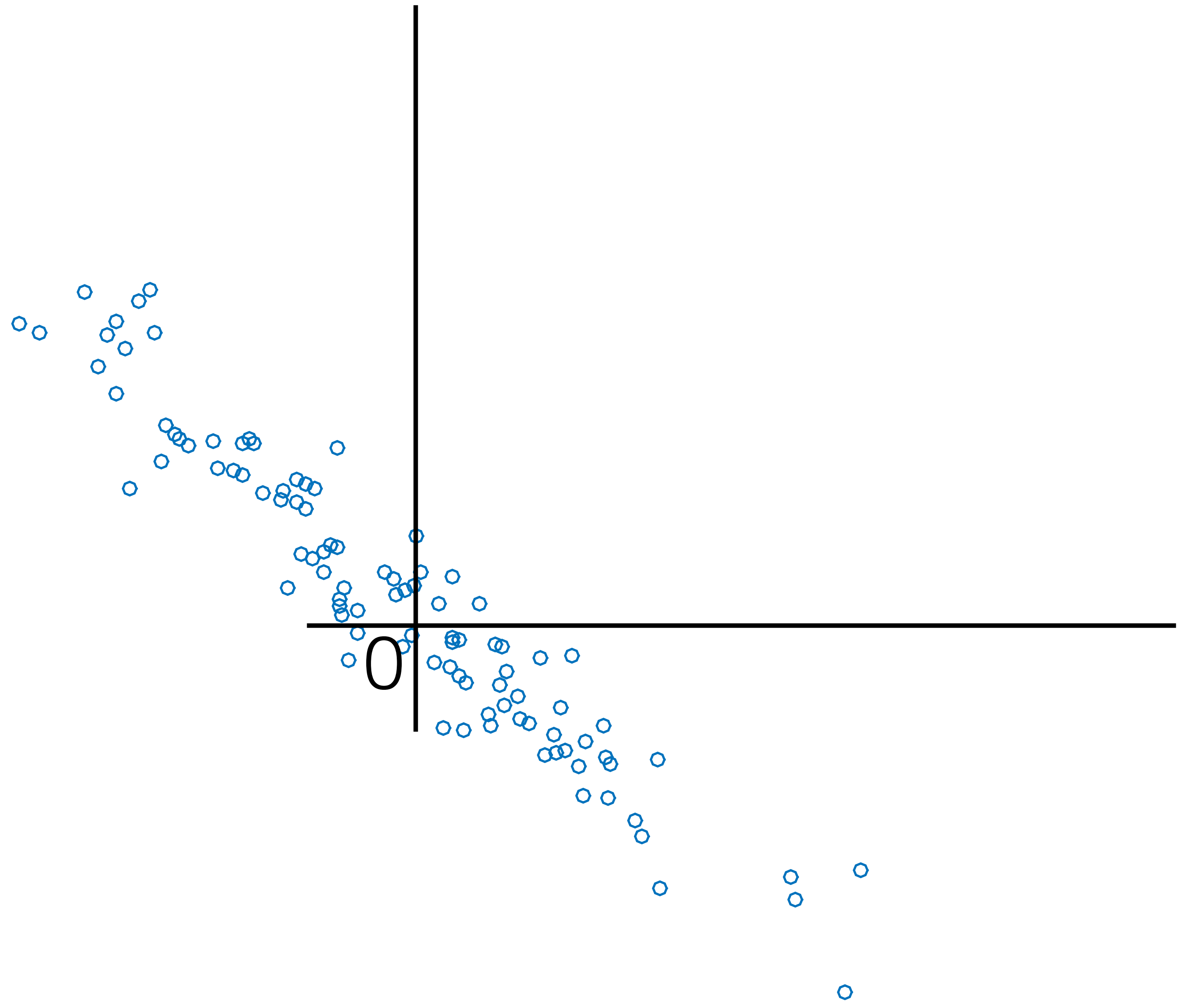
$$X = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1.1 & 2 & 3 & 4 \\ 3 & 2 & 3 & 4 \\ -1 & 2 & 3 & 4 \\ -0.2 & 2 & 3 & 4 \\ -2 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ 1.4 & 2 & 3 & 4 \\ -0.1 & 2 & 3 & 4 \\ 0.5 & 2 & 3 & 4 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 \\ 1.1 \\ 3 \\ -1 \\ -0.2 \\ -2 \\ 1.4 \\ 1.4 \\ -0.1 \\ 0.5 \end{bmatrix}}_{Y} \times \underbrace{[1, 0, 0, 0]}_{W} + \begin{bmatrix} 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \\ 0 & 2 & 3 & 4 \end{bmatrix}$$

Centering Data



Compressing these data points…

Centering Data

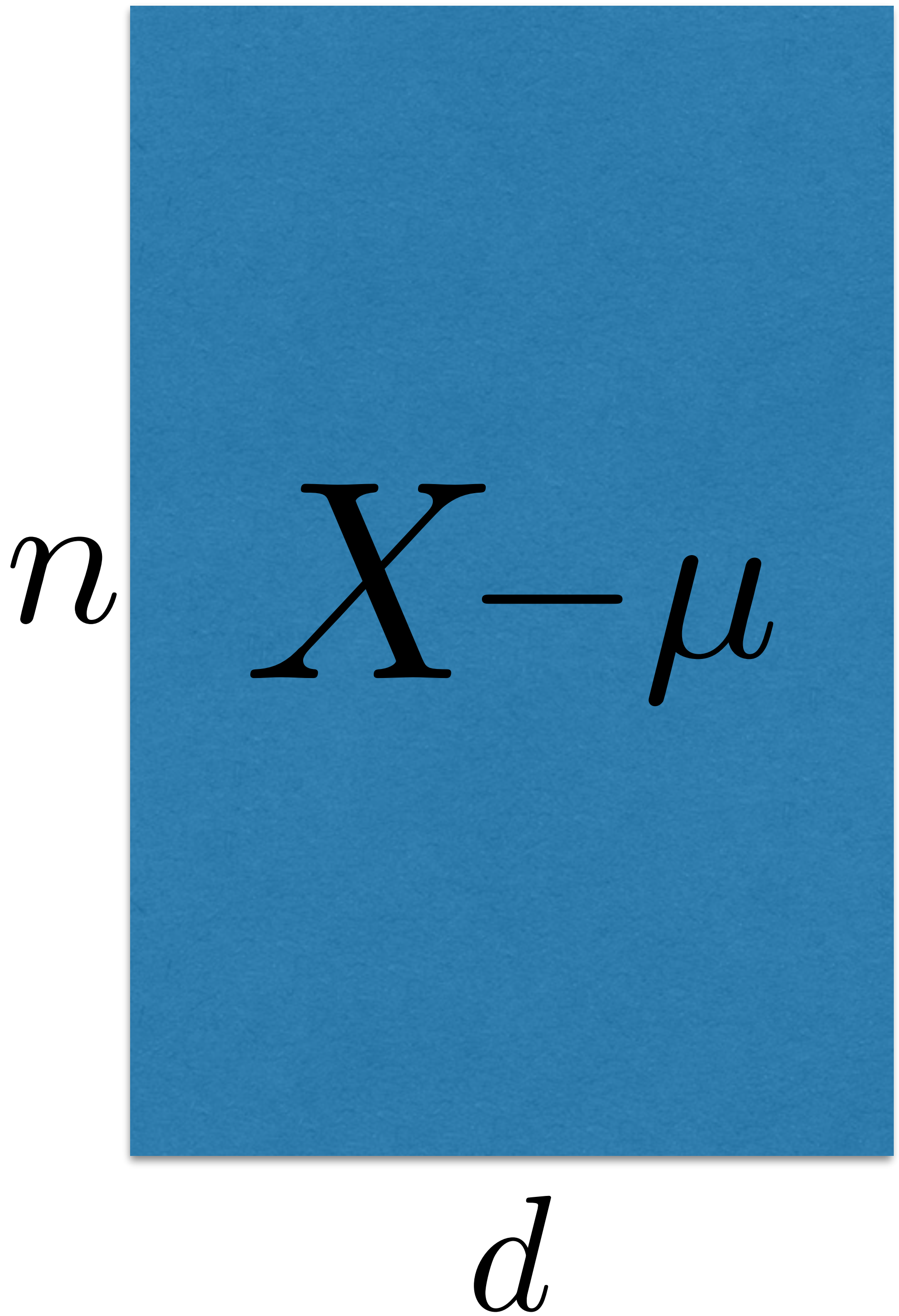


… is same as compressing these.

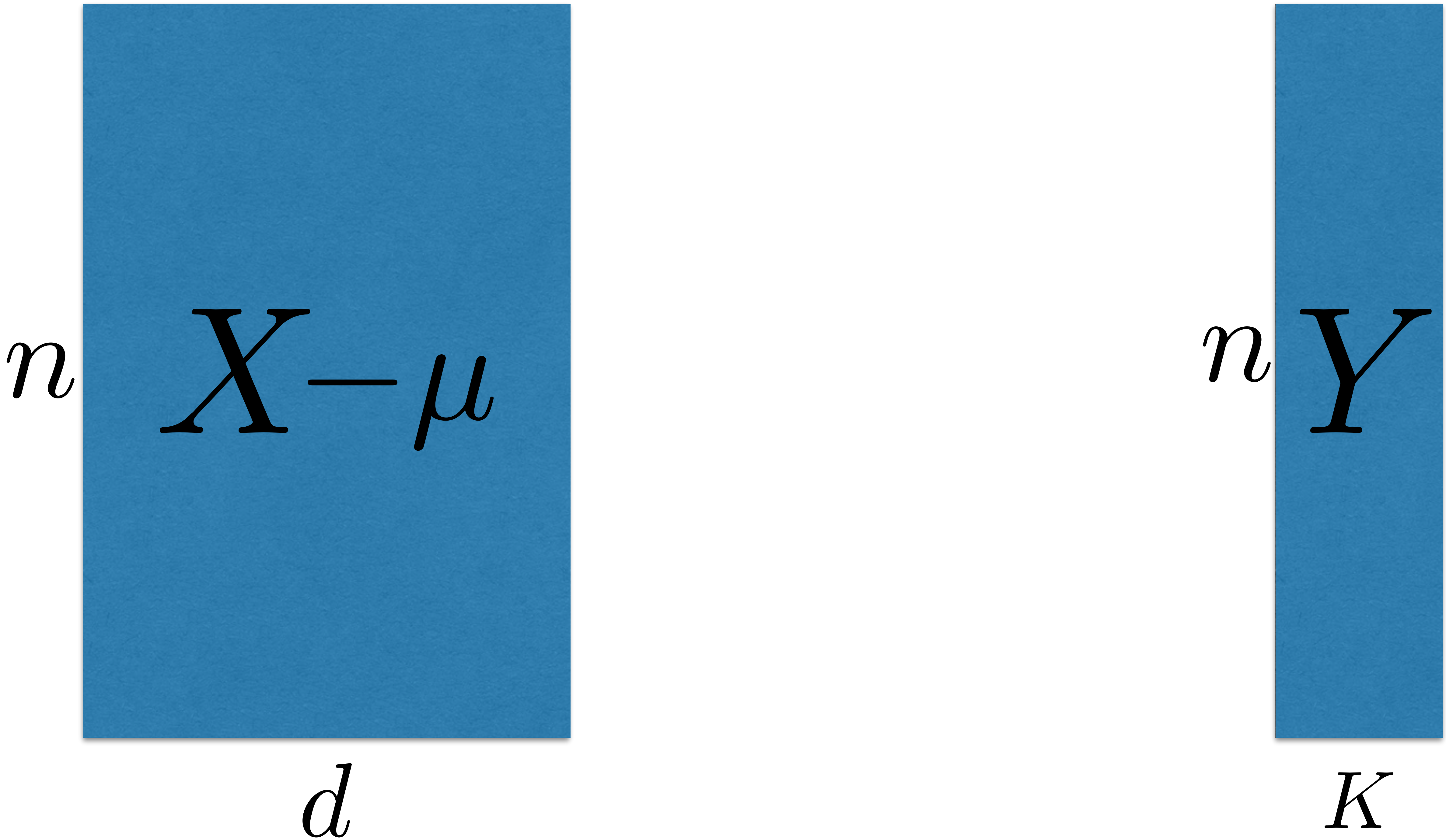
Centering Data

$$-\mu$$

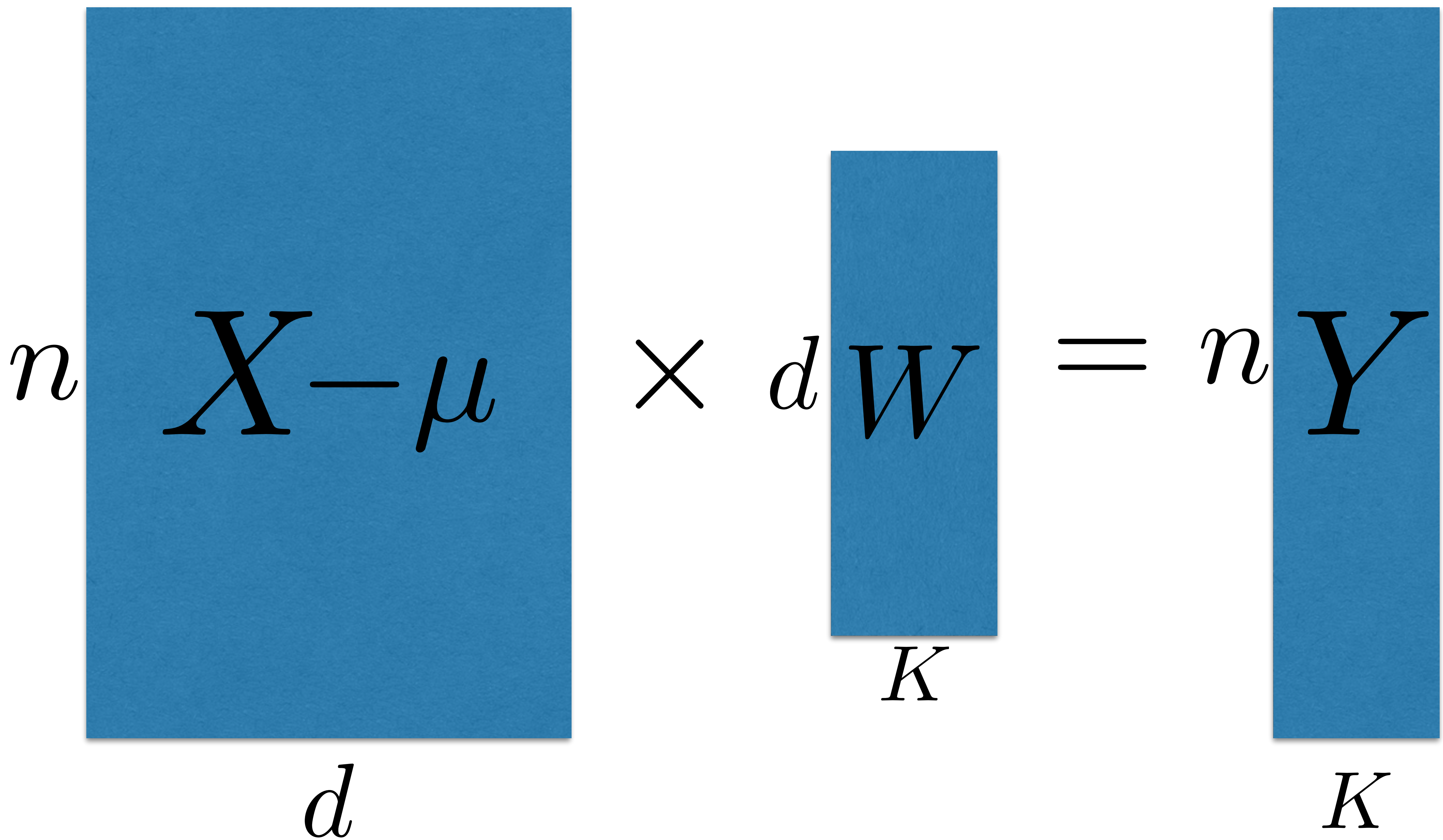
CENTERING DATA



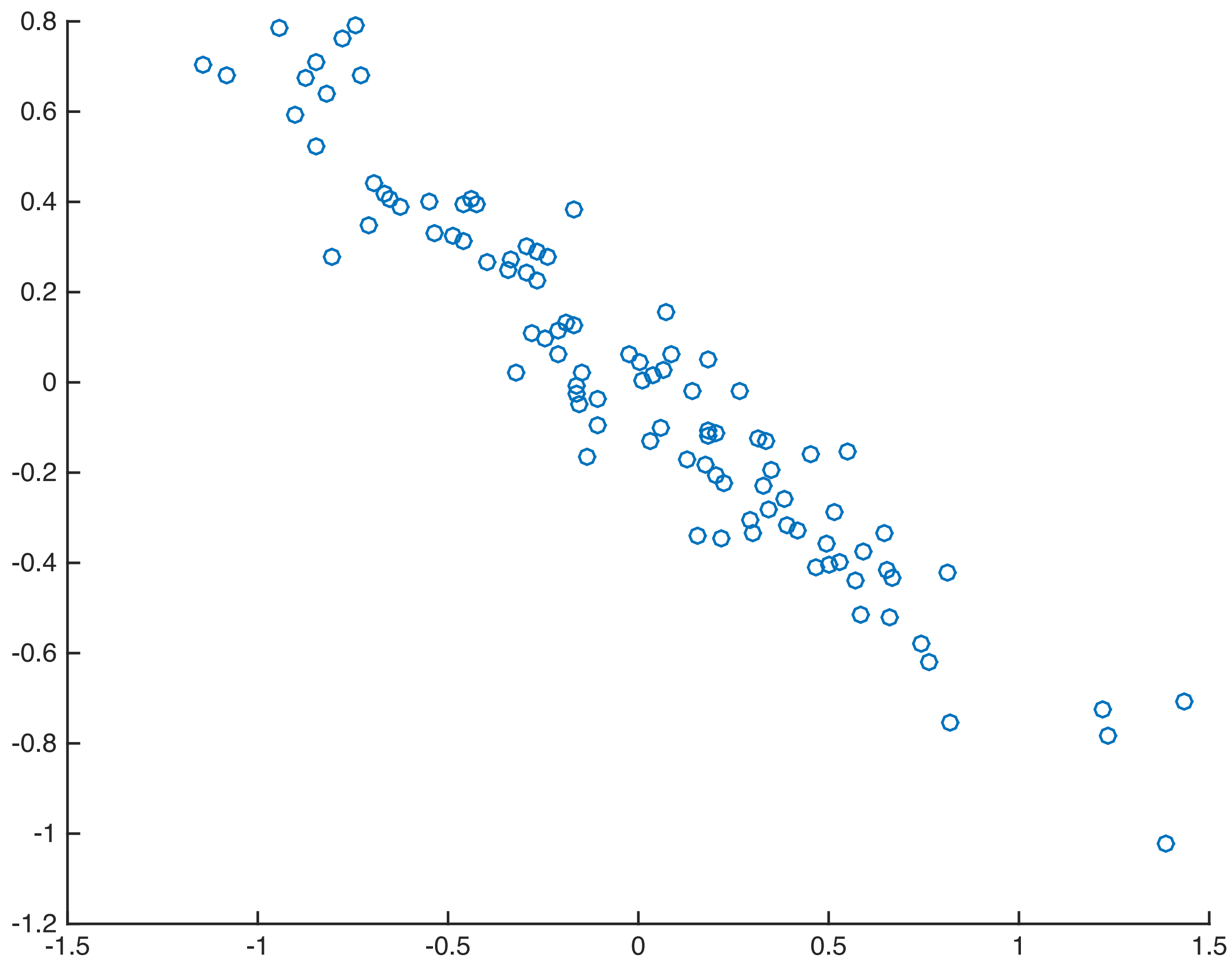
Centering Data



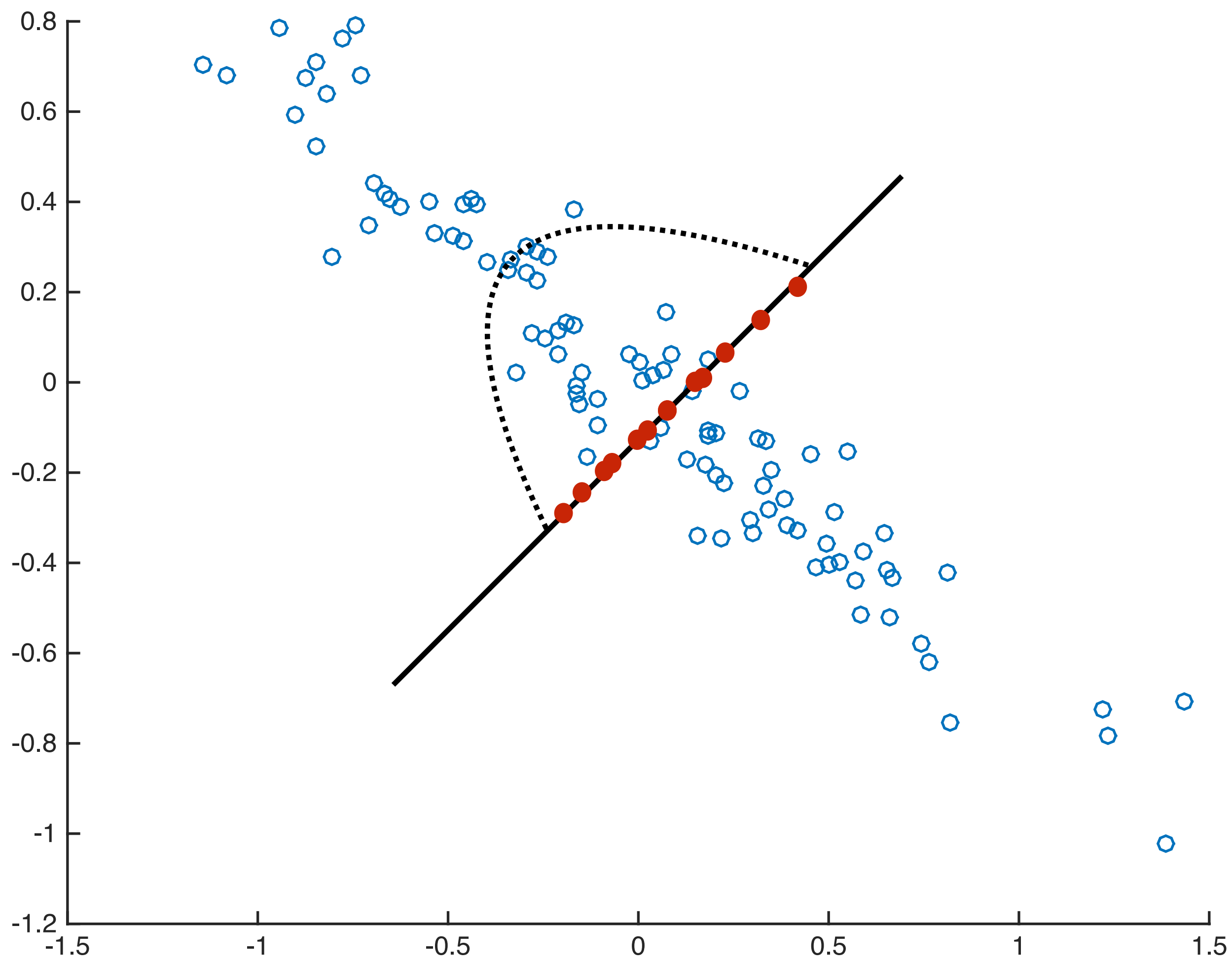
CENTERING DATA



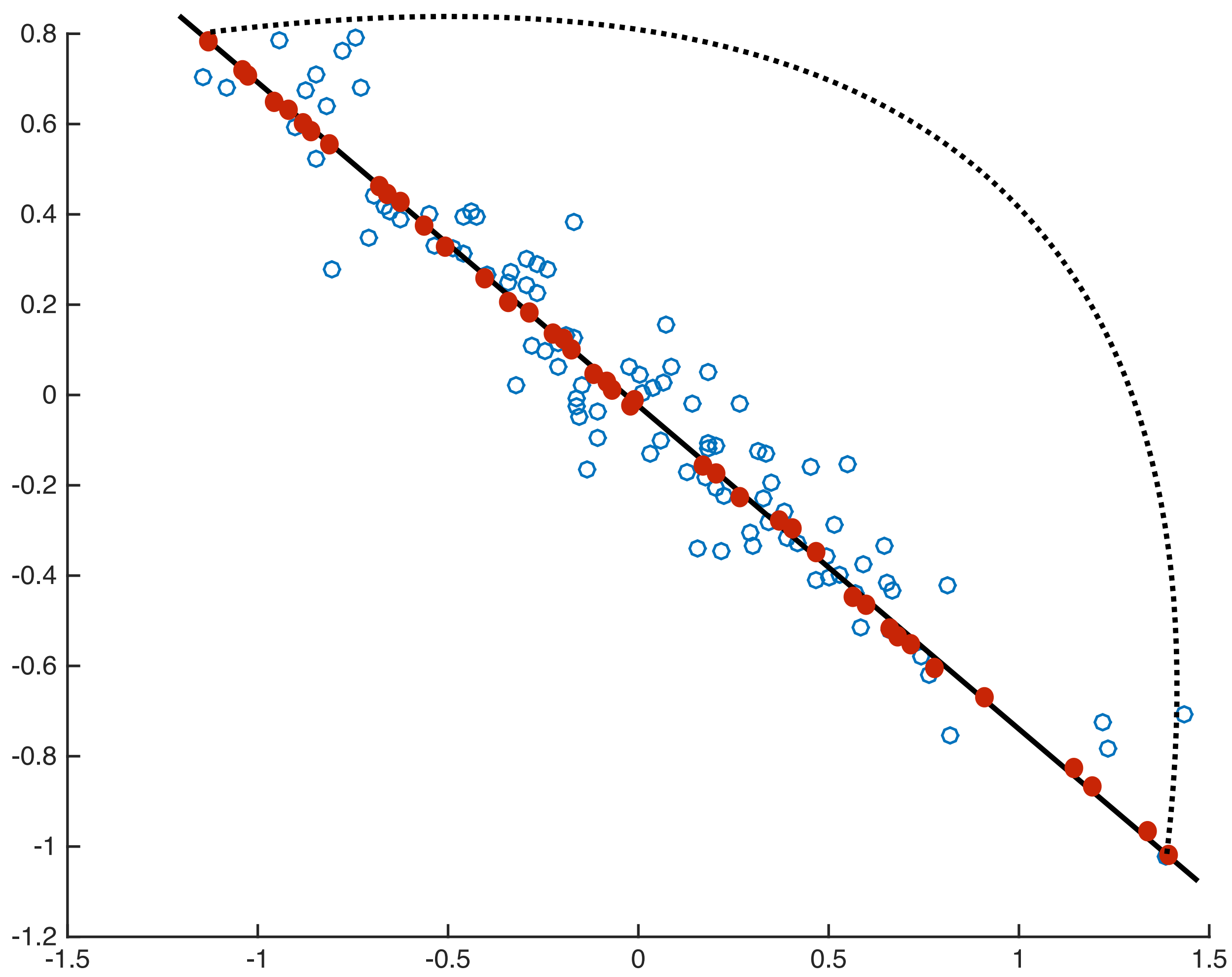
PCA: Variance Maximization



PCA: Variance Maximization



PCA: Variance Maximization



PCA: Variance Maximization

- Pick directions along which data varies the most

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:

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

PCA: Variance Maximization

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

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

PCA: Variance Maximization

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

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

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

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

$$= \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top\Sigma\mathbf{w}$$

Σ is the covariance matrix

PCA: Variance Maximization

Covariance matrix:

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

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

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

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
- Recall $\mathrm{cov}(A,B) = \mathbb{E}[(A - \mathbb{E}[A])(B - \mathbb{E}[B])]$
- Alternatively,

$$\Sigma[i,j] = \frac{1}{n}\begin{bmatrix} \mathbf{x}_1[j] - \mu[j] \\ \mathbf{x}_2[j] - \mu[j] \\ \cdots \\ \cdots \\ \cdots \\ \mathbf{x}_n[j] - \mu[j] \end{bmatrix}^\top \begin{bmatrix} \mathbf{x}_1[j] - \mu[j] \\ \mathbf{x}_2[j] - \mu[j] \\ \cdots \\ \cdots \\ \cdots \\ \mathbf{x}_n[j] - \mu[j] \end{bmatrix}$$

Inner products measure similarity.

PCA: Variance Maximization

- First principal component:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top \Sigma \mathbf{w} \tag{1}$$

PCA: Variance Maximization

- First principal component:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top \Sigma \mathbf{w} \tag{1}$$

To solve the above maximization problem, we use Lagrange multipliers. Specifically there exists λ such that solution $\mathbf{w}_1$ is:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} - \lambda \|\mathbf{w}\|_2^2$$

PCA: Variance Maximization

- First principal component:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}:\|\mathbf{w}\|_2=1} \mathbf{w}^\top \Sigma \mathbf{w} \tag{1}$$

To solve the above maximization problem, we use Lagrange multipliers. Specifically there exists λ such that solution $\mathbf{w}_1$ is:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}} \mathbf{w}^\top \Sigma \mathbf{w} - \lambda \|\mathbf{w}\|_2^2$$

Taking derivate and equality to 0 we find that $\Sigma \mathbf{w} = \lambda \mathbf{w}$ (ie. eigenvector). Plugging this back into Eq. 1,

$$\frac{1}{n} \sum_{t=1}^{n} \mathbf{w}^\top \Sigma \mathbf{w} = \mathbf{w}^\top (\lambda \mathbf{w}) = \lambda$$

Hence to maximize variance we pick direction with largest eigenvalue

Principal Component Analysis

- Eigenvectors of the covariance matrix are the principal components

Principal Component Analysis

- Eigenvectors of the covariance matrix are the principal components

- Top K principal components are the eigenvectors with K largest eigenvalues

PRINCIPAL COMPONENT ANALYSIS

- Eigenvectors of the covariance matrix are the principal components

- Top K principal components are the eigenvectors with K largest eigenvalues

- Projection = Data × Top K eigenvectors

Principal Component Analysis

- Eigenvectors of the covariance matrix are the principal components
- Top K principal components are the eigenvectors with K largest eigenvalues
- Projection = Data × Top K eigenvectors
- Reconstruction = Projection × Transpose of top K eigenvectors

Principal Component Analysis

- Eigenvectors of the covariance matrix are the principal components
- Top K principal components are the eigenvectors with K largest eigenvalues
- Projection = Data × Top K eigenvectors
- Reconstruction = Projection × Transpose of top K eigenvectors
- Independently discovered by Pearson in 1901 and Hotelling in 1933.

Principal Component Analysis: Demo