Principal Component Analysis

Announcement:

1. P2 will be out later this week

Recap on K-means

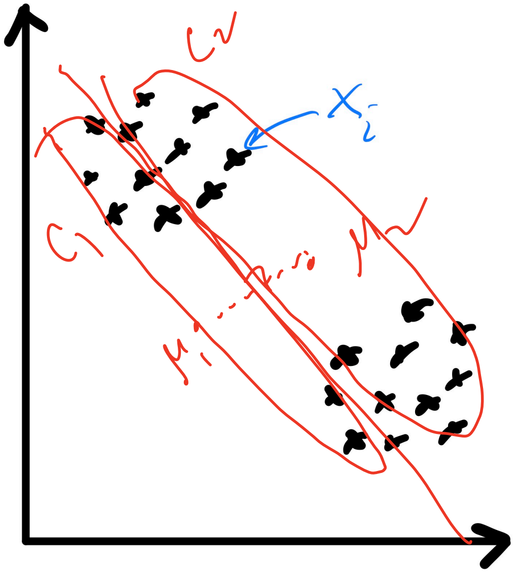
T/F: K-means also has curse of dimensionality

T/F: K-means solution depends on its initialization

Recap on K-means

T/F: K-means also has curse of dimensionality

T/F: K-means solution depends on its initialization



Outline for today:

1. Intro of PCA

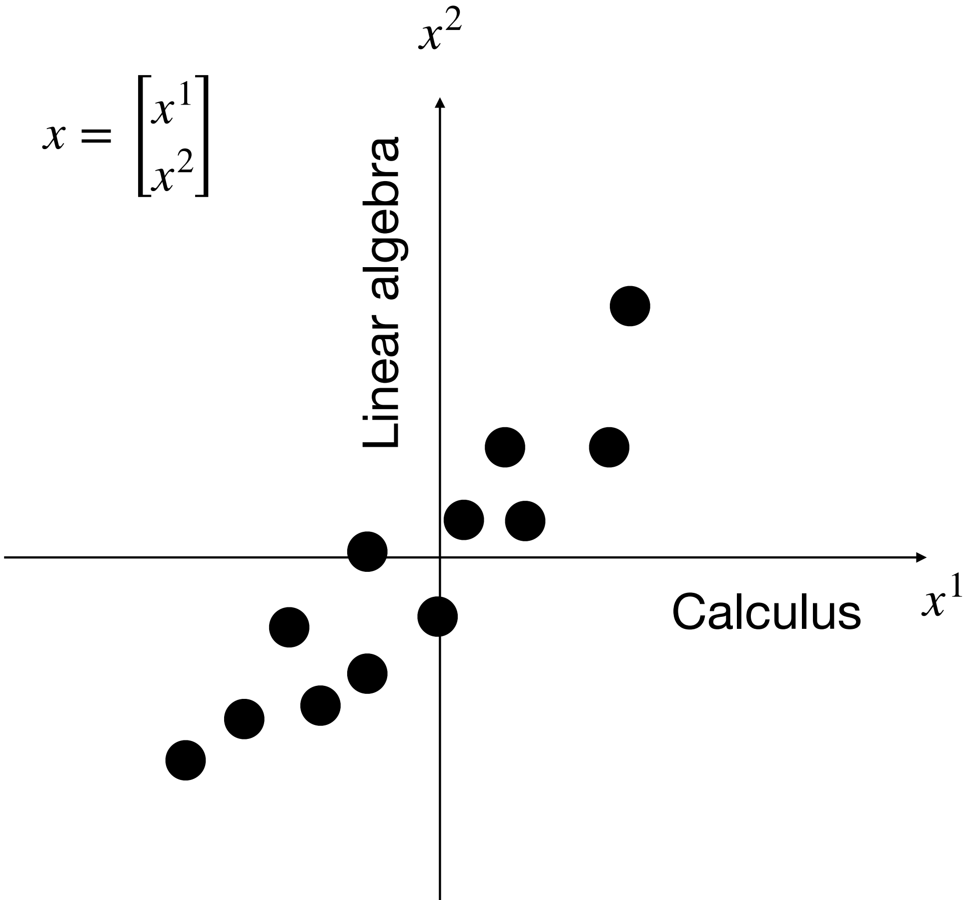
2. PCA via eigendecomposition

3. Example of PCA: eigenfaces

Data compression

Goal: reduce high dimensional data to low dimensional

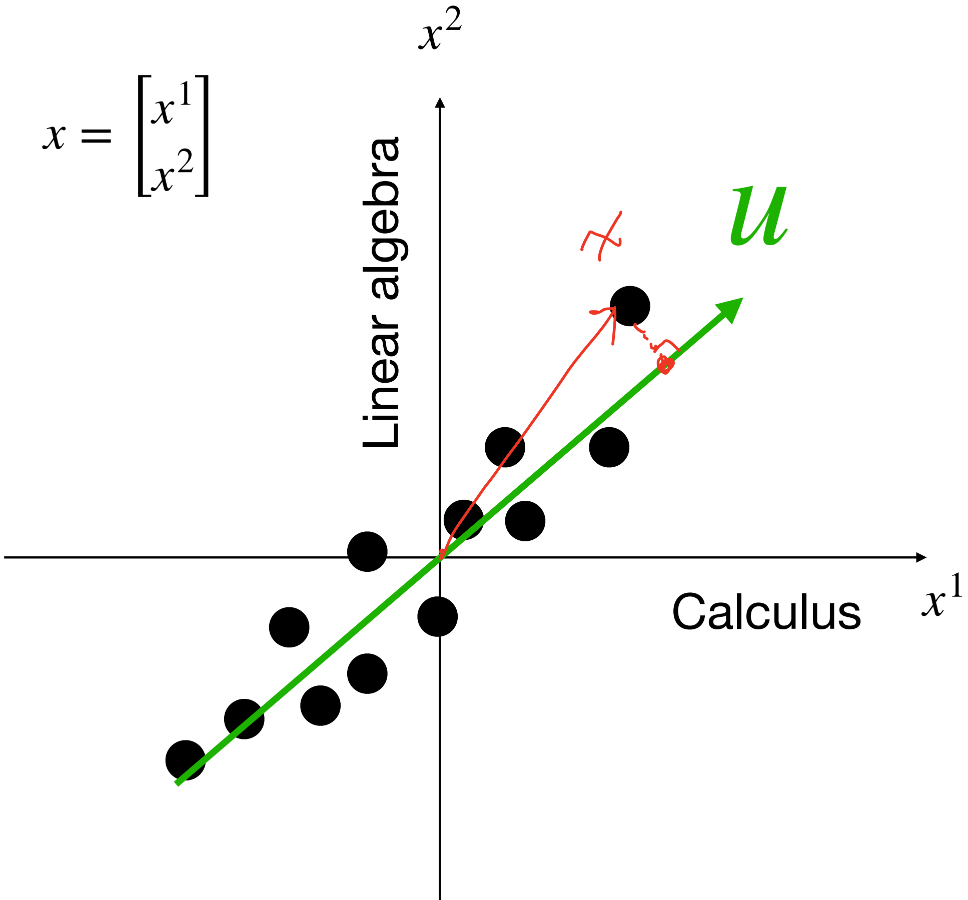
$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$$



Data compression

Goal: reduce high dimensional data to low dimensional

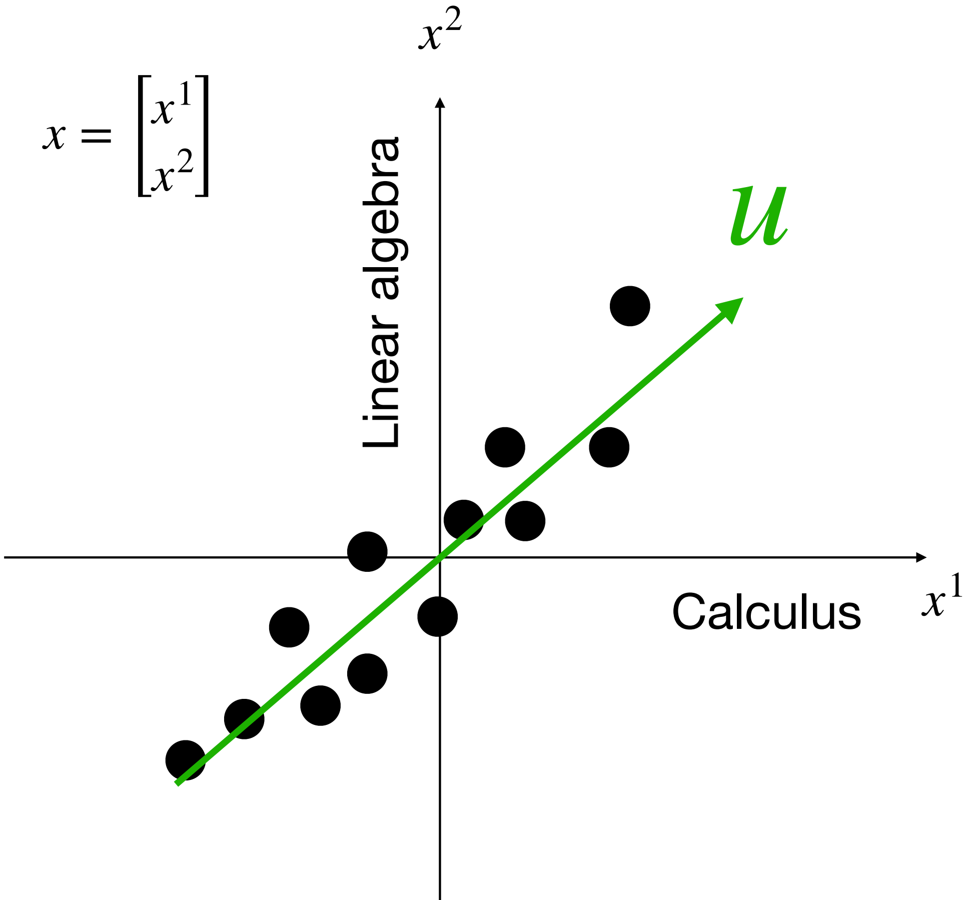
$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$



Data compression

Goal: reduce high dimensional data to low dimensional

$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$$

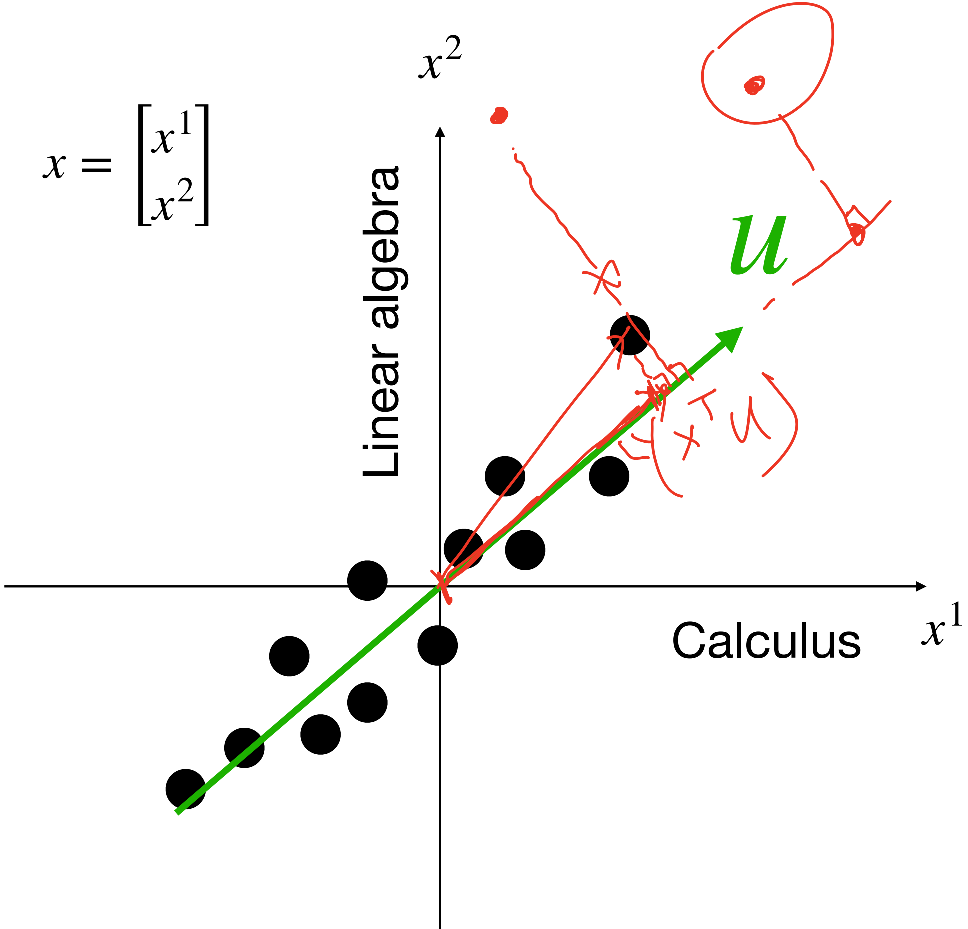


Normalize u s.t. $\|u\|_2 = 1$

Data compression

Goal: reduce high dimensional data to low dimensional

$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$$



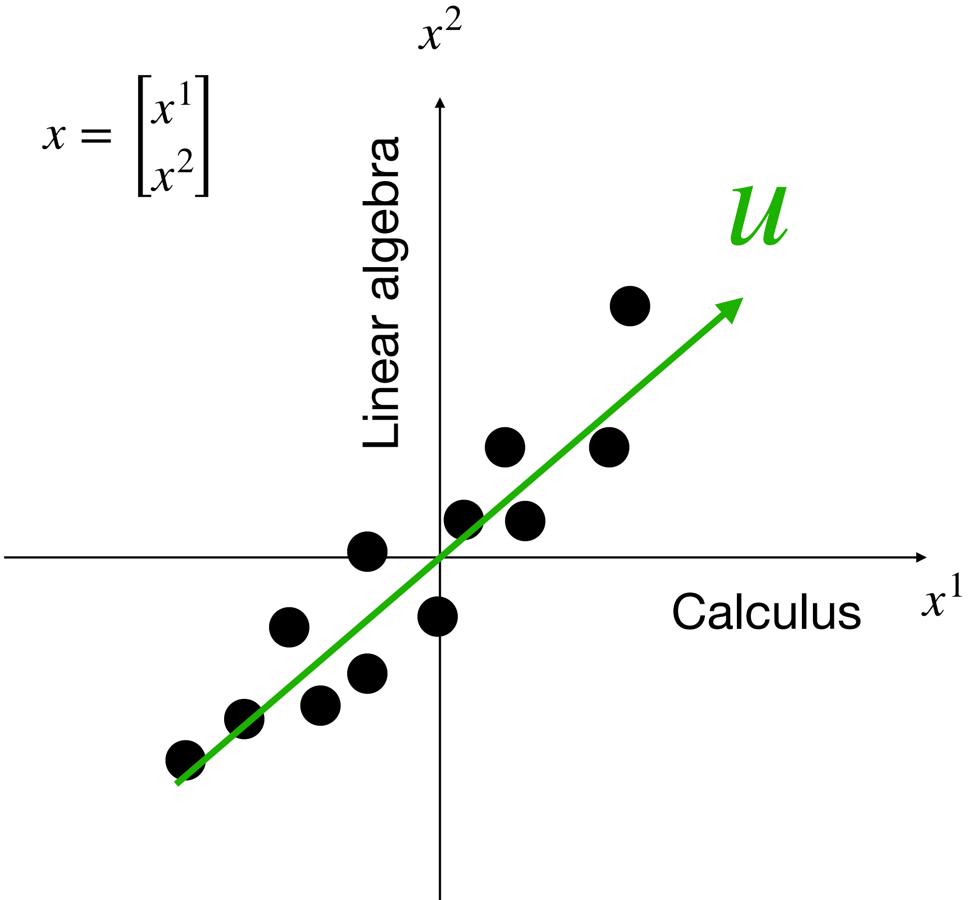
Normalize u s.t. $\|u\|_2 = 1$

Math skill: $z := x^\top u$

Data compression

Goal: reduce high dimensional data to low dimensional

$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$



Normalize u s.t. $\|u\|_2 = 1$

Math skill: $z := x^\top u$

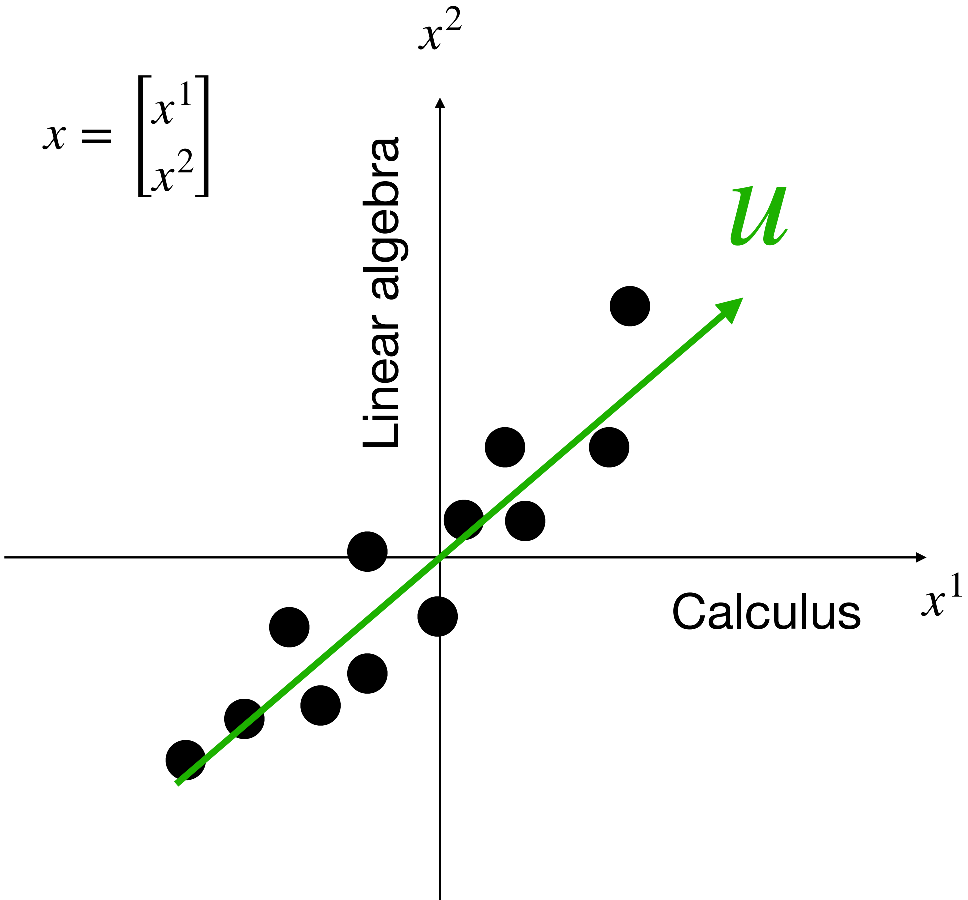
Dim-reduction:

Given $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^2$

Data compression

Goal: reduce high dimensional data to low dimensional

$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$



Normalize u s.t. $\|u\|_2 = 1$

Math skill: $z := x^\top u$

Dim-reduction:

Given $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^2$

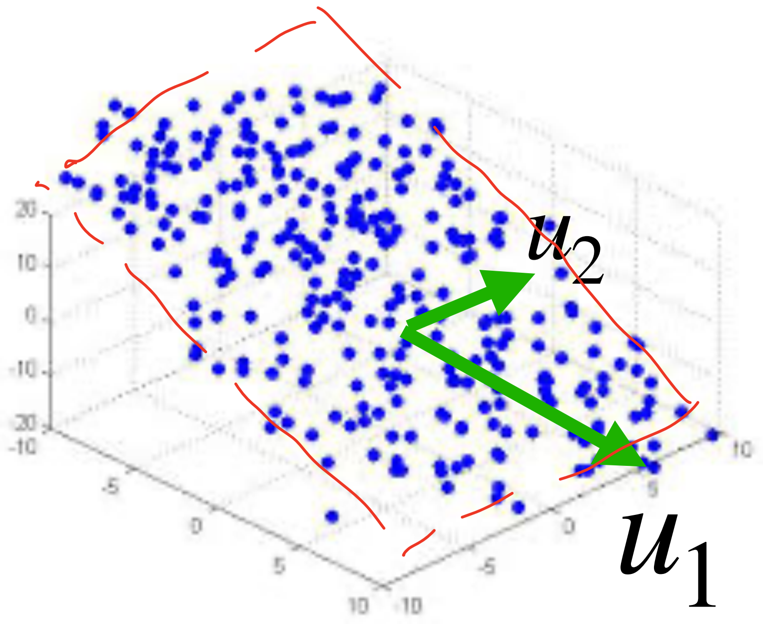
We get a 1-d dataset

$\mathscr{Z} = \{z_1, \ldots, z_n\}$, where $z_i = u^\top x_i$

Data compression

Goal: reduce high dimensional data to low dimensional

$$x = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix}$$

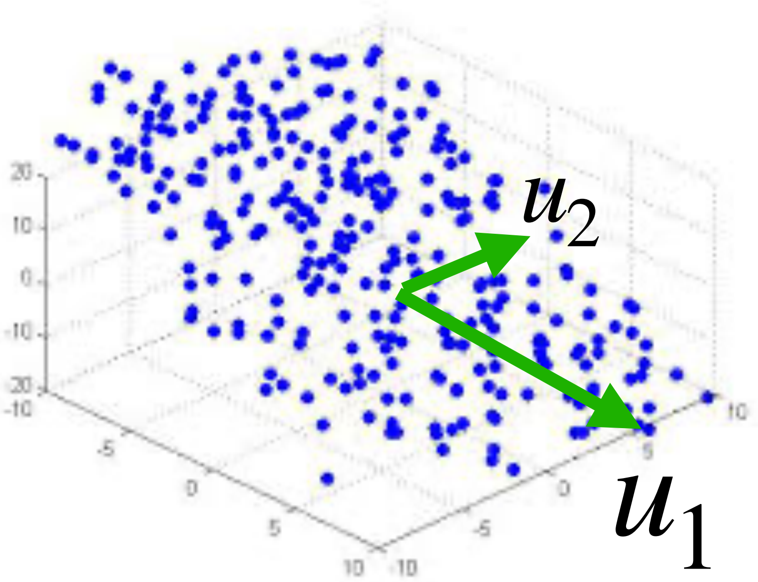


Reduce data from 3-d to 2-d:

Data compression

Goal: reduce high dimensional data to low dimensional

$$x = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix}$$



Reduce data from 3-d to 2-d:

Span(u_1, u_2)

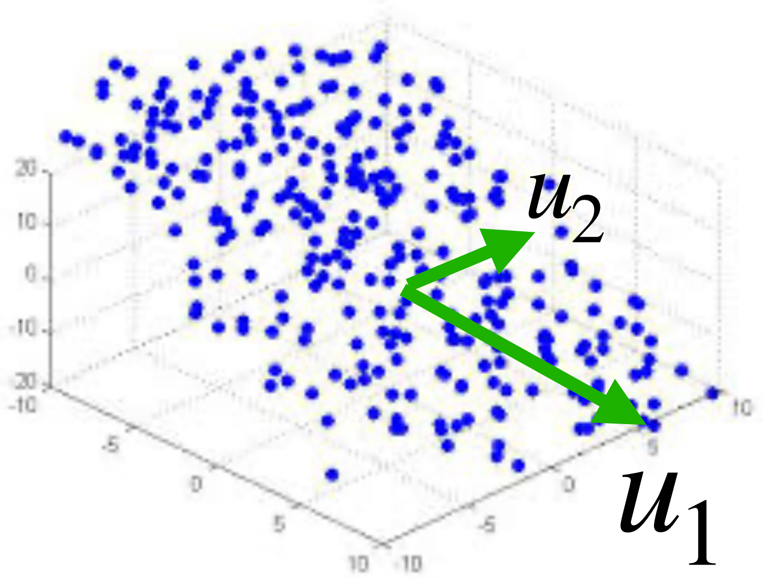
$$u_1 \in \mathbb{R}^3, u_2 \in \mathbb{R}^3$$

$$\mathcal{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^3$$

Data compression

Goal: reduce high dimensional data to low dimensional

$$x = \begin{bmatrix} x^1 \\ x^2 \\ x^3 \end{bmatrix}$$



Reduce data from 3-d to 2-d:

$$u_1 \in \mathbb{R}^3, u_2 \in \mathbb{R}^3$$

$$\mathcal{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^3$$

$$\mathcal{Z} = \{z_1, \ldots, z_n\}, z_i \in \mathbb{R}^3, z_i = [u_1^\top x_i, u_2^\top x_i]^\top$$

Outline for today:

1. Intro of PCA

2. PCA via eigendecomposition

3. Example of PCA: eigenfaces

Compute the Principal Component

Setup

Input: dataset $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{d \times n}$$

Compute the Principal Component

Setup

Input: dataset $\mathcal{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ $X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$

Output: K principle components $u_1, \ldots, u_K$ (they are orthonormal)

$u_i \in \mathbb{R}^d$

$\|u_i\|_2 = 1$

$u_i^\top u_j = 0, i \neq j$

Compute the Principal Component

Setup

Input: dataset $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ $\quad X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$

Output: K principle components $u_1, \ldots, u_K$ (they are orthonormal)

Step 1: data normalization

Compute the Principal Component

Setup

Input: dataset $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ $\quad X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$

Output: K principle components $u_1, \ldots, u_K$ (they are orthonormal)

Step 1: data normalization

For each coordinate $k \in [d]$, compute data mean $\mu[k]$ and std $\sigma[k]$

$$\mu[k] = \sum_{i=1}^{n} x_i[k] / n$$

$$\sigma^2[k] = \sum_{i=1}^{n} (x_i[k] - \mu[k])^2 / n$$

Compute the Principal Component

Setup

Input: dataset $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$ $\quad X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^{d \times n}$

Output: K principle components $u_1, \ldots, u_K$ (they are orthonormal)

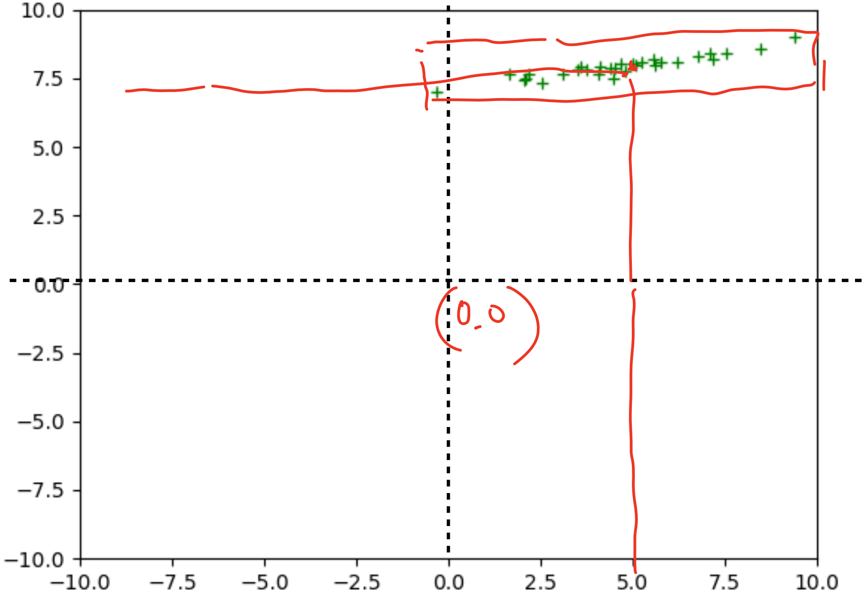
Step 1: data normalization

For each coordinate $k \in [d]$, compute data mean $\mu[k]$ and std $\sigma[k]$

$$\forall i, k: \;\; x_i[k] \leftarrow \frac{x_i[k] - \mu[k]}{\sigma[k]}$$

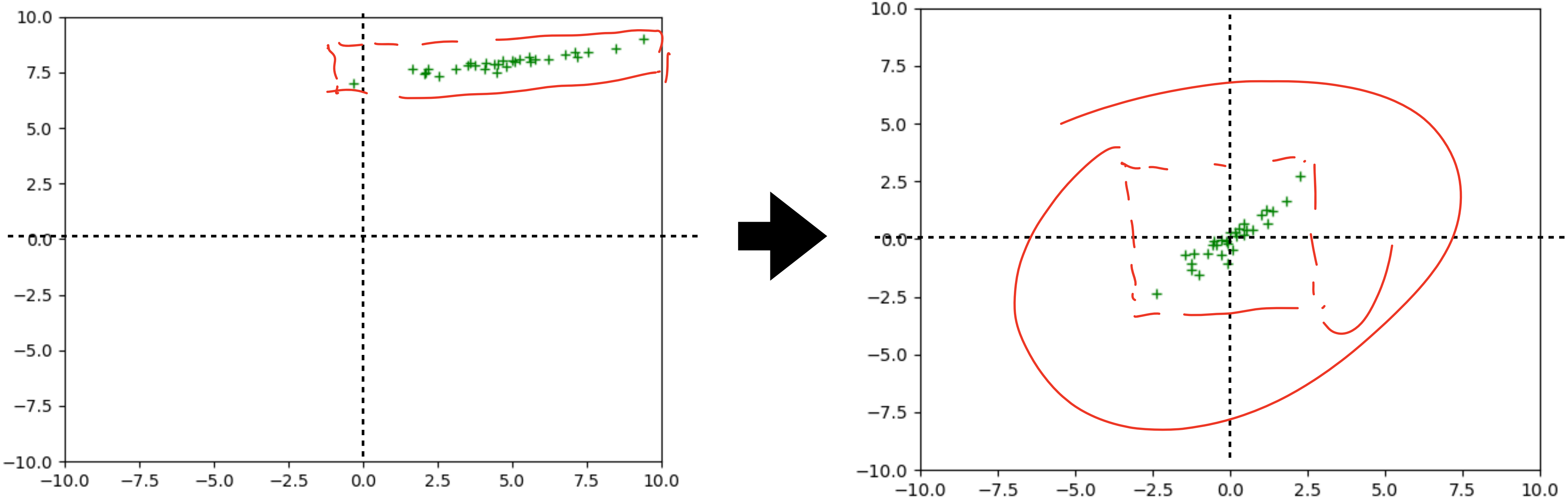
Compute the Principal Component

The outcome of data normalization

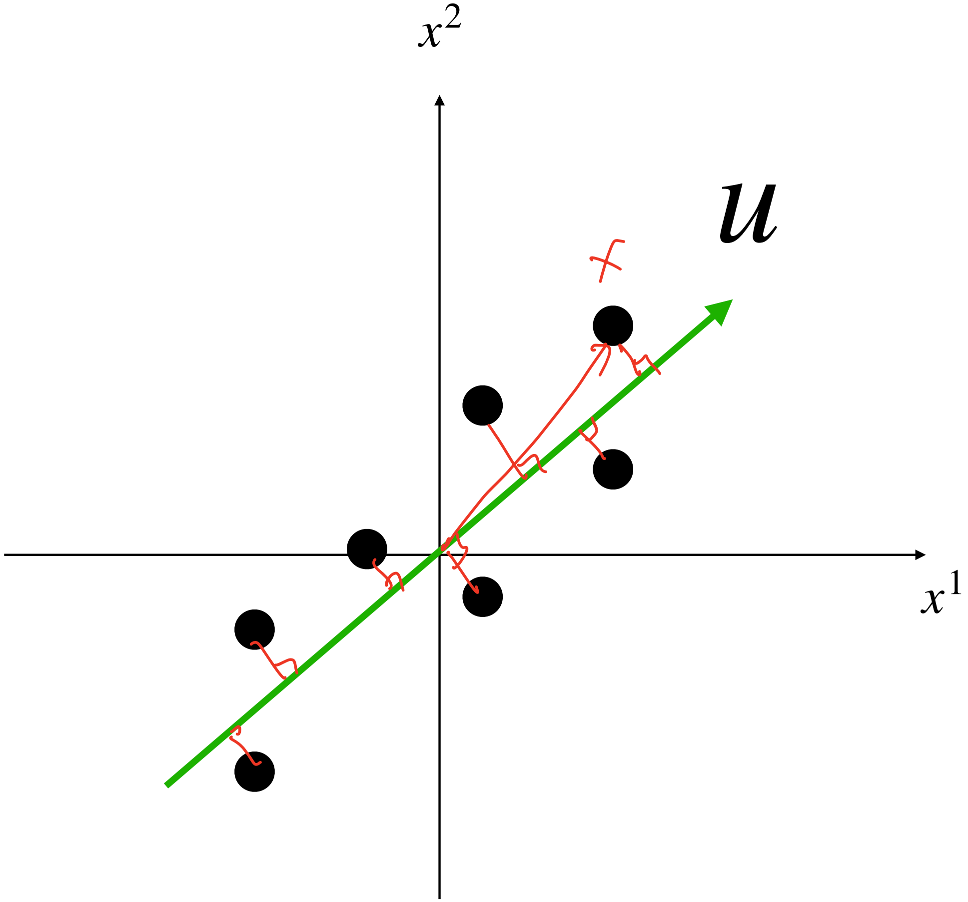


Compute the Principal Component

The outcome of data normalization



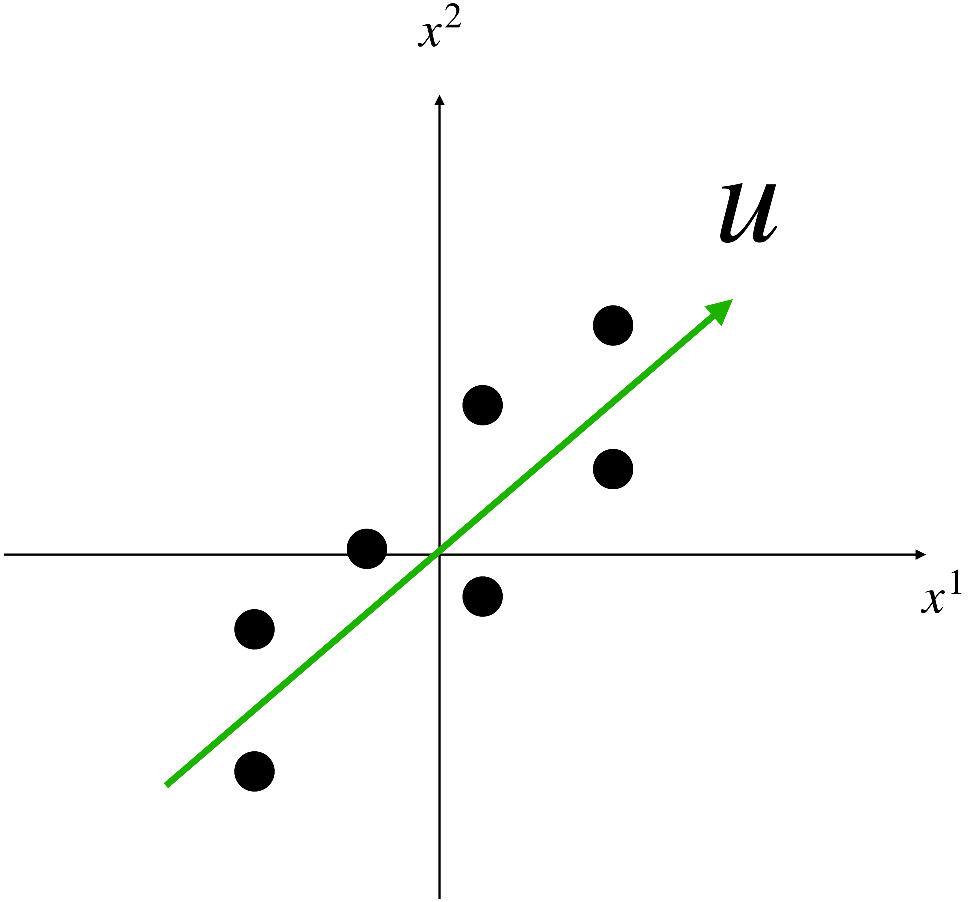
Compute the Principal Component



Step 2: compute first principle component

Intuition: find a direction such that the projected points are spread out

Compute the Principal Component

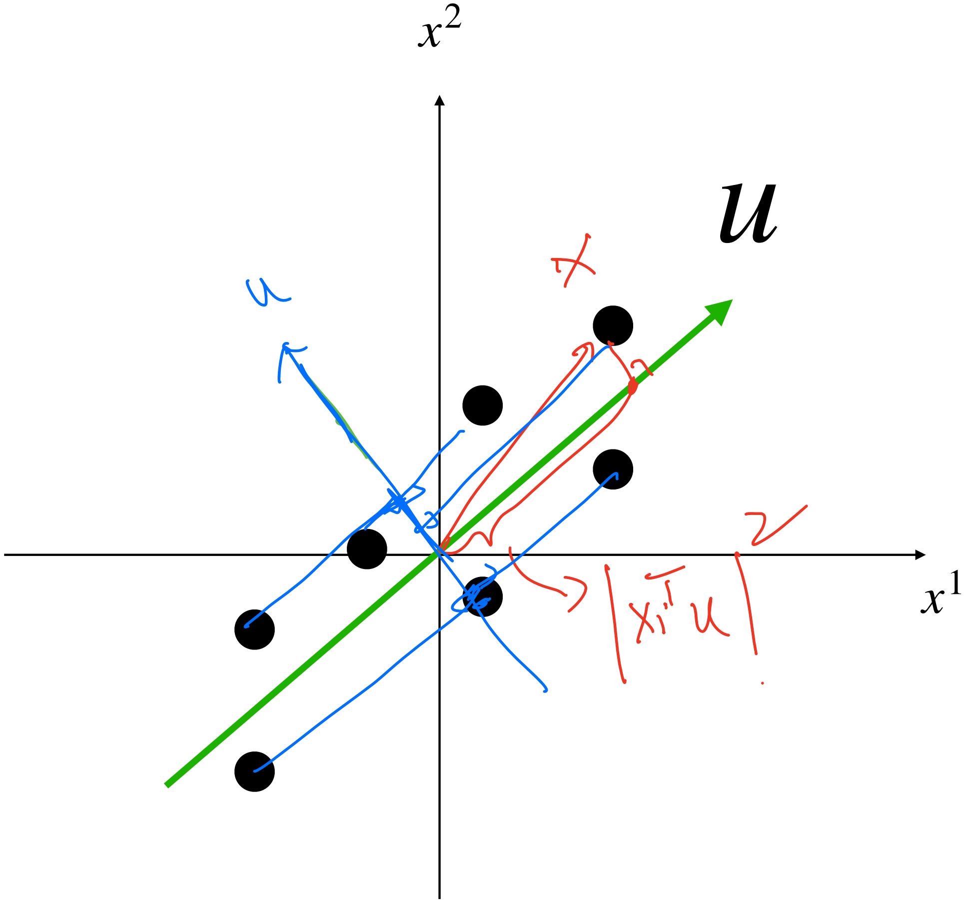


Step 2: compute first principle component

Intuition: find a direction such that the projected points are spread out

Mathematically, maximizes the variance of projected points

Compute the Principal Component



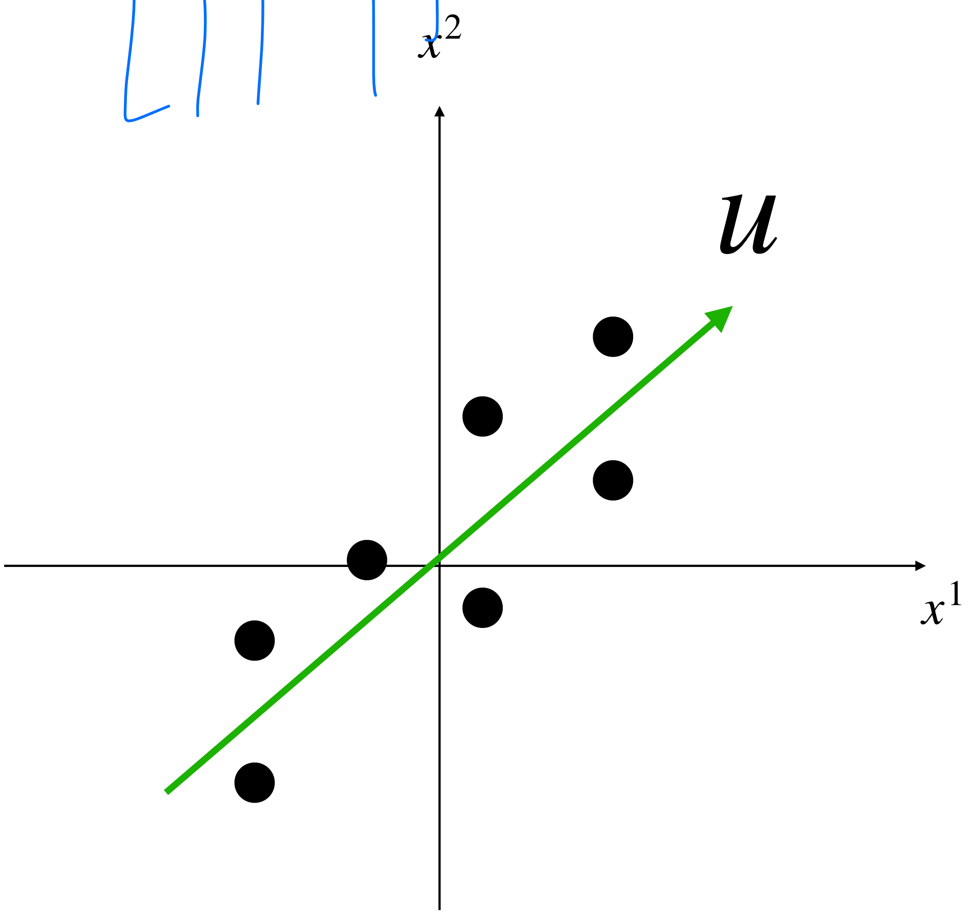
Step 2: compute first principle component

Intuition: find a direction such that the projected points are spread out

Mathematically, maximizes the variance of projected points

$$\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$$

Compute the Principal Component



$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_n \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{d\times n}$$

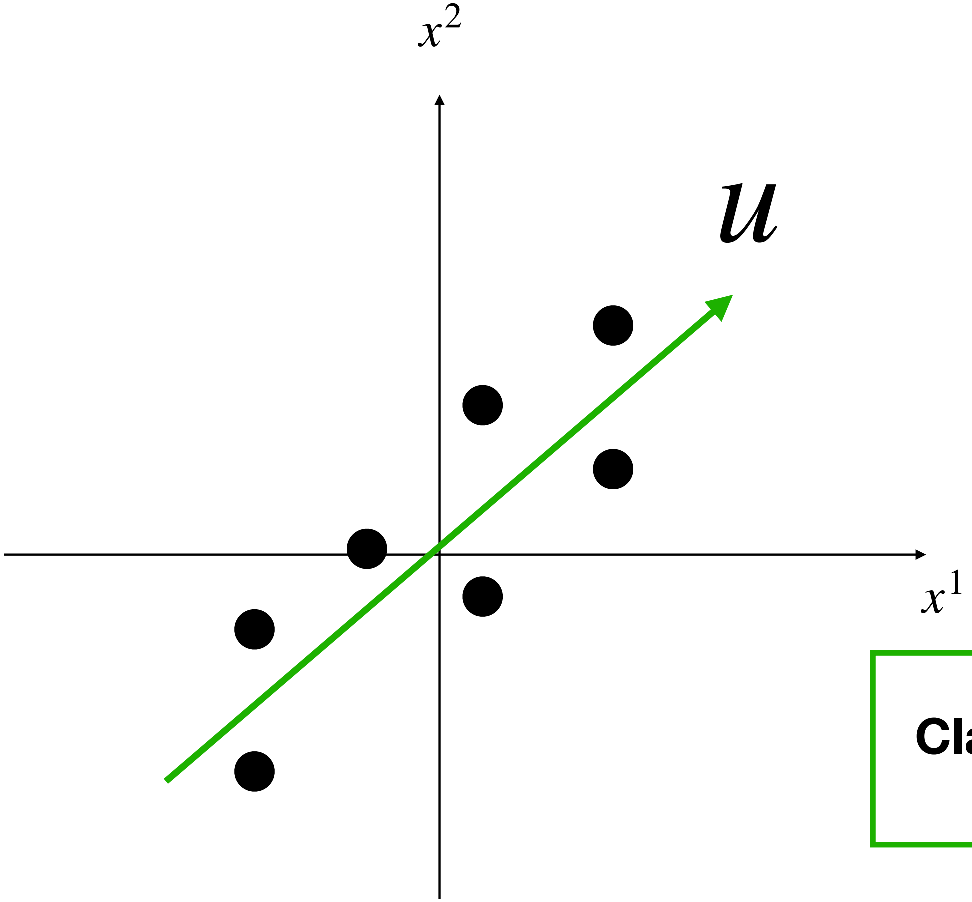
$$\arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$$

$$u^\top x_i x_i^\top u = u^\top (x_i x_i^\top) u$$

$$= \arg\max_{u:\|u\|_2=1} u^\top \underbrace{\left[\sum_{i=1}^{n} x_i x_i^\top\right]}_{XX^\top} u$$

$$\underbrace{XX^\top}_{\mathbb{R}^{d\times d}} = \underbrace{\sum_{i=1}^{n} x_i x_i^\top}_{\mathbb{R}^{d\times d}}$$

Compute the Principal Component



$$\arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$$

$$= \arg\max_{u:\|u\|_2=1} u^\top \underbrace{\left[\sum_{i=1}^{n} x_i x_i^\top\right]}_{XX^\top} u$$

Claim: the maximizer is the first eigenvector of $XX^\top$

Compute the Principal Component

Definition of Eigenvalue/Eigenvectors

(λ, u) is a pair of eigenvalue / eigenvector of $XX^\top$ if:

$$(XX^\top)u = \lambda u$$

$$\arg \max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$$

$$= \arg \max_{u:\|u\|_2=1} u^\top \underbrace{\left[\sum_{i=1}^{n} x_i x_i^\top \right]}_{XX^\top} u$$

Compute the Principal Component

Definition of Eigenvalue/Eigenvectors

(λ, u) is a pair of eigenvalue / eigenvector of $XX^\top$ if:

$$(XX^\top)u = \lambda u \Rightarrow u^\top(XX^\top)u = \lambda$$

$u^\top = \lambda$ $u^\top u = 1$

$$\arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$$

$$= \arg\max_{u:\|u\|_2=1} u^\top \underbrace{\left[\sum_{i=1}^{n} x_i x_i^\top\right]}_{XX^\top} u$$

Compute the Principal Component

Definition of Eigenvalue/Eigenvectors

(λ, u) is a pair of eigenvalue / eigenvector of $XX^\top$ if:

$$(XX^\top)u = \lambda u \;\Rightarrow\; u^\top(XX^\top)u = \lambda$$

Eigendecomposition:

$$XX^\top = U\Lambda U^\top$$

$$= \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_d \\ | & | & & | \end{bmatrix} \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_d \end{bmatrix} \begin{bmatrix} - & u_1^\top & - \\ - & u_2^\top & - \\ & \vdots & \\ - & u_d^\top & - \end{bmatrix}$$

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \cdots \geq \lambda_d \geq 0$$

$$\arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$$

$$= \arg\max_{u:\|u\|_2=1} u^\top \underbrace{\left[\sum_{i=1}^{n} x_i x_i^\top\right]}_{XX^\top} u$$

$$u^\top (XX^\top) u$$

$$u_1^\top (XX^\top) u_1 = \lambda_1$$

Compute the Principal Component

Definition of Eigenvalue/Eigenvectors

(λ, u) is a pair of eigenvalue / eigenvector of $XX^\top$ if:

$$(XX^\top)u = \lambda u \quad \Rightarrow u^\top (XX^\top) u = \lambda$$

Eigendecomposition:

$$XX^\top = U \Lambda U^\top$$

$$\arg \max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$$

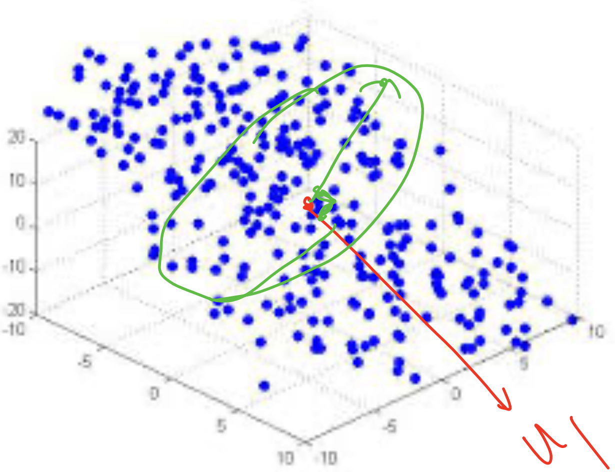
$$= \arg \max_{u:\|u\|_2=1} u^\top \underbrace{\left[\sum_{i=1}^{n} x_i x_i^\top \right]}_{XX^\top} u$$

Solution:

The arg max returns the first eigenvector of $XX^\top$

What about computing the second Principal component?

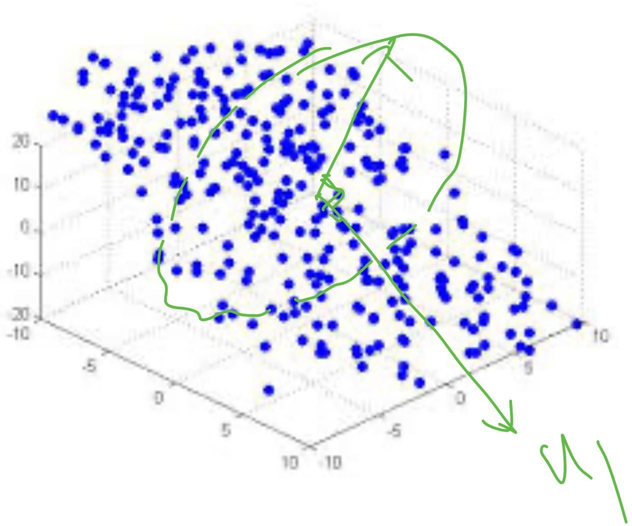
First Principle component $u_1 = \arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$



To compute the second PC:

What about computing the second Principal component?

First Principle component $u_1 = \arg \max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$

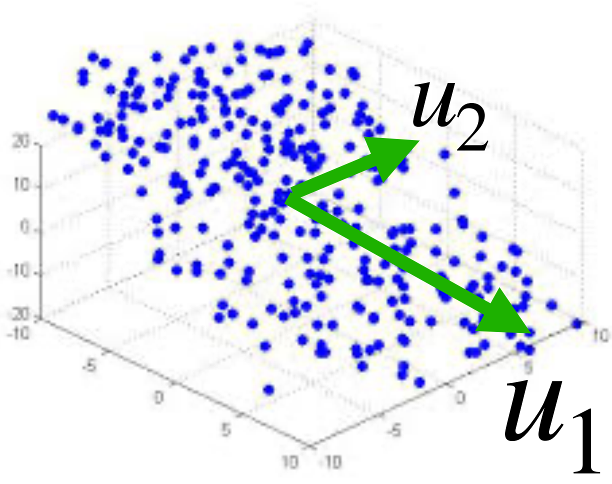


To compute the second PC:

Force constraints: $\|u_2\|_2 = 1,\ u_2^\top u_1 = 0$

What about computing the second Principal component?

First Principle component $u_1 = \arg \max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$

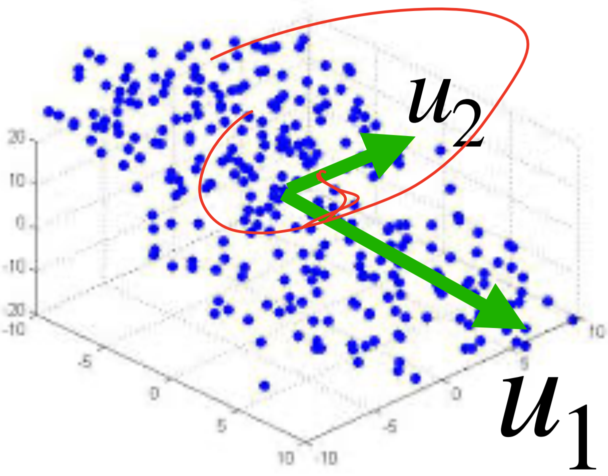


To compute the second PC:

Force constraints: $\|u_2\|_2 = 1, u_2^\top u_1 = 0$

What about computing the second Principal component?

First Principle component $u_1 = \arg \max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$



To compute the second PC:

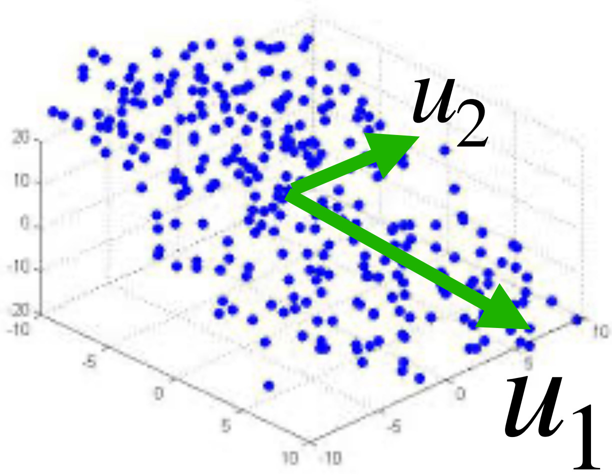
Force constraints: $\|u_2\|_2 = 1,\ u_2^\top u_1 = 0$

$$u_2 = \arg \max_{u:\|u\|_2=1, u^\top u_1 = 0} \sum_{i=1}^{n} (x_i^\top u)^2$$

length of the projected point

What about computing the second Principal component?

First Principle component $u_1 = \arg \max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$



To compute the second PC:

Force constraints: $\|u_2\|_2 = 1, u_2^\top u_1 = 0$

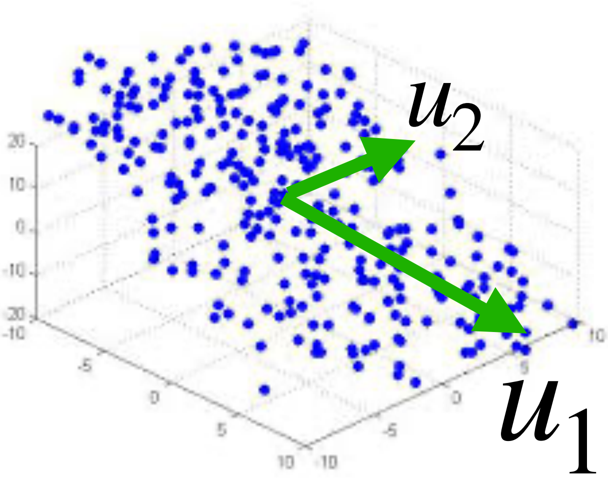
$$u_2 = \arg \max_{u:\|u\|_2=1, u^\top u_1=0} \sum_{i=1}^{n} (x_i^\top u)^2$$

$$= \arg \max_{u:\|u\|_2=1, u^\top u_1=0} u^\top (XX^\top) u$$

$u = u_2$

What about computing the second Principal component?

First Principle component $u_1 = \arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$



To compute the second PC:

Force constraints: $\|u_2\|_2 = 1,\, u_2^\top u_1 = 0$

$$u_2 = \arg\max_{u:\|u\|_2=1, u^\top u_1=0} \sum_{i=1}^{n} (x_i^\top u)^2$$

$$= \arg\max_{u:\|u\|_2=1, u^\top u_1=0} u^\top (XX^\top) u$$

Solution: u_2 will be the second eigenvector

Algorithm: PCA

Input: given the normalized dateset $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$, and parameter $K < d$

Algorithm: PCA

Input: given the normalized dateset $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$, and parameter $K < d$

1. Compute **Eigendecomposition** of $XX^\top := U\Lambda U^\top$

$$XX^\top = \sum_{i=1}^{n} x_i x_i^\top \in \mathbb{R}^{d \times d}$$

$$U = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_d \\ | & | & & | \end{bmatrix} \quad \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \ddots \; \lambda_d \end{pmatrix}$$

Algorithm: PCA

Input: given the normalized dateset $\mathcal{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$, and parameter $K < d$

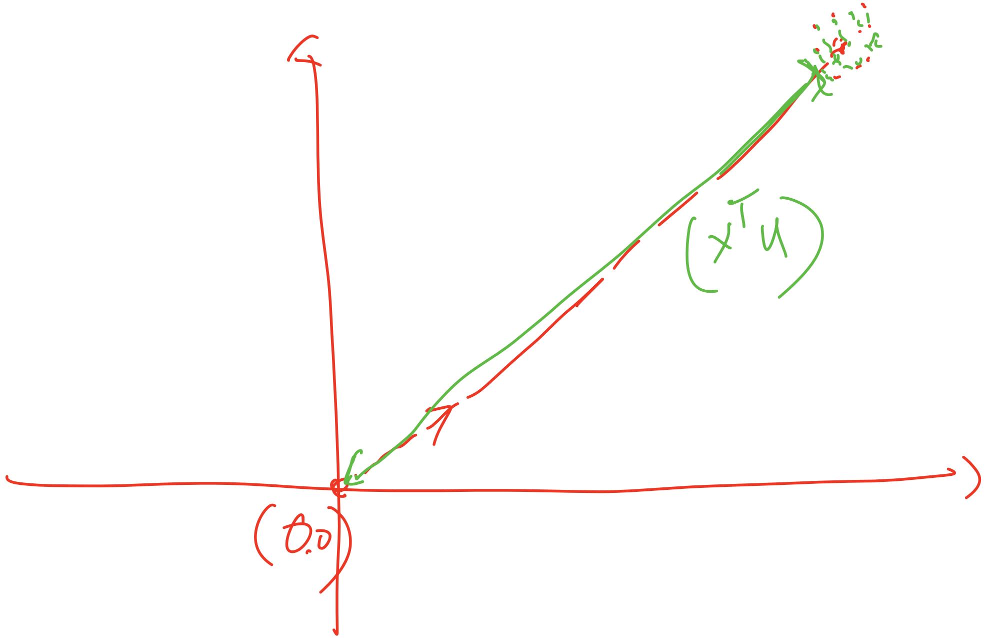
1. Compute **Eigendecomposition** of $XX^\top := U\Lambda U^\top$

2. Return the **top K eigenvectors** (corresponding to the top k largest eigenvalues)

$$U = [\underbrace{u_1, u_2, \ldots, u_k,}_{\text{top k eigenvectors}} \quad u_{k+1} \ldots, u_d], u_i \in \mathbb{R}^d$$

PCA

Q: what happens if we do not center the data?



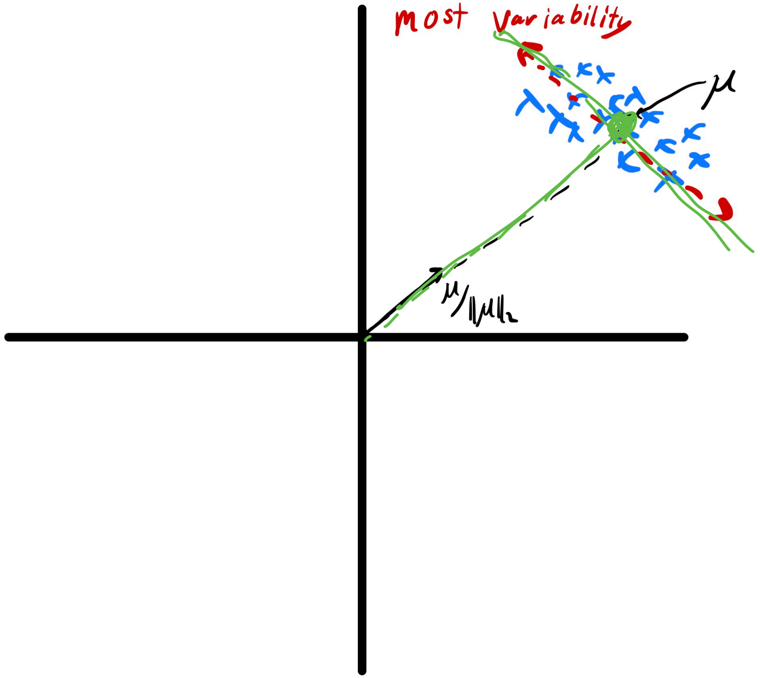
$\mu = \sum_{i=1}^{n} x_i / n$

$u = \mu / \|\mu\|$

PCA

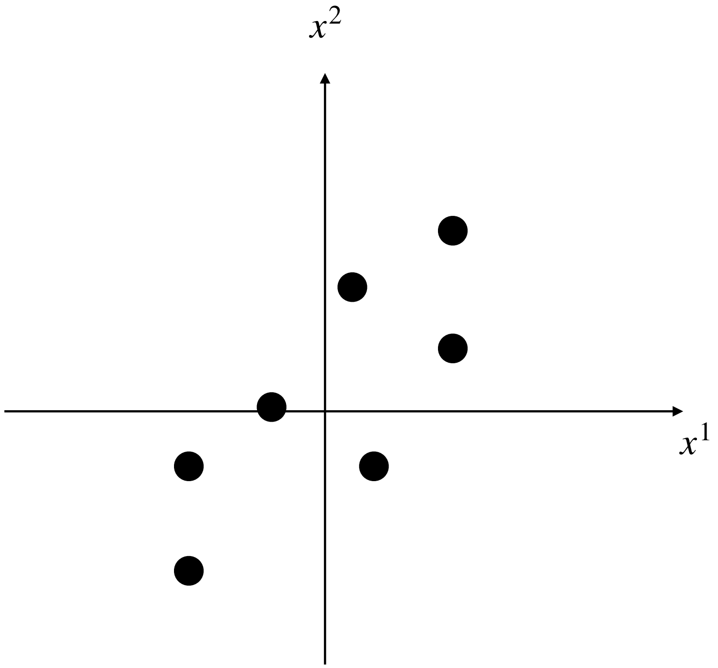
Q: what happens if we do not center the data?

A: the first principle might be just the mean of the data



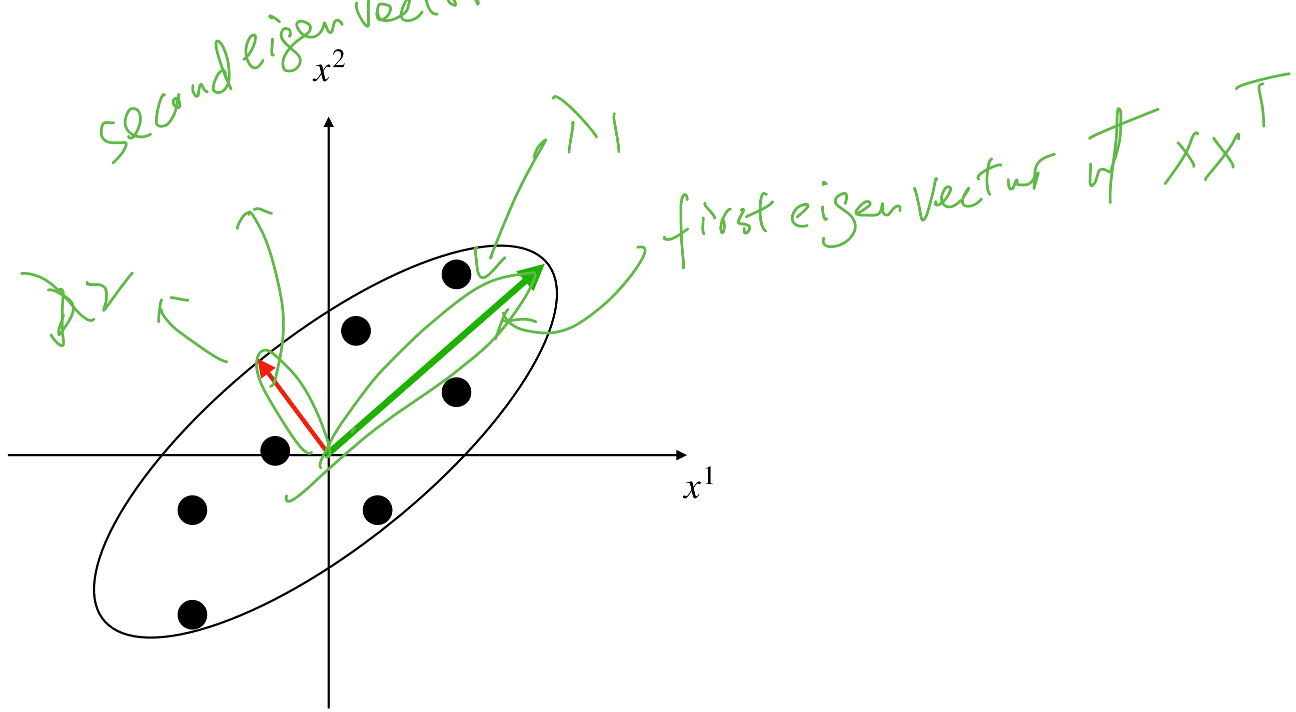
Geometric interpretation of the dataset

Denote $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$



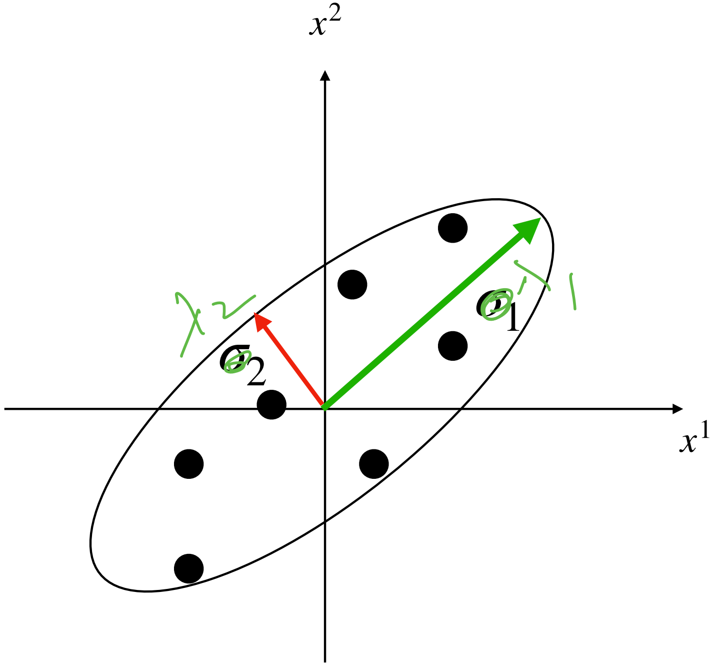
Geometric interpretation of the dataset

Denote $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$



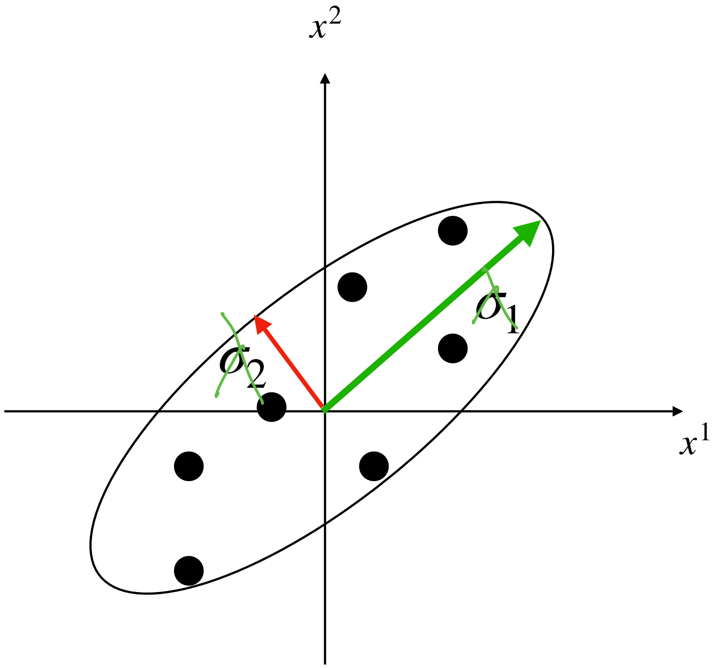
Geometric interpretation of the dataset

Denote $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$



Geometric interpretation of the dataset

Denote $X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}$



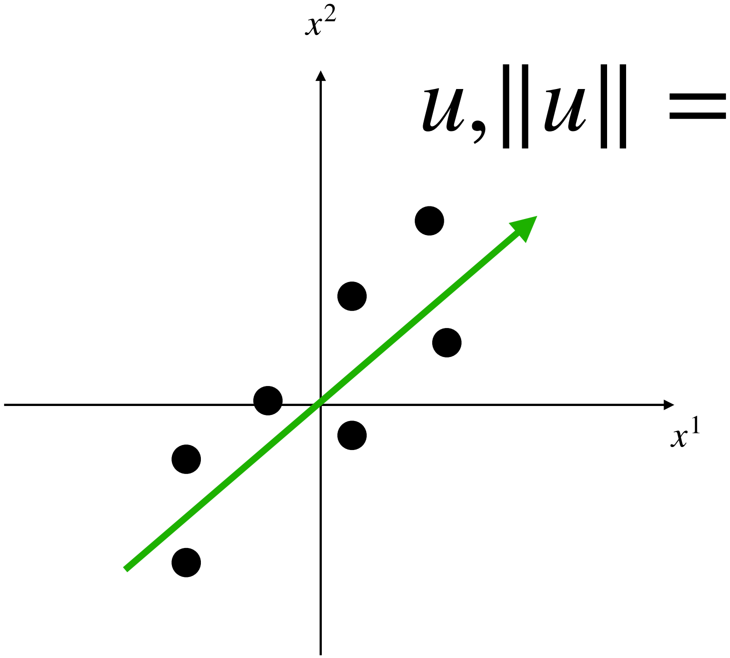
$\det(XX^\top)$ can be interpreted as Volume of this d-dim ellipsoid

$\det(XX^\top) = \prod_{i=1}^{d} \lambda_i$

$\text{Volume} = c \cdot \prod_{i=1}^{d} \lambda_i$

Think about PCA from a data re-construction perspective

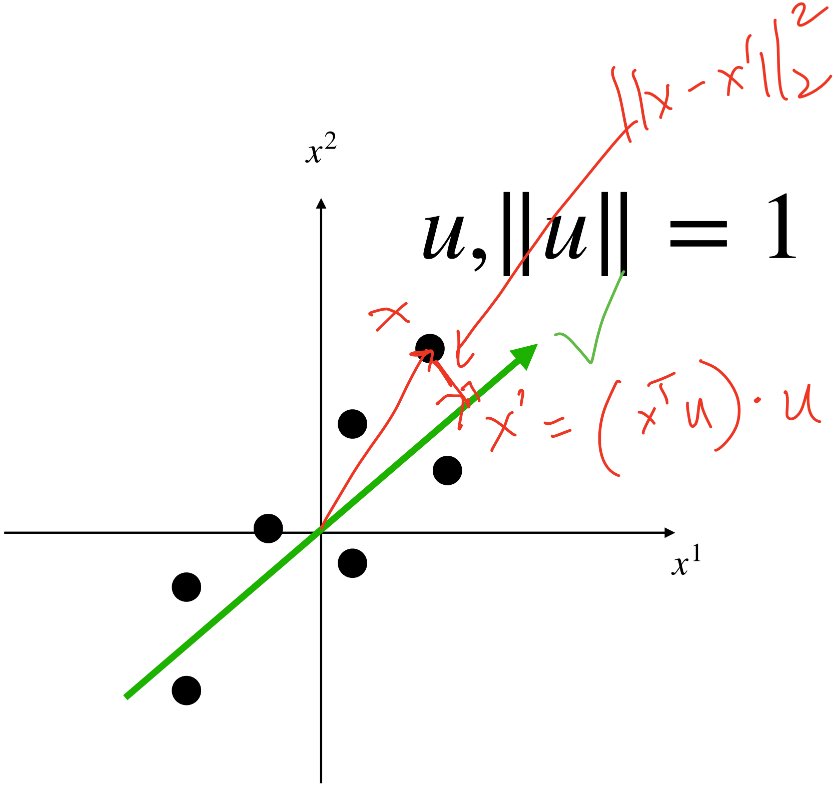
$$u, \|u\| = 1$$



$$XX^T = \sum_{i=1}^{n} x_i x_i^T$$

PSD

Think about PCA from a data re-construction perspective

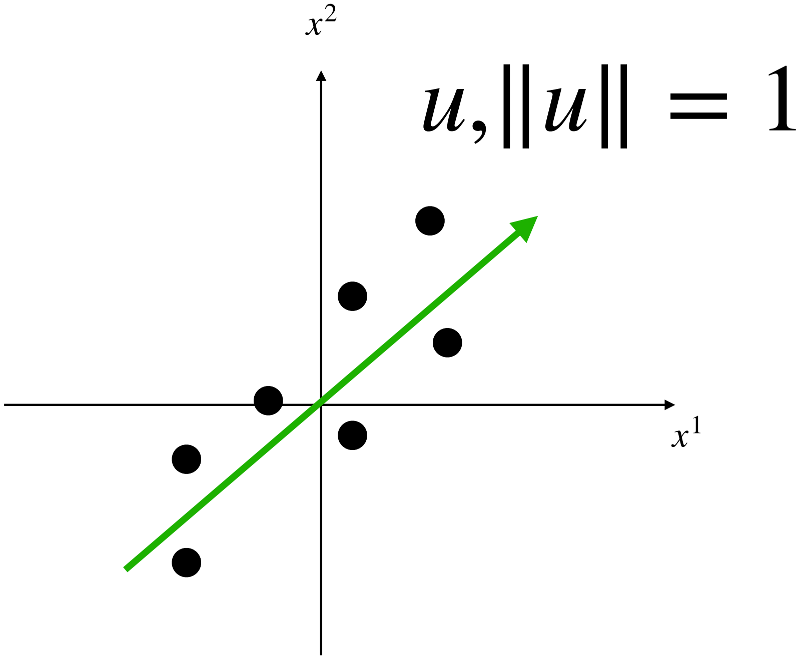


Represent x using u: $x \to (x^\top u)u$

(i.e., project x on u)

$\|x - x'\|_2^2$

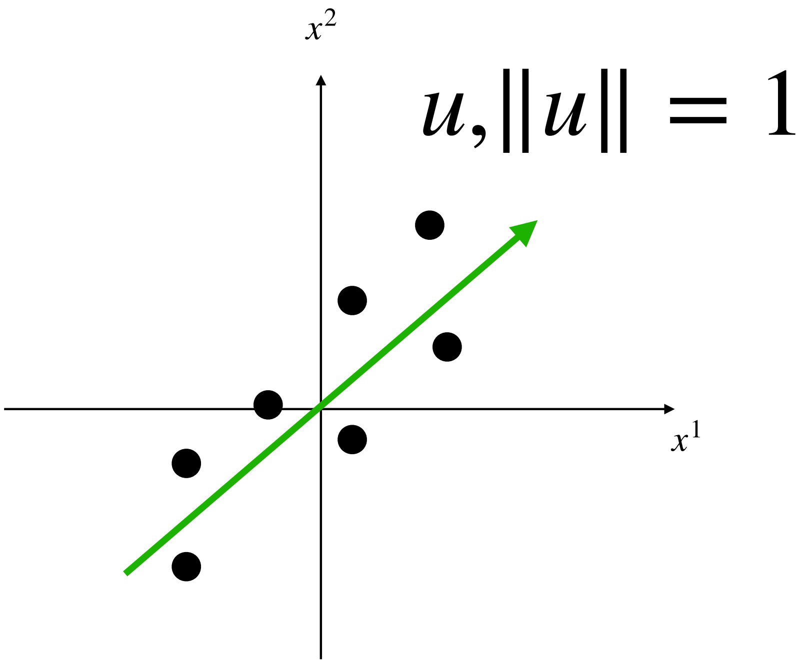
Think about PCA from a data re-construction perspective



Represent x using u: $x \to (x^\top u)u$
(i.e., project x on u)

Reconstruct error: $\|(x^\top u)u - x\|_2^2$

Think about PCA from a data re-construction perspective

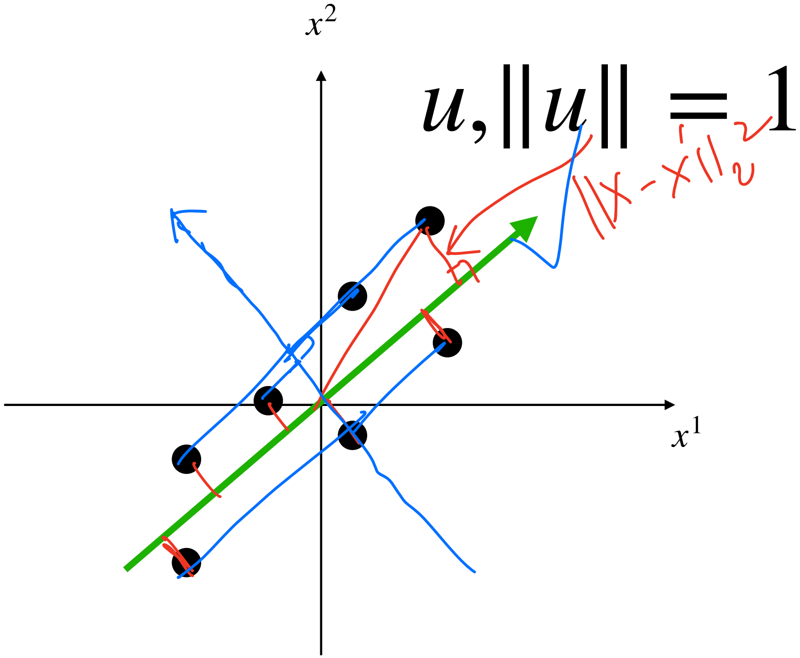


Represent x using u: $x \to (x^\top u)u$
(i.e., project x on u)

Reconstruct error: $\|(x^\top u)u - x\|_2^2$

PCA first principle component procedure : find u that minimizes the total reconstruction error

Think about PCA from a data re-construction perspective



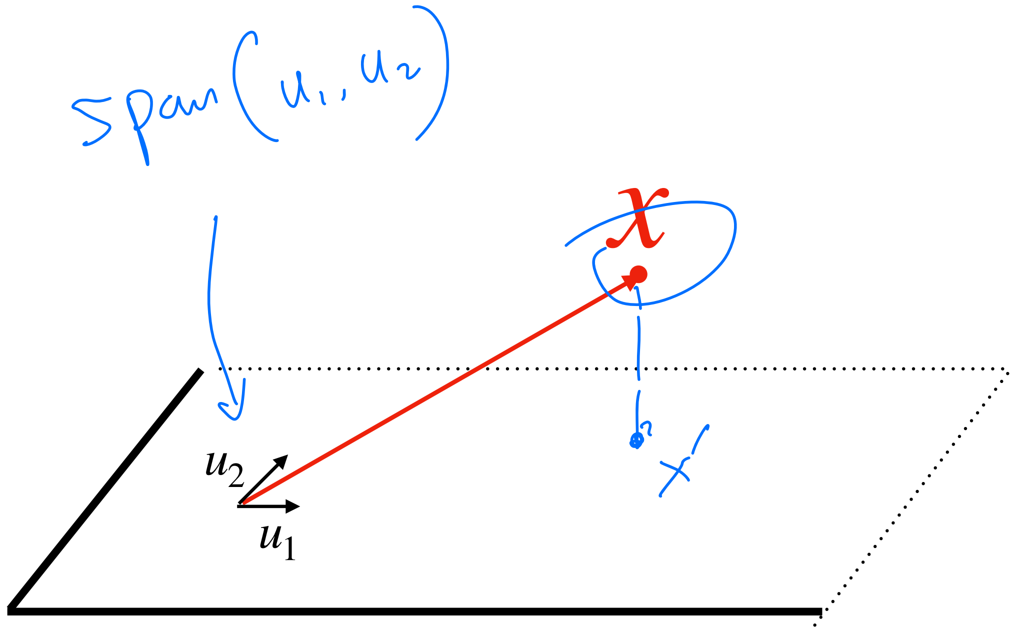
Represent x using u: $x \to (x^\top u)u$
(i.e., project x on u)

Reconstruct error: $\|(x^\top u)u - x\|_2^2$

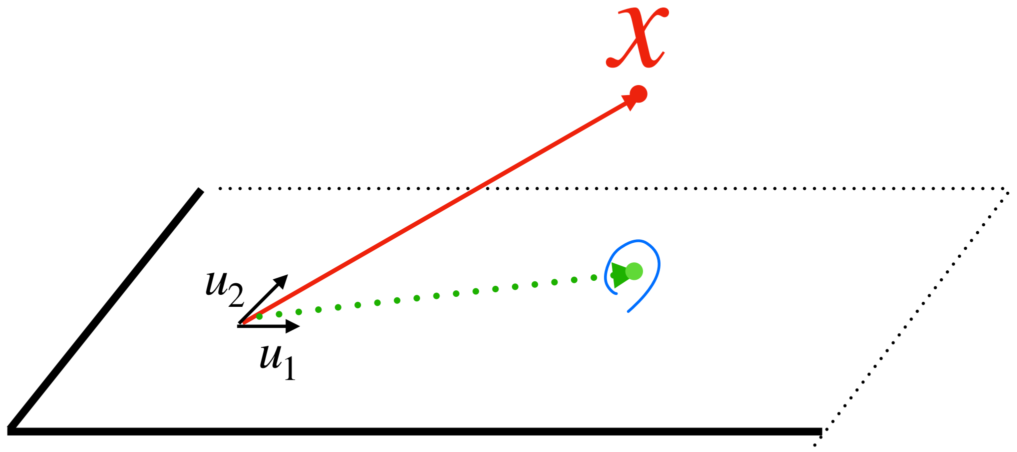
PCA first principle component procedure : find u that minimizes the total reconstruction error

$$\arg\min_{u:\|u\|_2=1} \sum_{i=1}^{n} \|uu^\top x_i - x_i\|_2^2$$

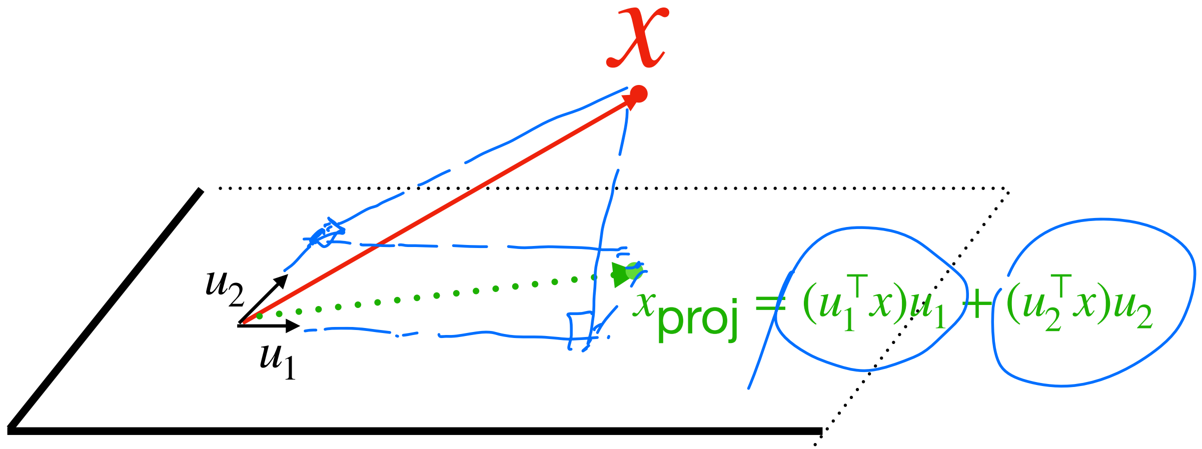
Think about PCA from a data re-construction perspective



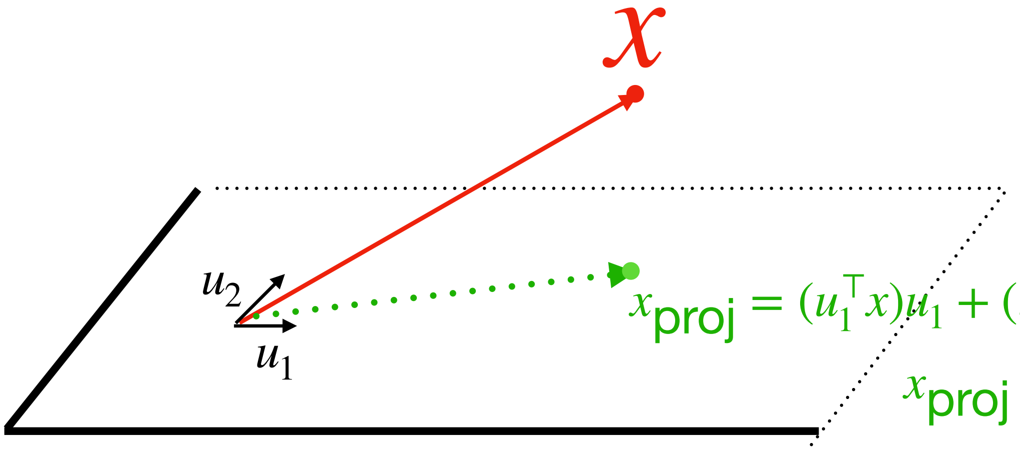
Think about PCA from a data re-construction perspective



Think about PCA from a data re-construction perspective



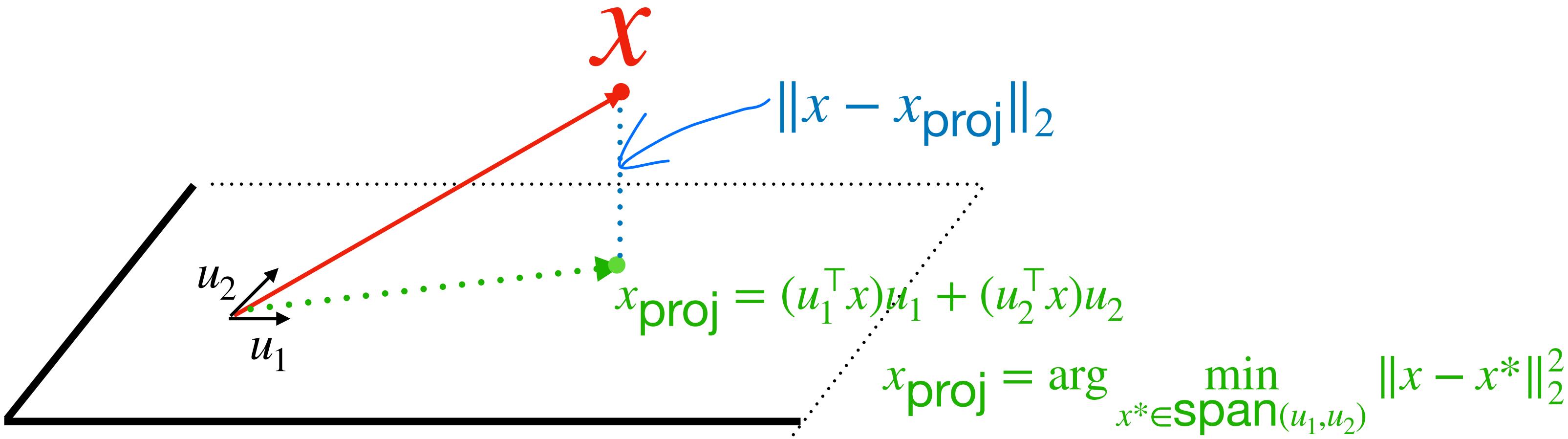
Think about PCA from a data re-construction perspective



$$x_{\text{proj}} = (u_1^\top x)u_1 + (u_2^\top x)u_2$$

$$x_{\text{proj}} = \arg\min_{x^* \in \text{span}(u_1, u_2)} \|x - x^*\|_2^2$$

Think about PCA from a data re-construction perspective

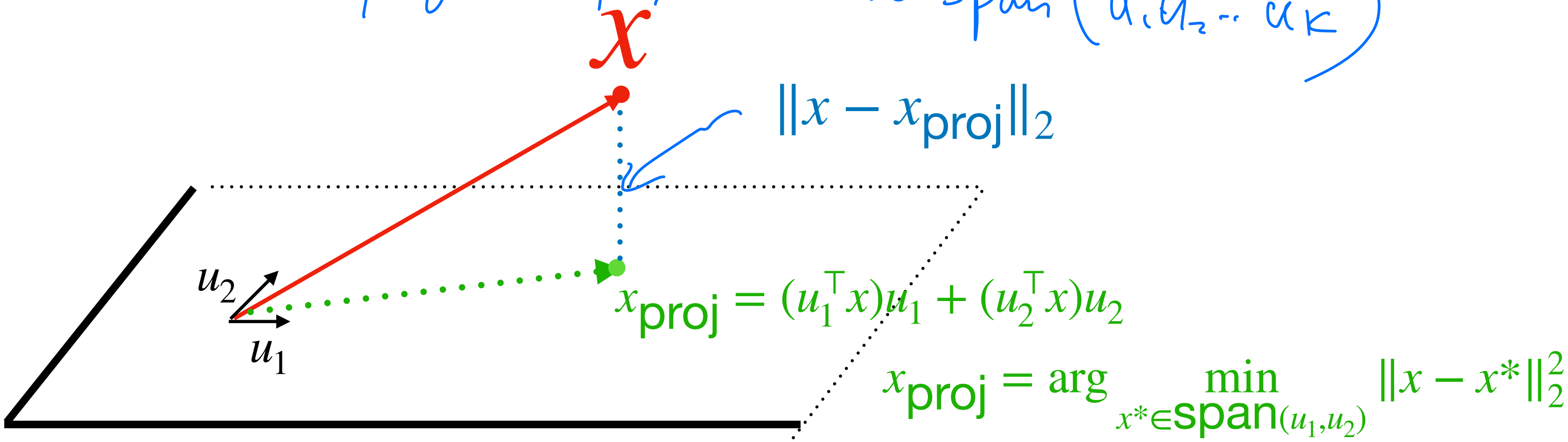


Think about PCA from a data re-construction perspective

Another way to think about PCA is to find $u_1, u_2, \ldots, u_k$ to minimize re-construction error

$$\min_{u_1, u_2, \ldots, u_k} \sum_{i=1}^{n} \left\| \sum_{j=1}^{k} (u_j^\top x_i) u_j - x_i \right\|_2^2, \text{ s.t. } \forall i : u_i^\top u_i = 1, \text{ and } u_i^\top u_j = 0, \forall i \neq j$$

↳ projection of x_i onto the span $(u_1, u_2, \ldots u_k)$



Outline for today:

1. Intro of PCA

2. PCA via eigendecomposition

3. Example of PCA: eigenfaces

Application of PCA: Eigenfaces

$$\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^{64^2}$$

$XX^T \in \mathbb{R}^{d \times d}$

$d = 64^2$

$u_i \in \mathbb{R}^d$

Reshape it to matrix $M \in \mathbb{R}^{64 \times 64}$

64×64

$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix}$ Reshape → $\begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{bmatrix}$

← Reshape back to matrix

Application of PCA: Eigenfaces

The top 15 Eigenfaces (top 15 eigenvectors reshaped into 64×64 matrices)

Application of PCA: Eigenfaces

Reconstruct original images using Eigenfaces

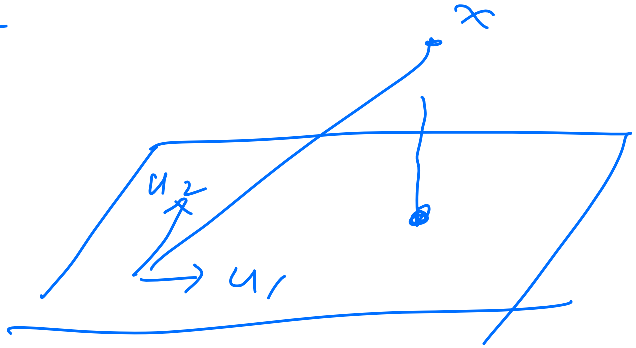
Application of PCA: Eigenfaces

Reconstruct original images using Eigenfaces

Given $x \in \mathbb{R}^{64^2}$, and top K eigenvectors $u_1, \ldots, u_k$, we can approximate x as follows:

$$x' = (x^\top u_1)u_1 + (x^\top u_2)u_2 + \ldots + (x^\top u_k)u_k$$

projected point



Application of PCA: Eigenfaces

Reconstruct original images using Eigenfaces

Given $x \in \mathbb{R}^{64^2}$, and top K eigenvectors $u_1, \ldots, u_k$, we can approximate x as follows:

$$x' = (x^\top u_1)u_1 + (x^\top u_2)u_2 + \ldots + (x^\top u_k)u_k$$

(Q: when $k \to 64^2$, we should expect $x' \to x$, why?)

Application of PCA: Eigenfaces

Reconstruct original images using Eigenfaces

Given $x \in \mathbb{R}^{64^2}$, and top K eigenvectors $u_1, \ldots, u_k$, we can approximate x as follows:

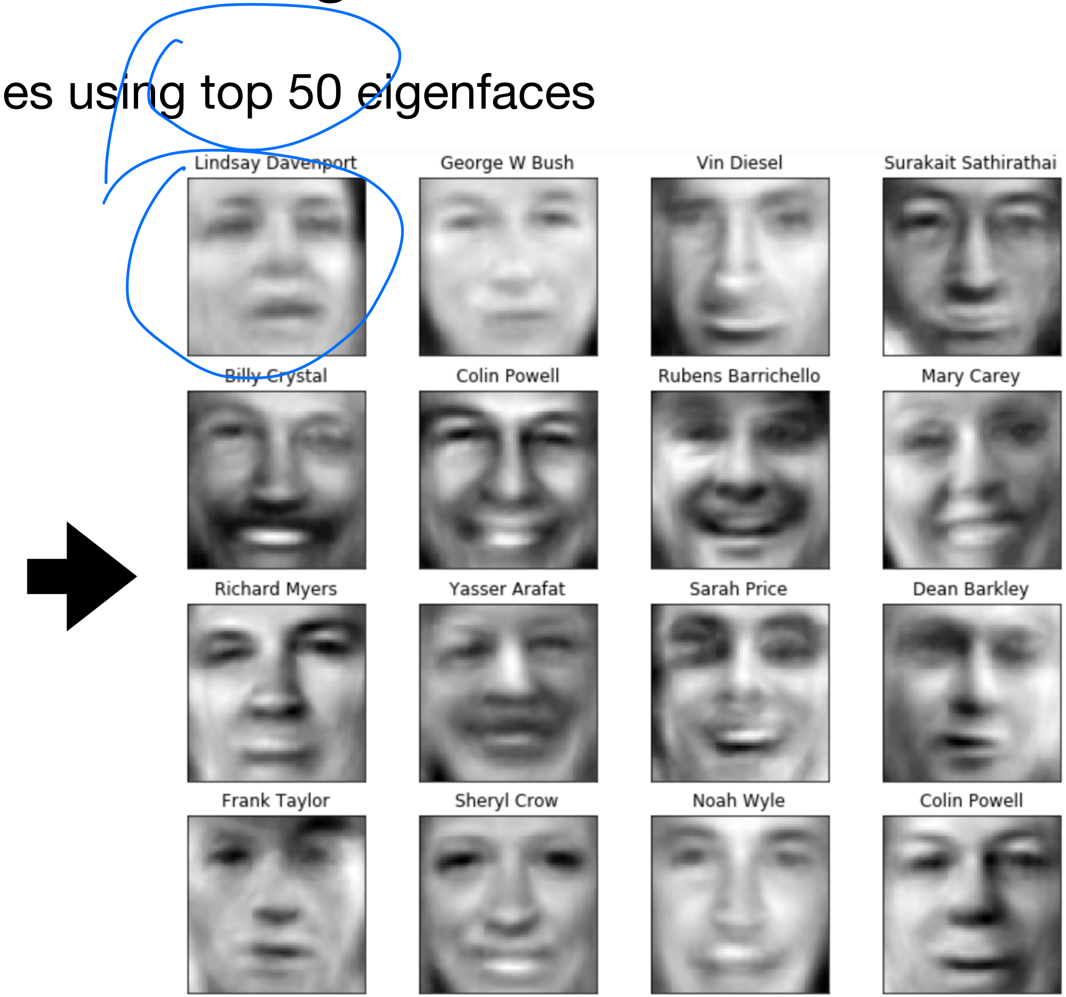
$$x' = (x^\top u_1)u_1 + (x^\top u_2)u_2 + \ldots + (x^\top u_k)u_k$$

(Q: when $k \to 64^2$, we should expect $x' \to x$, why?)

We will check if the visualization of x' is similar to that of x

Application of PCA: Eigenfaces

Reconstruct images using top 50 eigenfaces



Application of PCA: Eigenfaces

Reconstruct images using top 200 eigenfaces

Summary

1. The PCA algorithm: Eigendecomposition on $XX^\top$

2. Dimensionality reduction and Data reconstruction via PCA