Principal Component Analysis

Announcement:

Recap on K-means

Given any K disjoint groups $C_1, C_2, \ldots, C_K$, and any K centroids $\mu_1, \ldots, \mu_K$, define

$$\ell(\{C_i\}, \{\mu_i\}) = \sum_{i=1}^{K} \left[\sum_{x \in C_i} \|x - \mu_i\|_2^2 \right]$$

Recap on K-means

Given any K disjoint groups $C_1, C_2, \ldots, C_K$, and any K centroids $\mu_1, \ldots, \mu_K$, define

$$\ell(\{C_i\}, \{\mu_i\}) = \sum_{i=1}^{K} \left[\sum_{x \in C_i} \|x - \mu_i\|_2^2 \right]$$

K-means Algorithm:

Initialize $\mu_1, \ldots, \mu_K$

Repeat until convergence:

Recap on K-means

Given any K disjoint groups $C_1, C_2, \ldots, C_K$, and any K centroids $\mu_1, \ldots, \mu_K$, define

$$\ell(\{C_i\}, \{\mu_i\}) = \sum_{i=1}^{K} \left[\sum_{x \in C_i} \|x - \mu_i\|_2^2 \right]$$

K-means Algorithm:

Initialize $\mu_1, \ldots, \mu_K$

Repeat until convergence:

$$C_1, \ldots, C_K = \arg\min_{C_1, \ldots, C_k} \ell(\{C_i\}, \{\mu_i\})$$

Recap on K-means

Given any K disjoint groups $C_1, C_2, \ldots, C_K$, and any K centroids $\mu_1, \ldots, \mu_K$, define

$$\ell(\{C_i\}, \{\mu_i\}) = \sum_{i=1}^{K} \left[\sum_{x \in C_i} \|x - \mu_i\|_2^2 \right]$$

K-means Algorithm:

Initialize $\mu_1, \ldots, \mu_K$

Repeat until convergence:

$$C_1, \ldots, C_K = \arg \min_{C_1, \ldots, C_k} \ell(\{C_i\}, \{\mu_i\})$$

$$\mu_1, \ldots, \mu_K = \arg \min_{\mu_1, \ldots, \mu_k} \ell(\{C_i\}, \{\mu_i\})$$

Outline for today:

1. Intro of PCA

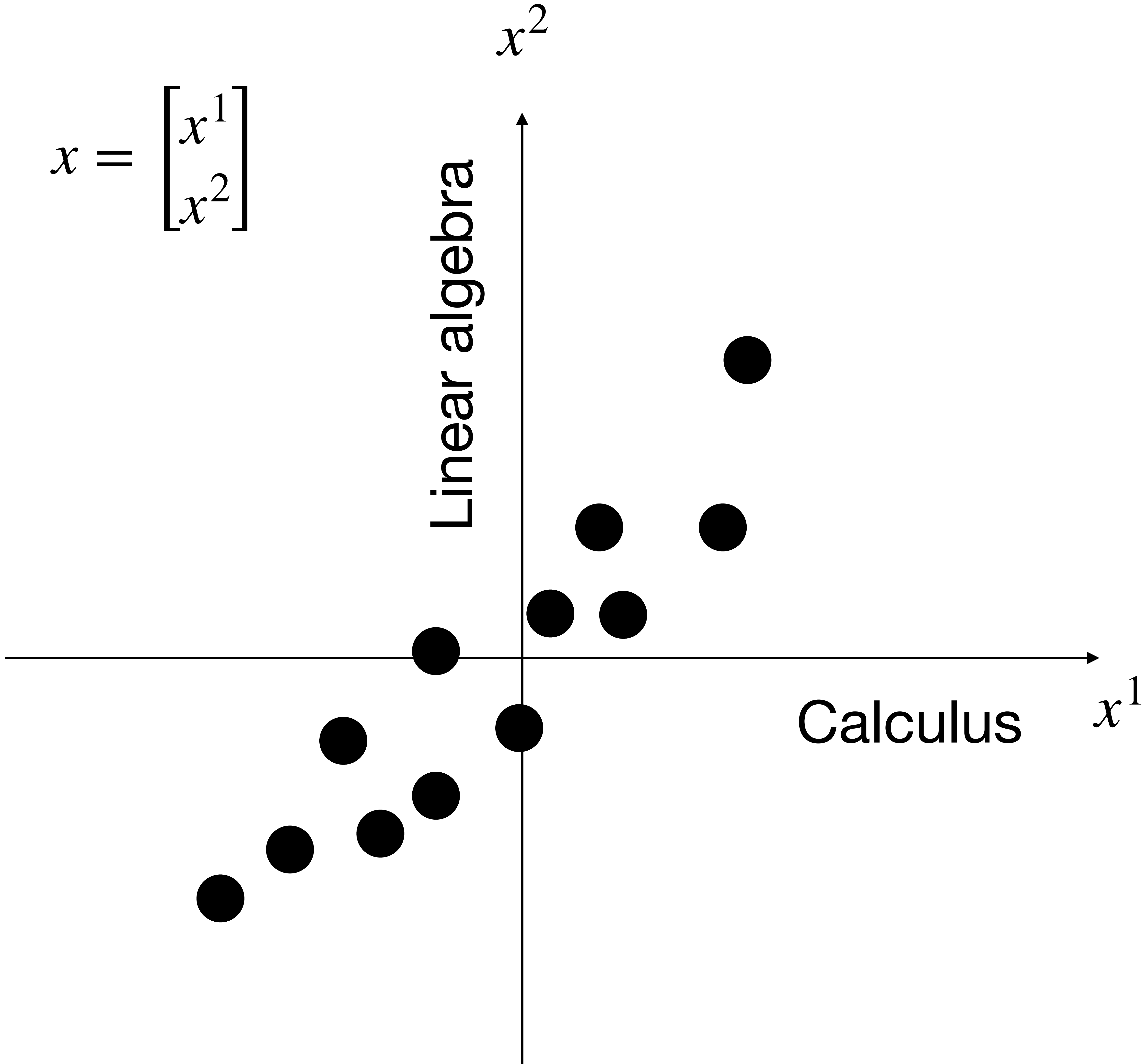
2. PCA via eigendecomposition

3. Example of PCA: eigenfaces

Data compression

Goal: reduce high dimensional data to low dimensional

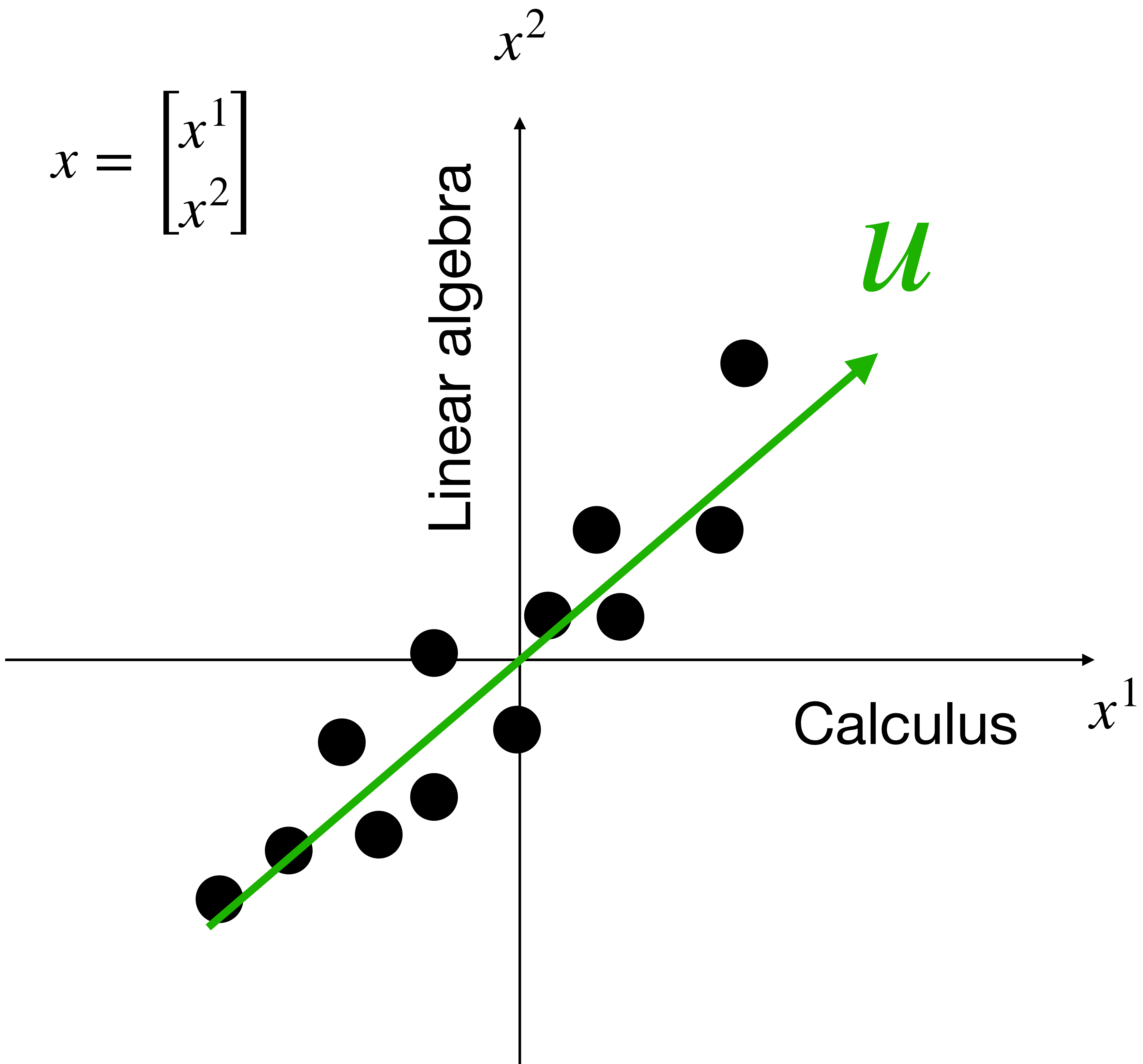
$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$$



Data compression

Goal: reduce high dimensional data to low dimensional

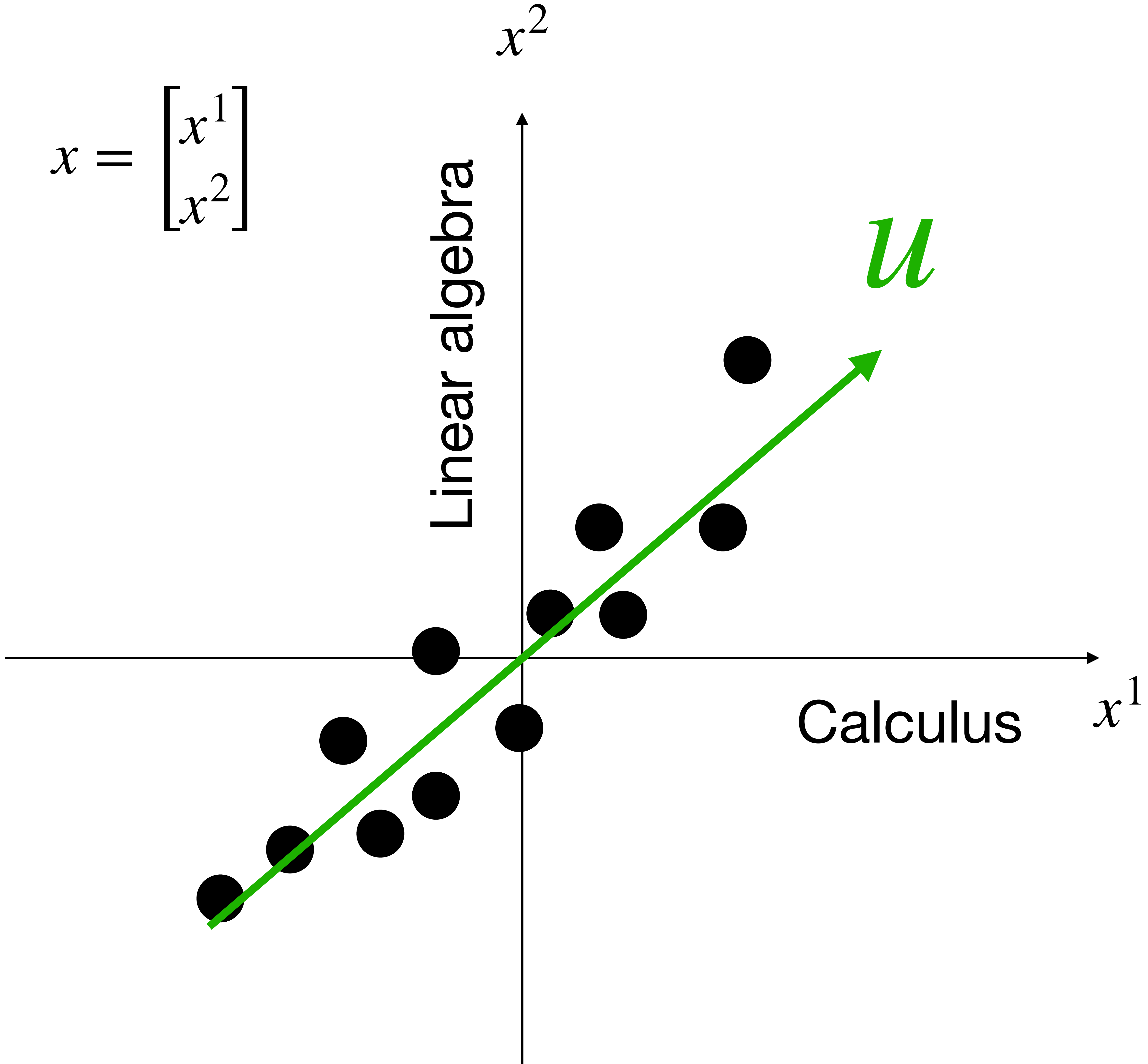
$$x = \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}$$



Data compression

Goal: reduce high dimensional data to low dimensional

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

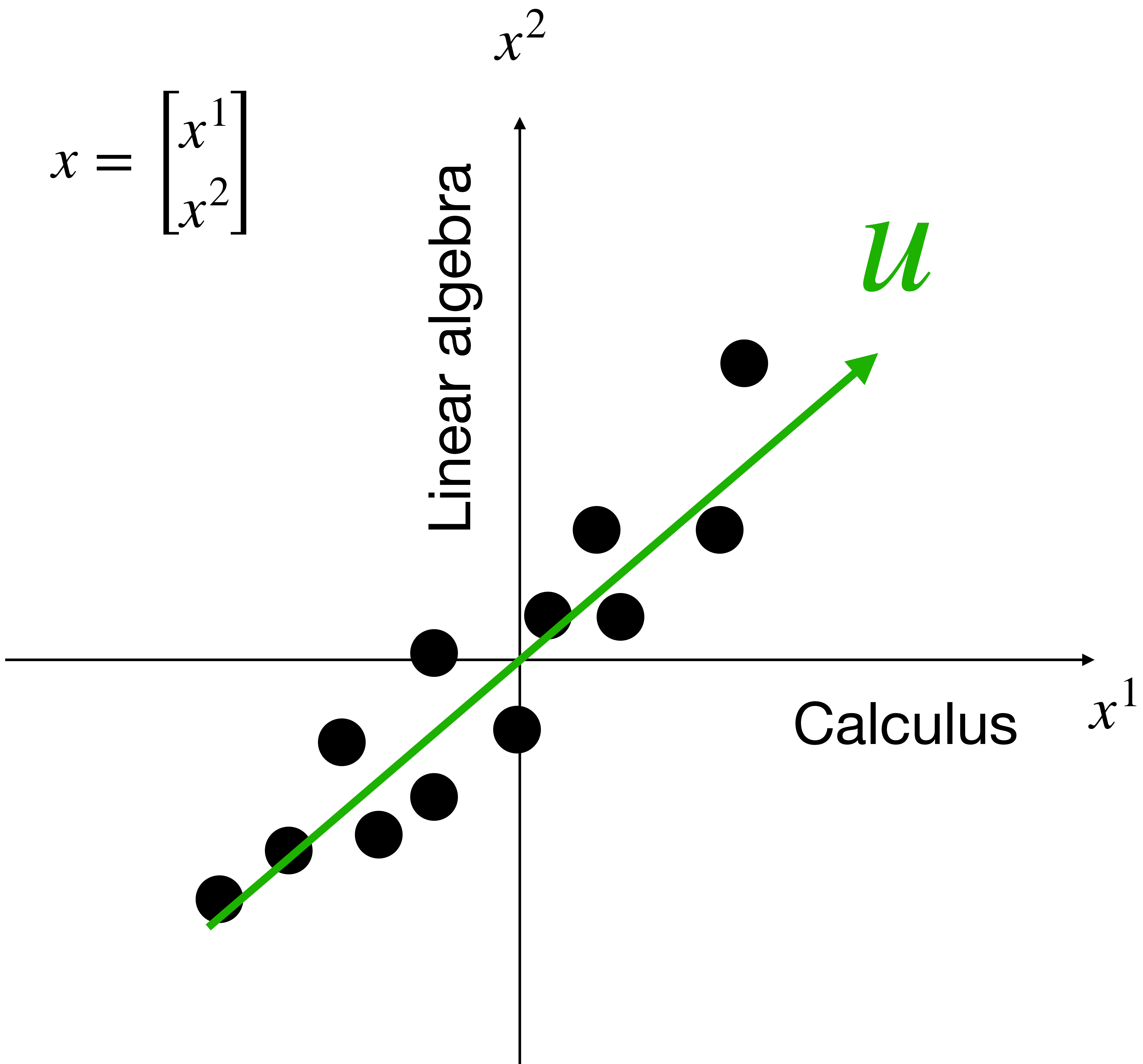


Normalize u s.t. $\|u\|_2 = 1$

Data compression

Goal: reduce high dimensional data to low dimensional

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



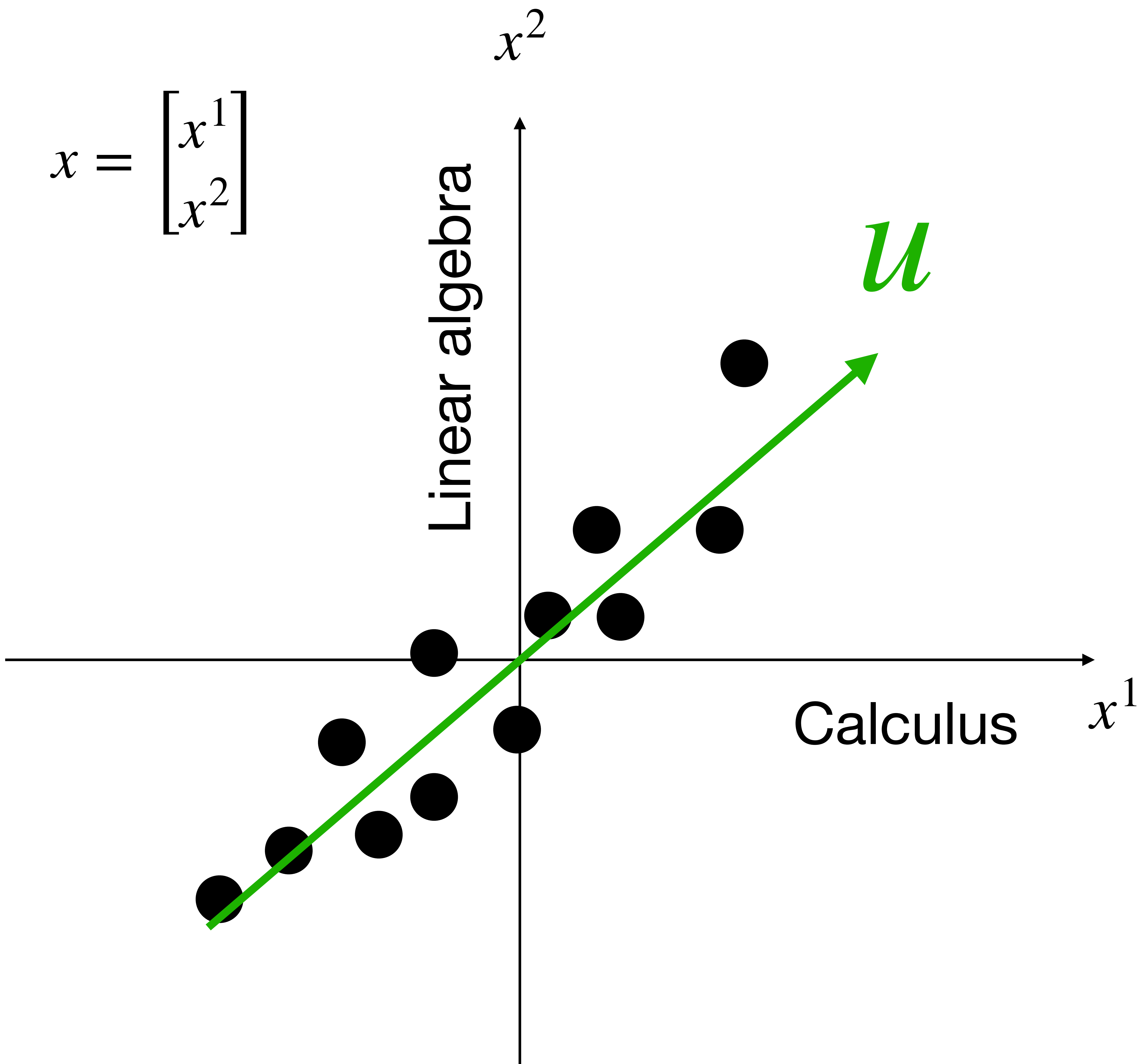
Normalize u s.t. $\|u\|_2 = 1$

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: $z := x^\top u$

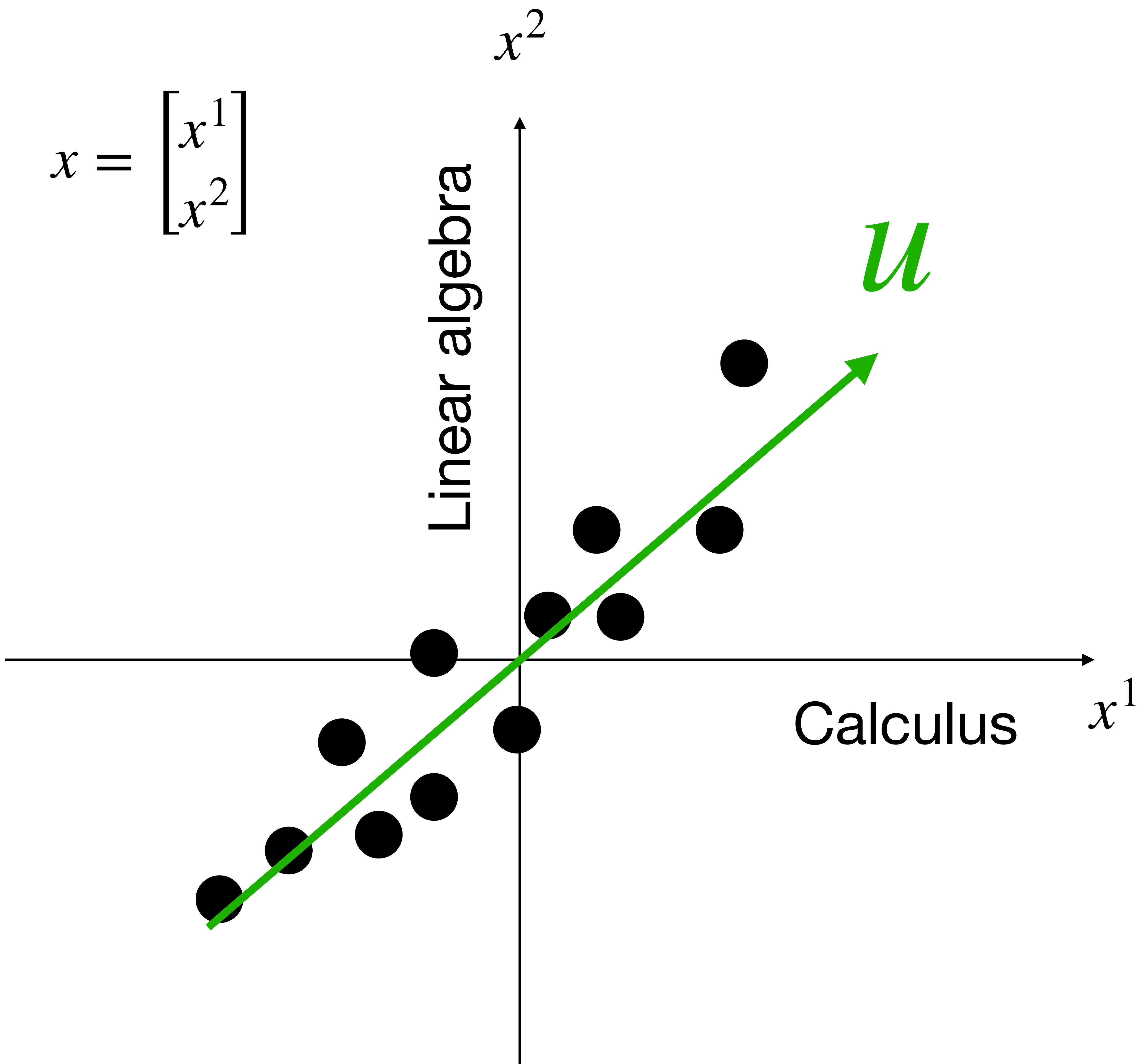
Dim-reduction:

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

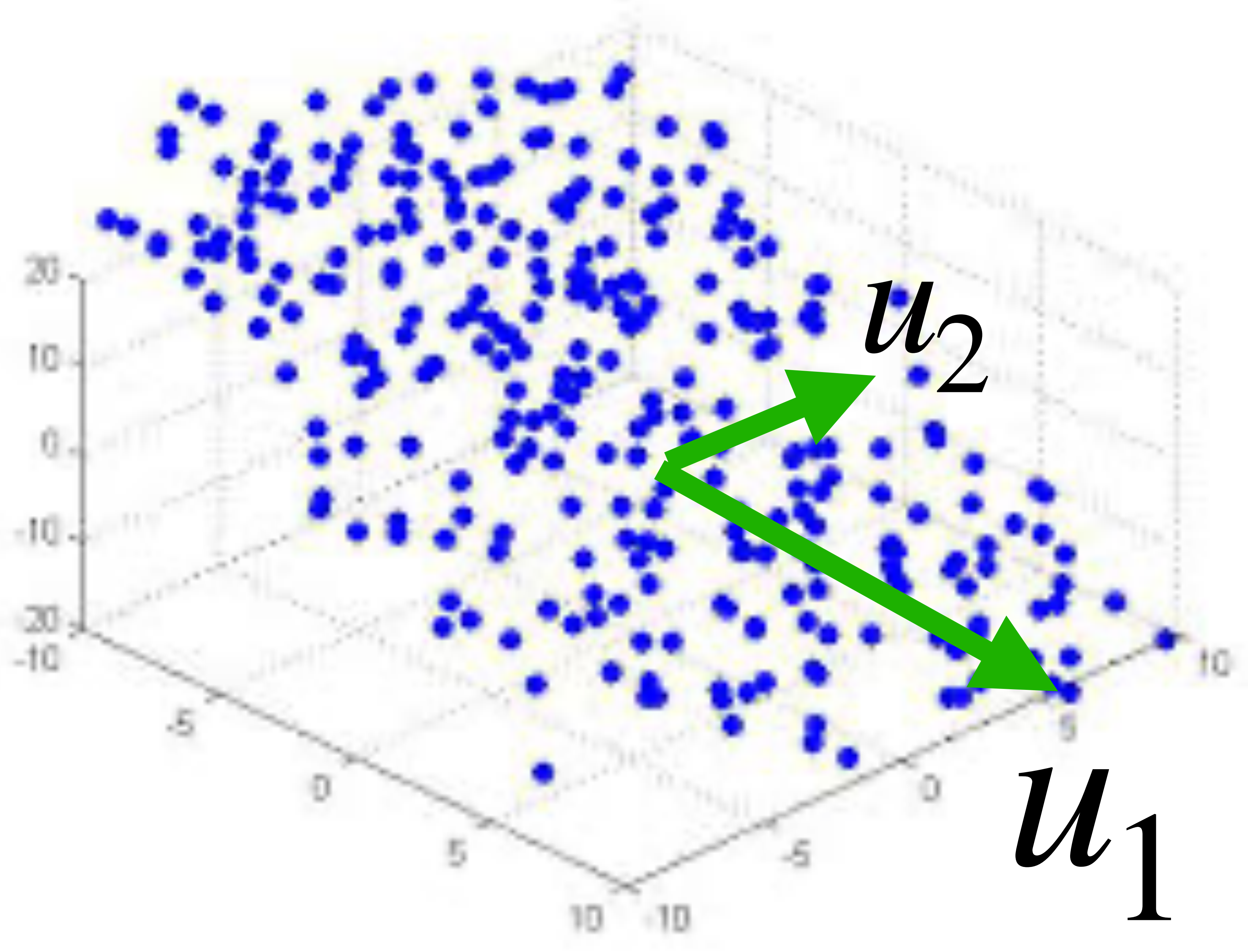
We get a 1-d dataset

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

Data compression

Goal: reduce high dimensional data to low dimensional

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

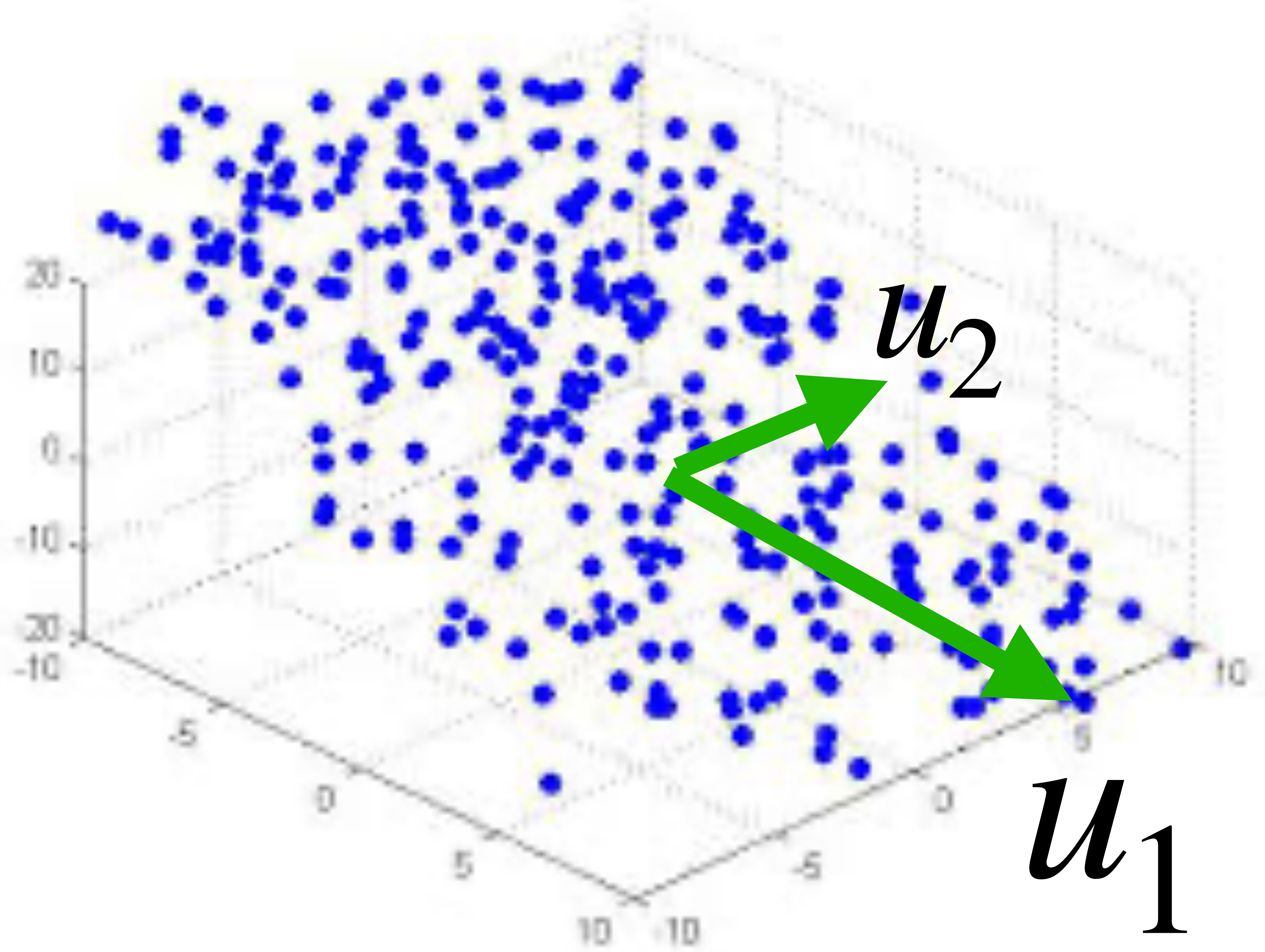


Reduce data from 3-d to 2-d:

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:

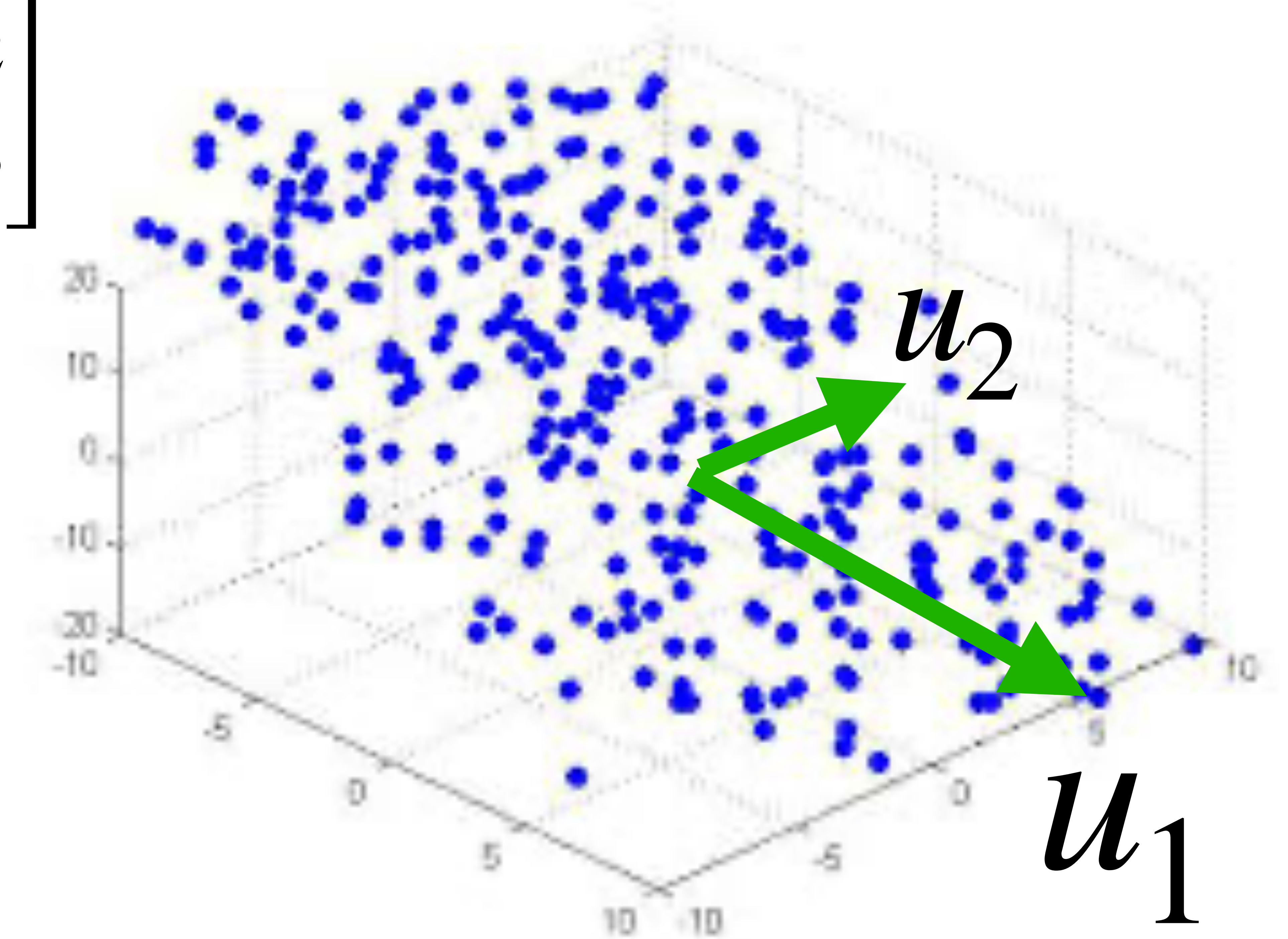
$$u_1 \in \mathbb{R}^3, u_2 \in \mathbb{R}^3$$

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

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:

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

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

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

Outline for today:

1. Intro of PCA

2. PCA via eigendecomposition

3. Example of PCA: eigenfaces

Compute the Principal Component

Setup

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

Compute the Principal Component

Setup

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

Assume data is centered, i.e.,

$$\sum_{i=1}^{n} x_i / n = 0$$

(Otherwise, compute the mean and shift every data point)

Compute the Principal Component

Setup

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

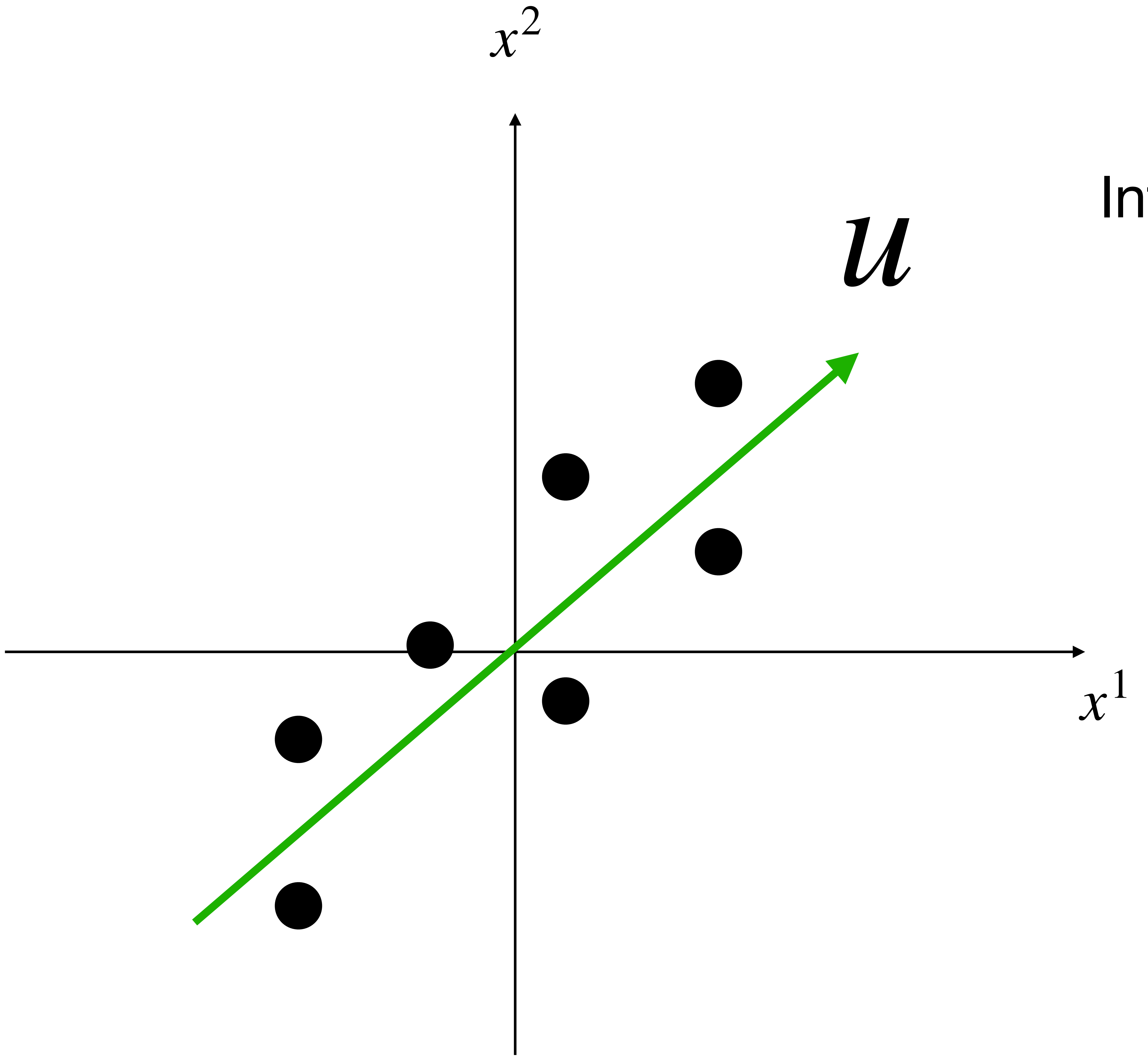
Assume data is centered, i.e.,

$$\sum_{i=1}^{n} x_i / n = 0$$

(Otherwise, compute the mean and shift every data point)

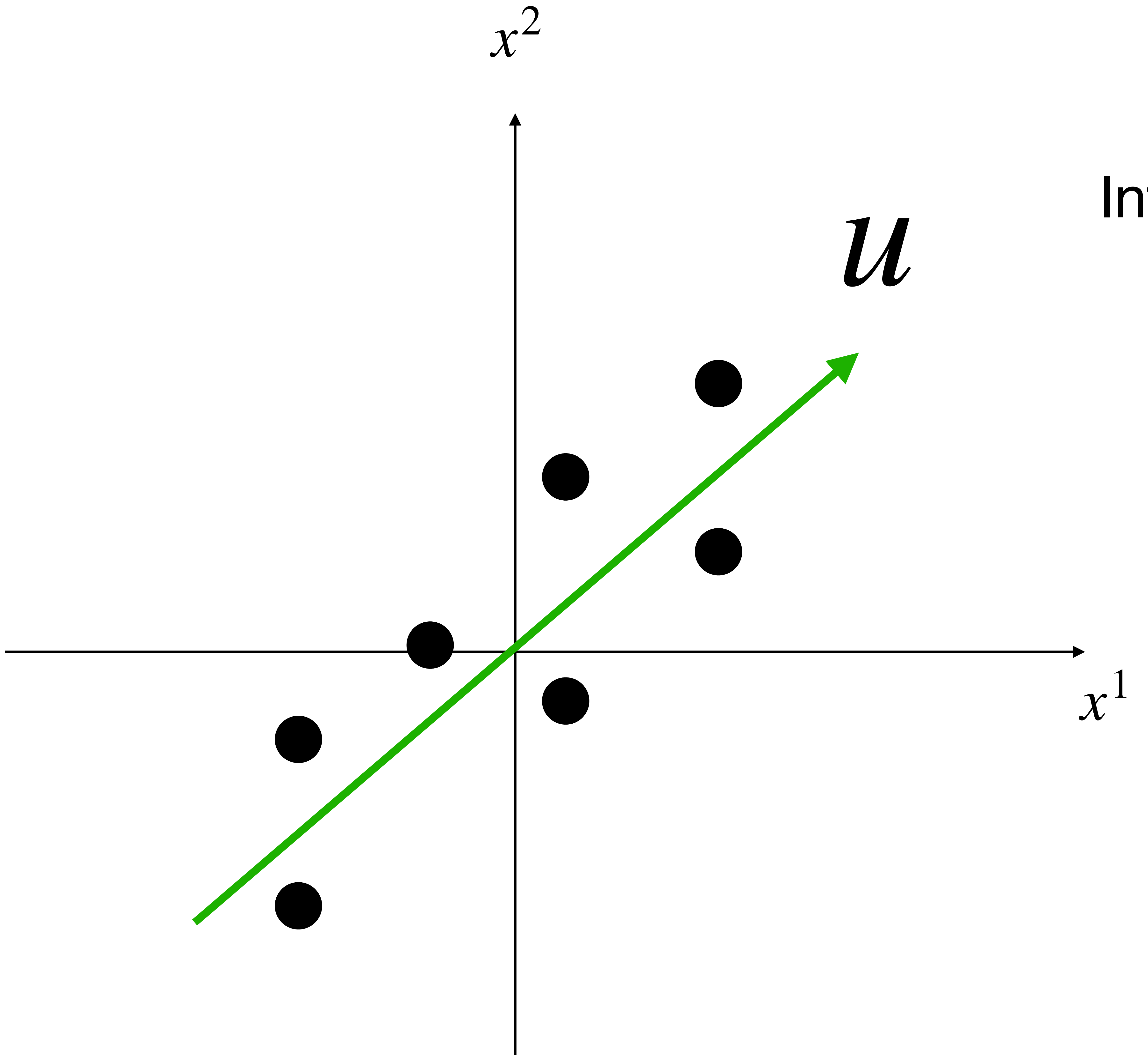
Output: K principle components $u_1, \ldots, u_K$ (they are orthonormal)

Compute the Principal Component



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

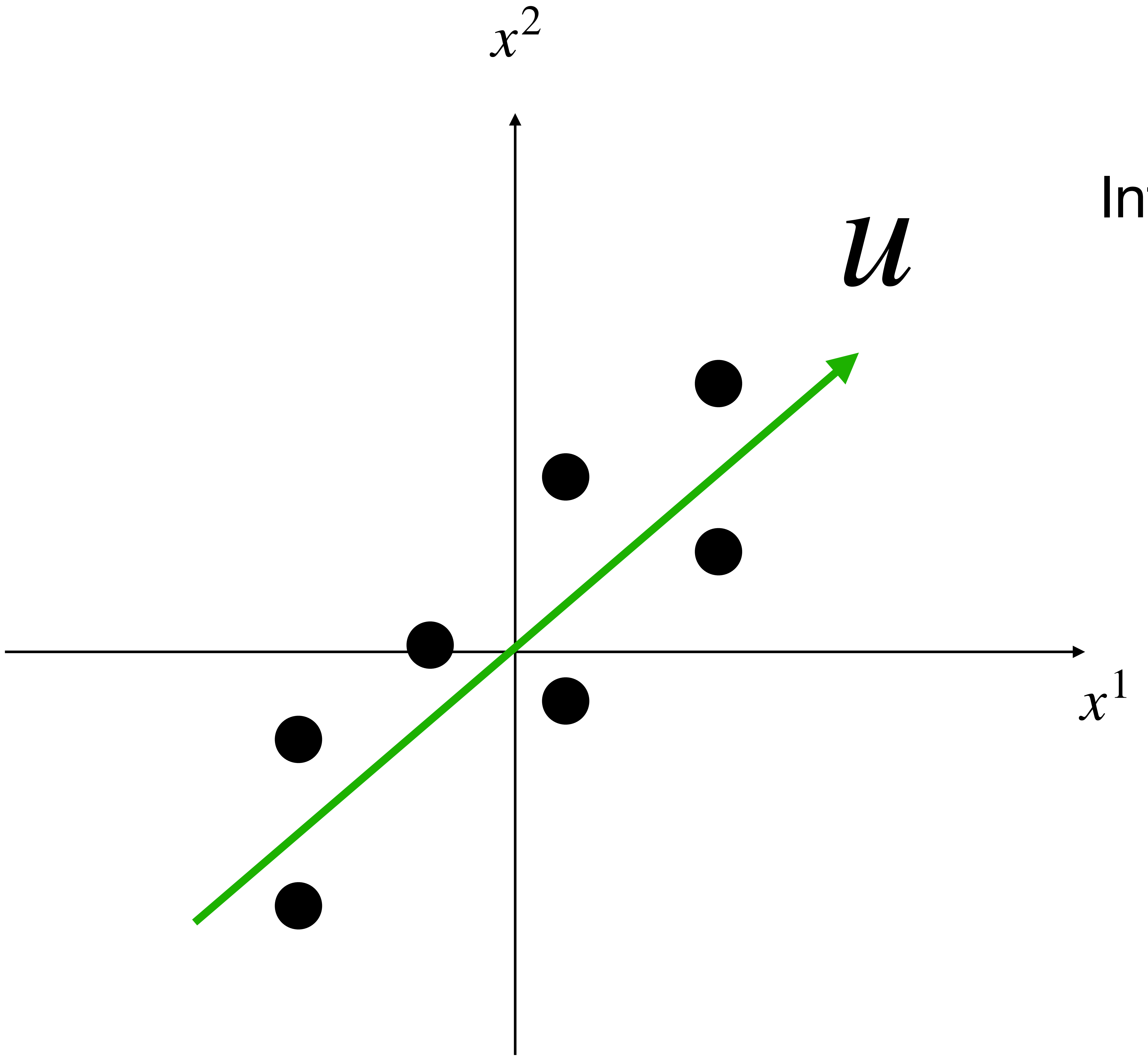
Compute the Principal Component



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

Mathematically, maximizes the variance of projected points

Compute the Principal Component

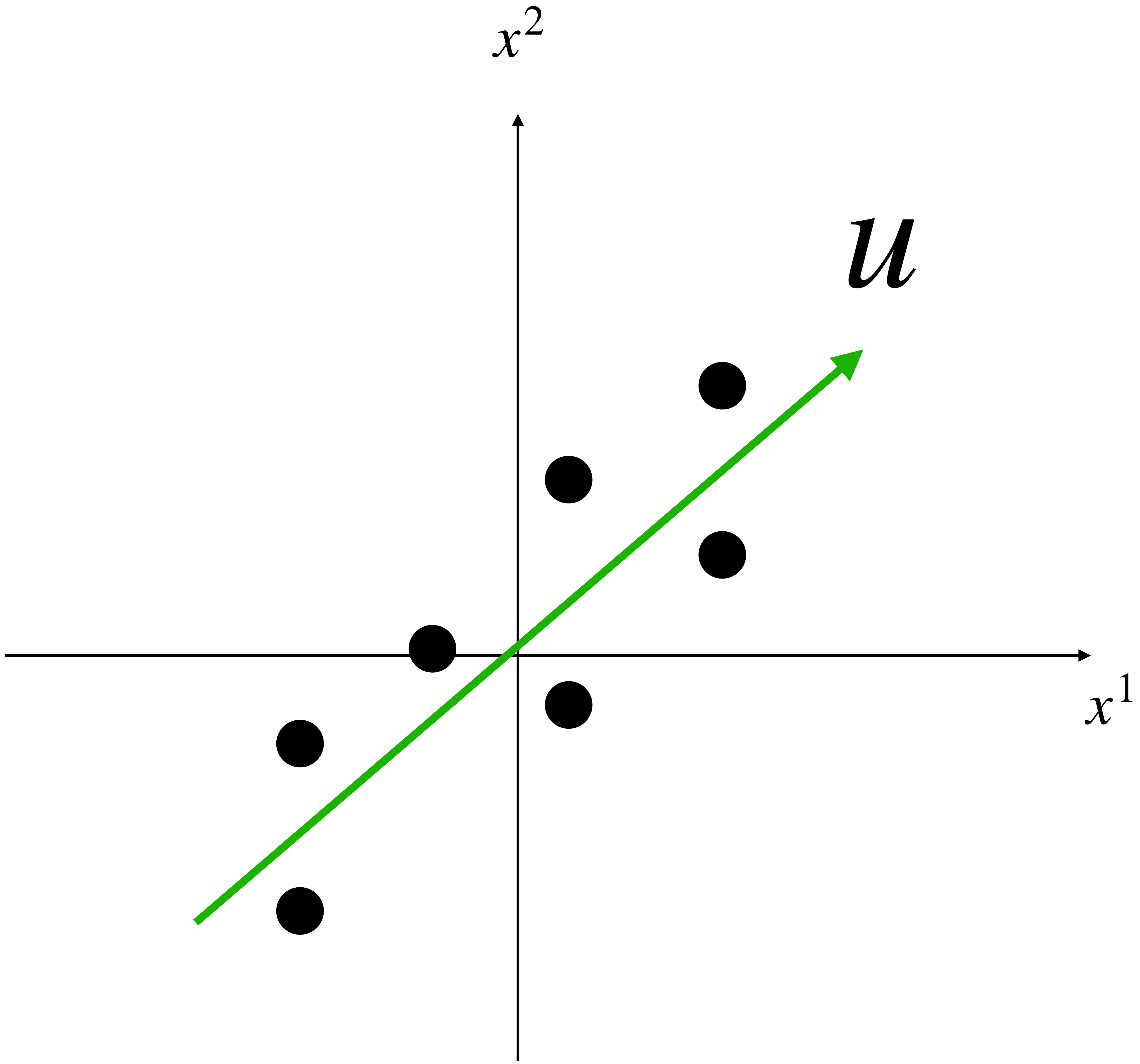


Intuition: find a direction such that the projected points are spread out

Mathematically, maximizes the variance of projected points

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

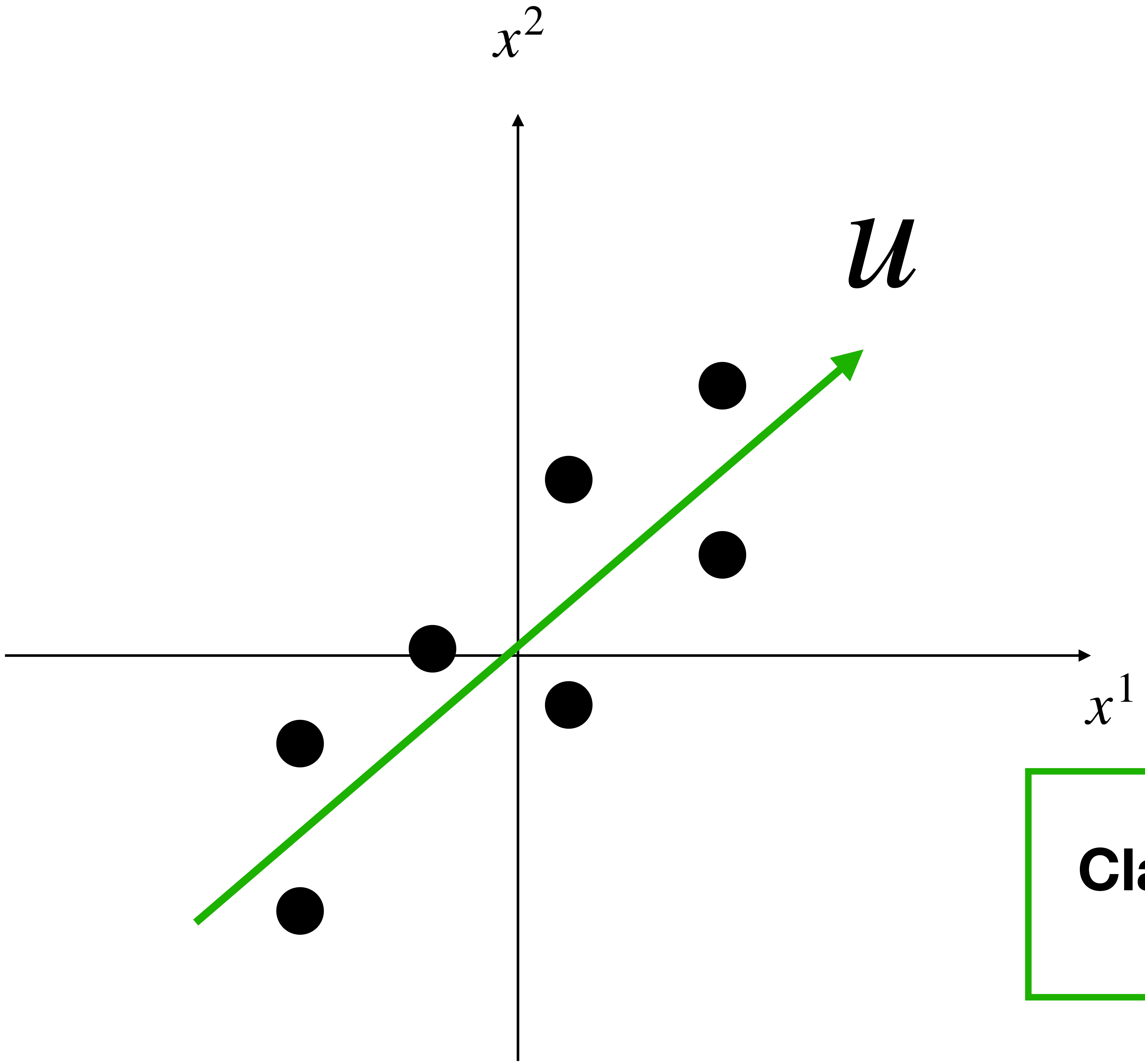
Compute the Principal Component



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

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

Compute the Principal Component



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

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

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 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 if:

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

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

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

Compute the Principal Component

Definition of Eigenvalue/Eigenvectors

(λ, u) is a pair of eigenvalue / eigenvector if:

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

Eigendecomposition:

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

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

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

Compute the Principal Component

Definition of Eigenvalue/Eigenvectors

(λ, u) is a pair of eigenvalue / eigenvector if:

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

Eigendecomposition:

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

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

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

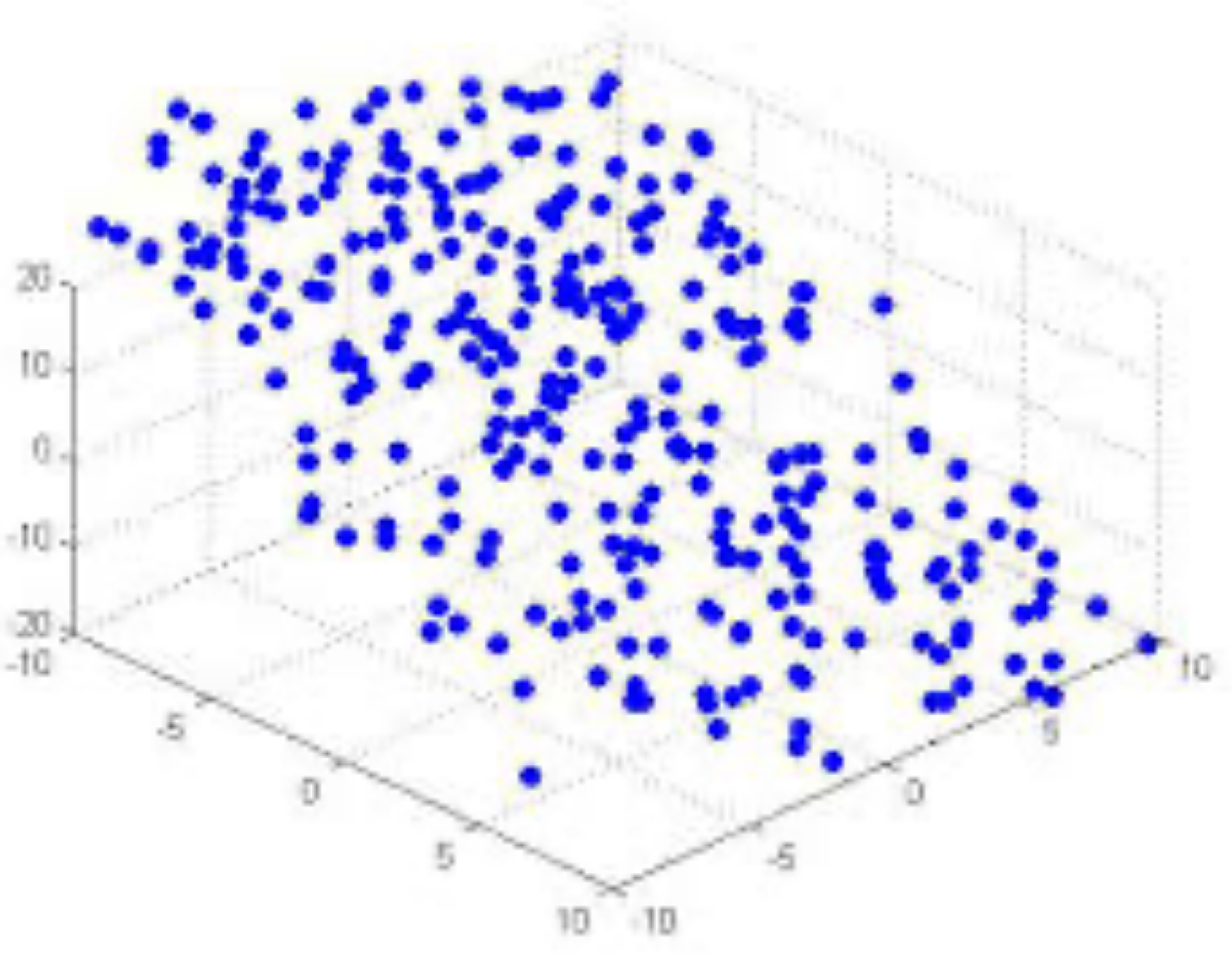
Solution:

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

What about computing the second Principal component?

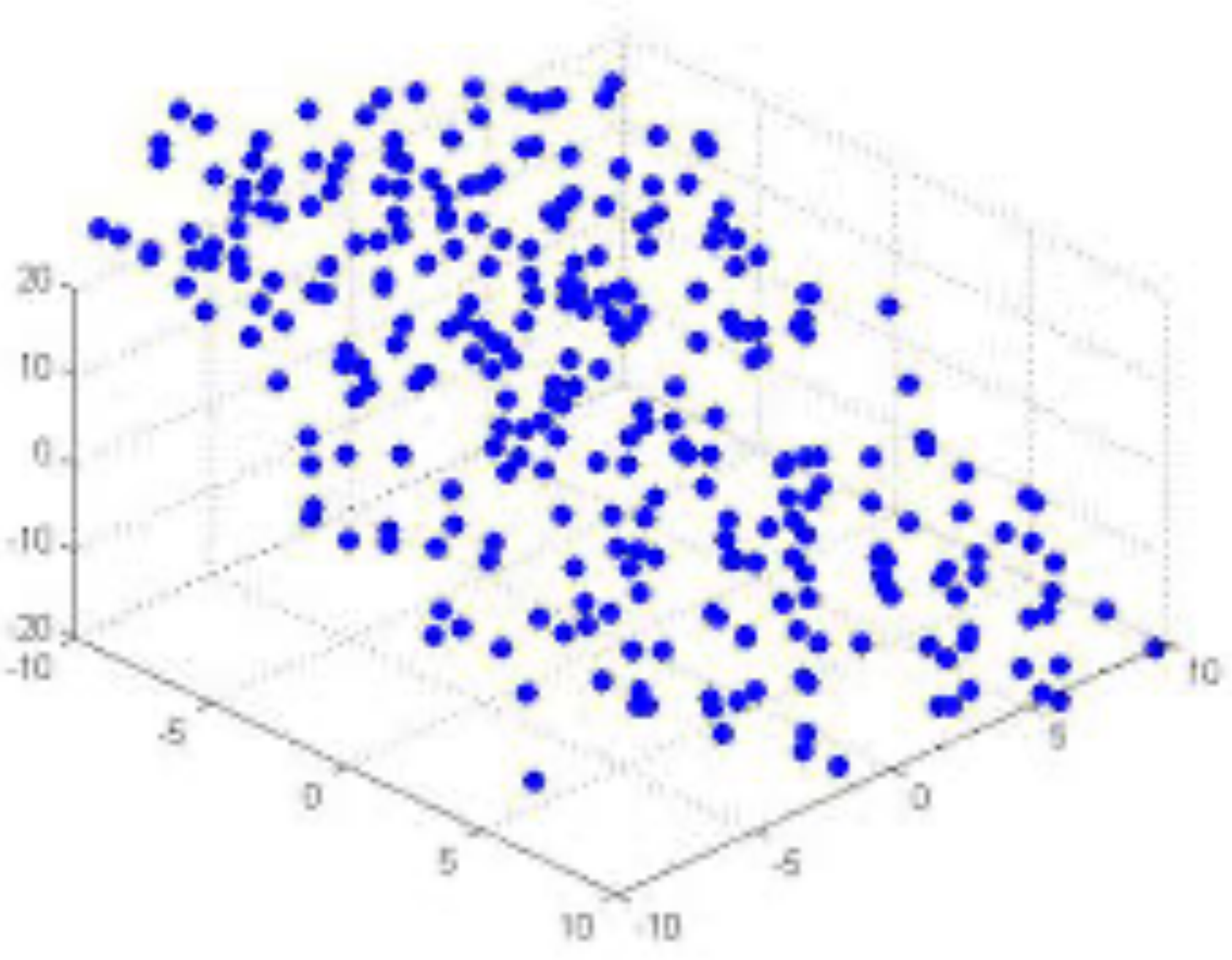
First Principle component $u_1 = \arg\max_{u:\|u\|_2=1} \sum_{i=1}^{n} (x_i^\top u)^2$

To compute the second PC:



What about computing the second Principal component?

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

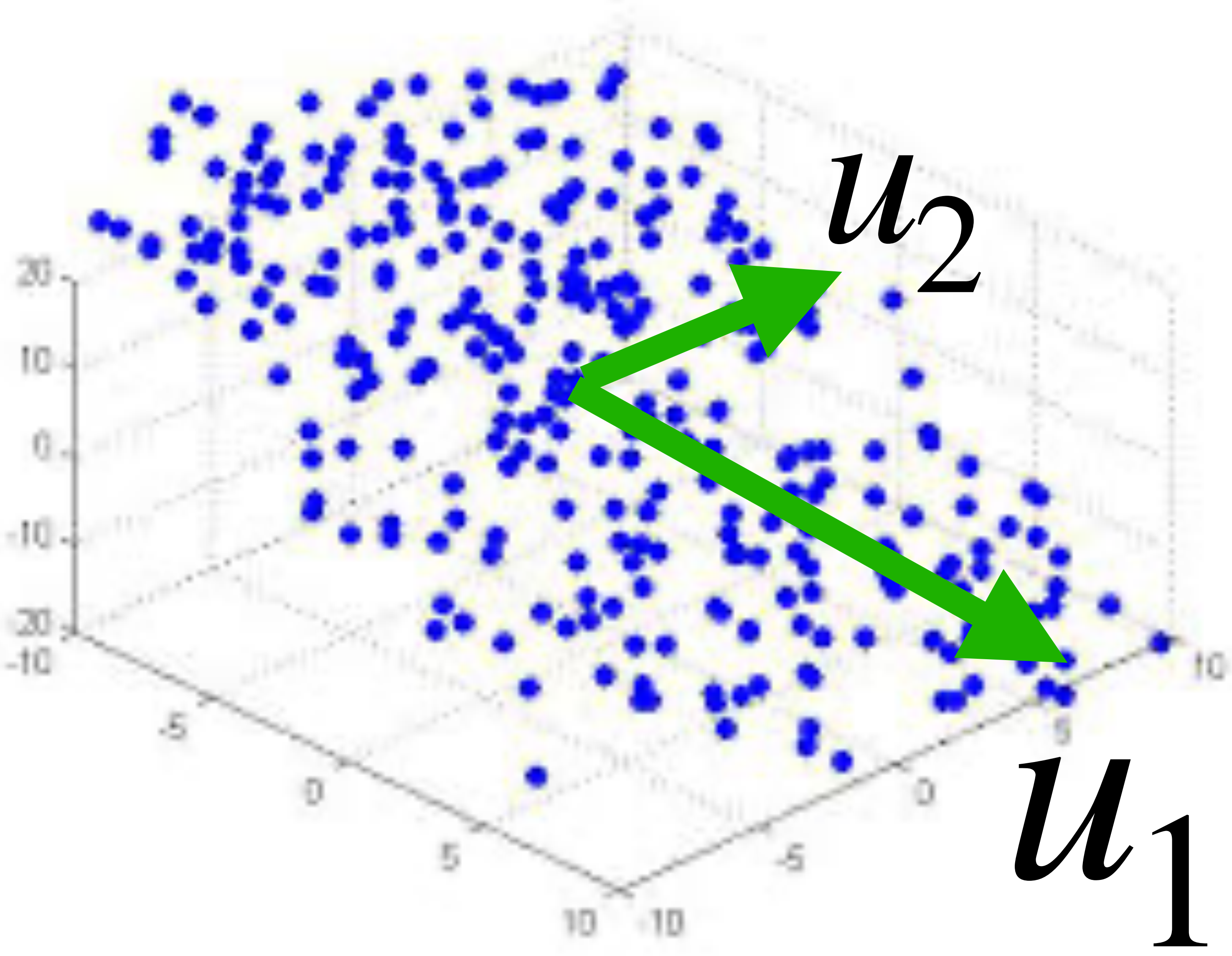


To compute the second PC:

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

What about computing the second Principal component?

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



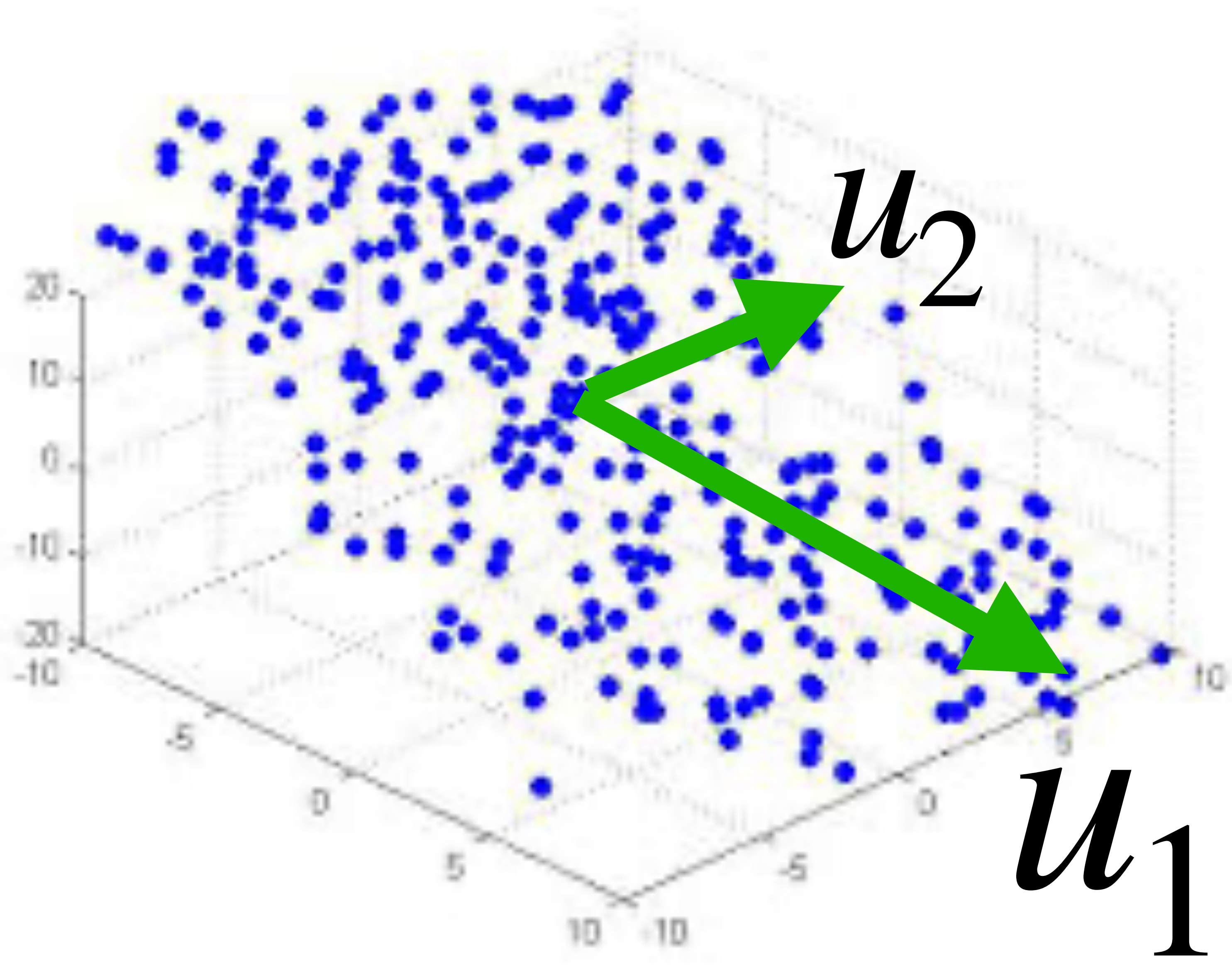
To compute the second PC:

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:

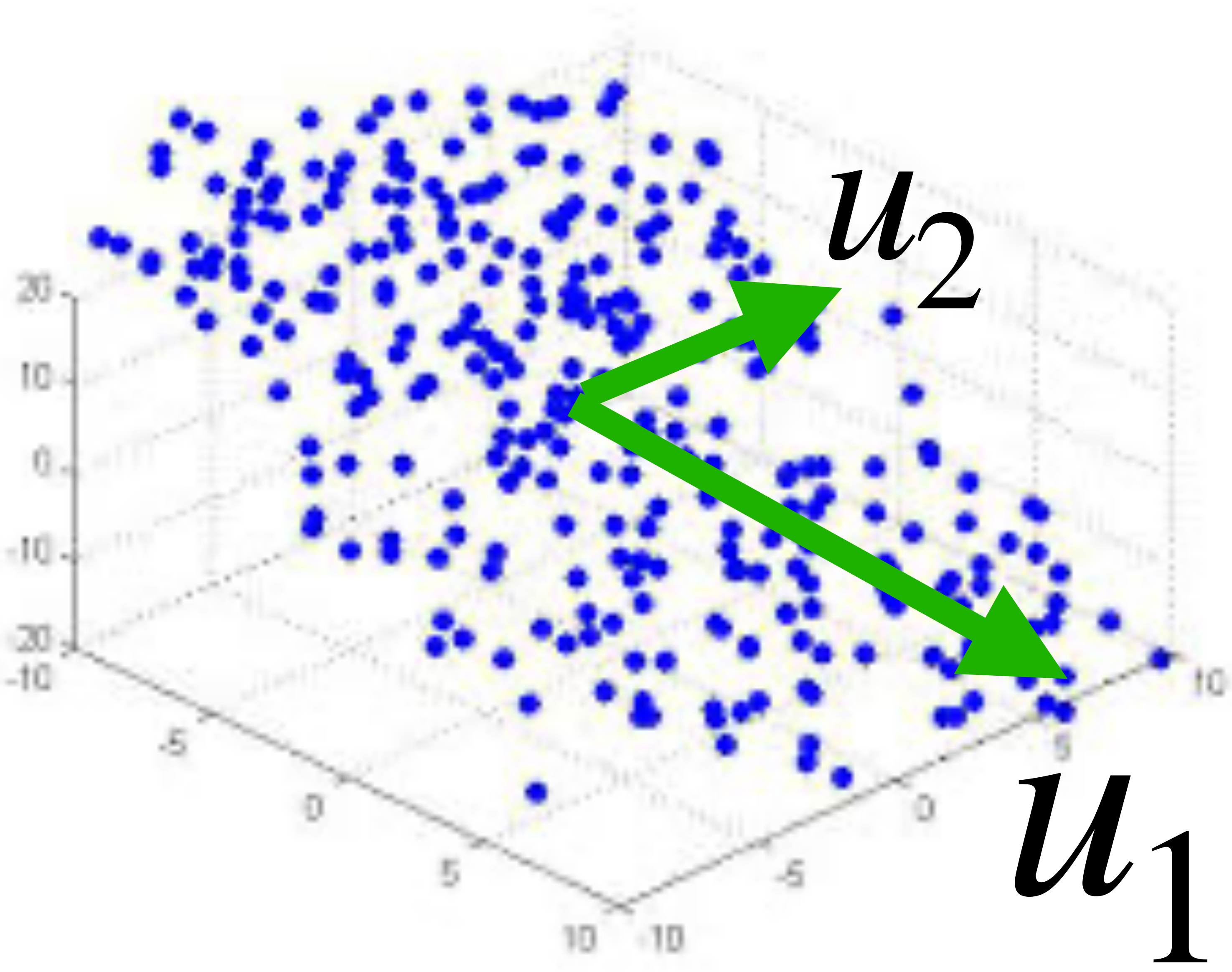


Force constraints: $\|u_2\|_2 = 1,\ u_2^\top u_1 = 0$

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

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:

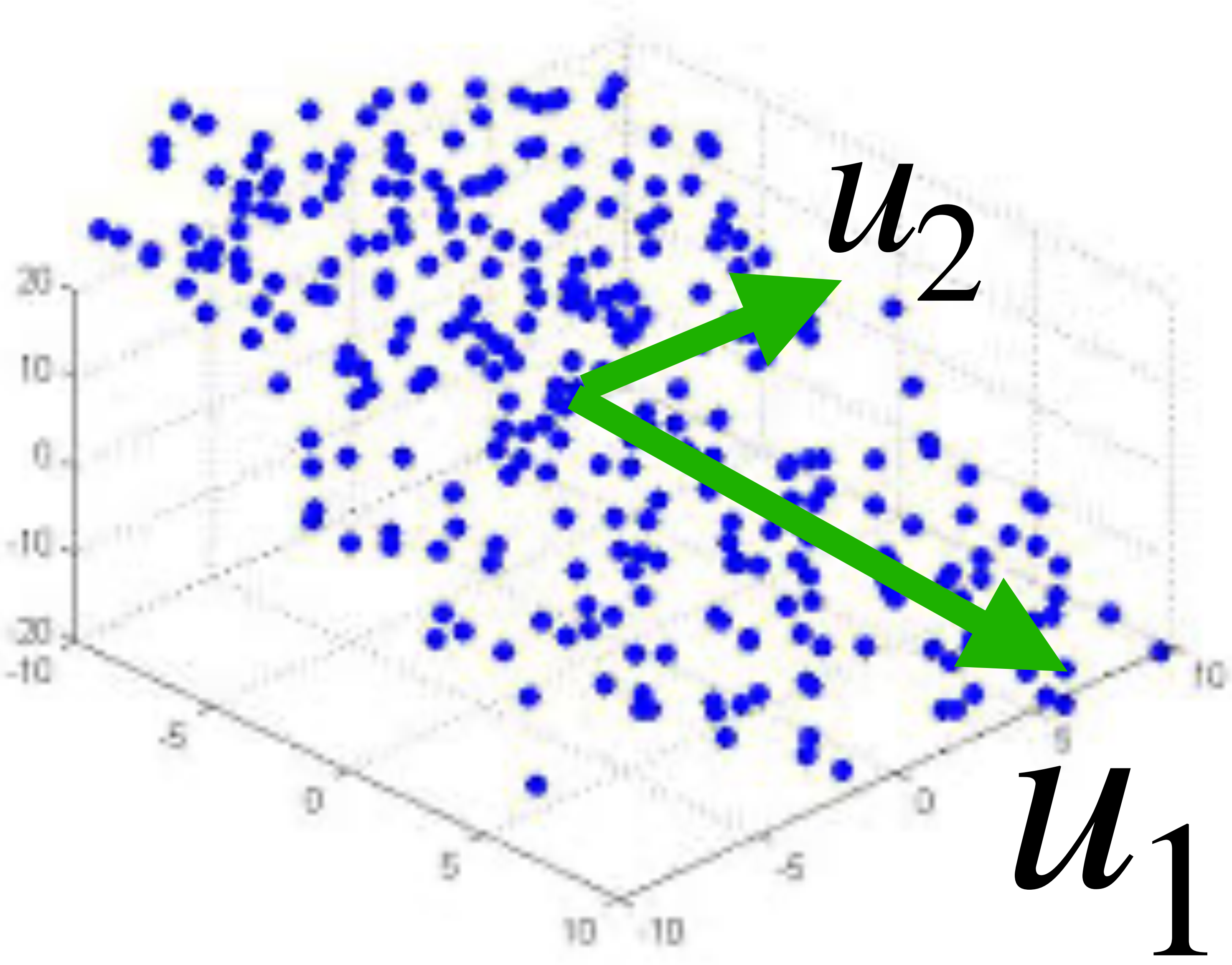
Force constraints: $\|u_2\|_2 = 1,\ u_2^\top u_1 = 0$

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

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

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 centered dateset $\mathscr{D} = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^d$, and parameter $K < d$

Algorithm: PCA

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

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

Algorithm: PCA

Input: given the centered 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)

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

Algorithm for data compression via PCA

Input: the centered dateset $\mathscr{D} = \{x_1, \ldots, x_n\}$, parameter K

Algorithm for data compression via PCA

Input: the centered dateset $\mathscr{D} = \{x_1, \ldots, x_n\}$, parameter K

1. $u_1, u_2, \ldots, u_K$ = PCA$(\{x_1, \ldots, x_n\}, K)$

Algorithm for data compression via PCA

Input: the centered dateset $\mathscr{D} = \{x_1, \ldots, x_n\}$, parameter K

1. $u_1, u_2, \ldots, u_K$ = PCA$(\{x_1, \ldots, x_n\}, K)$

2. $\forall x \in \mathscr{D}$, compute $z = [u_1^\top x, u_2^\top x, \ldots, u_K^\top x]^\top \in \mathbb{R}^K$

Algorithm for data compression via PCA

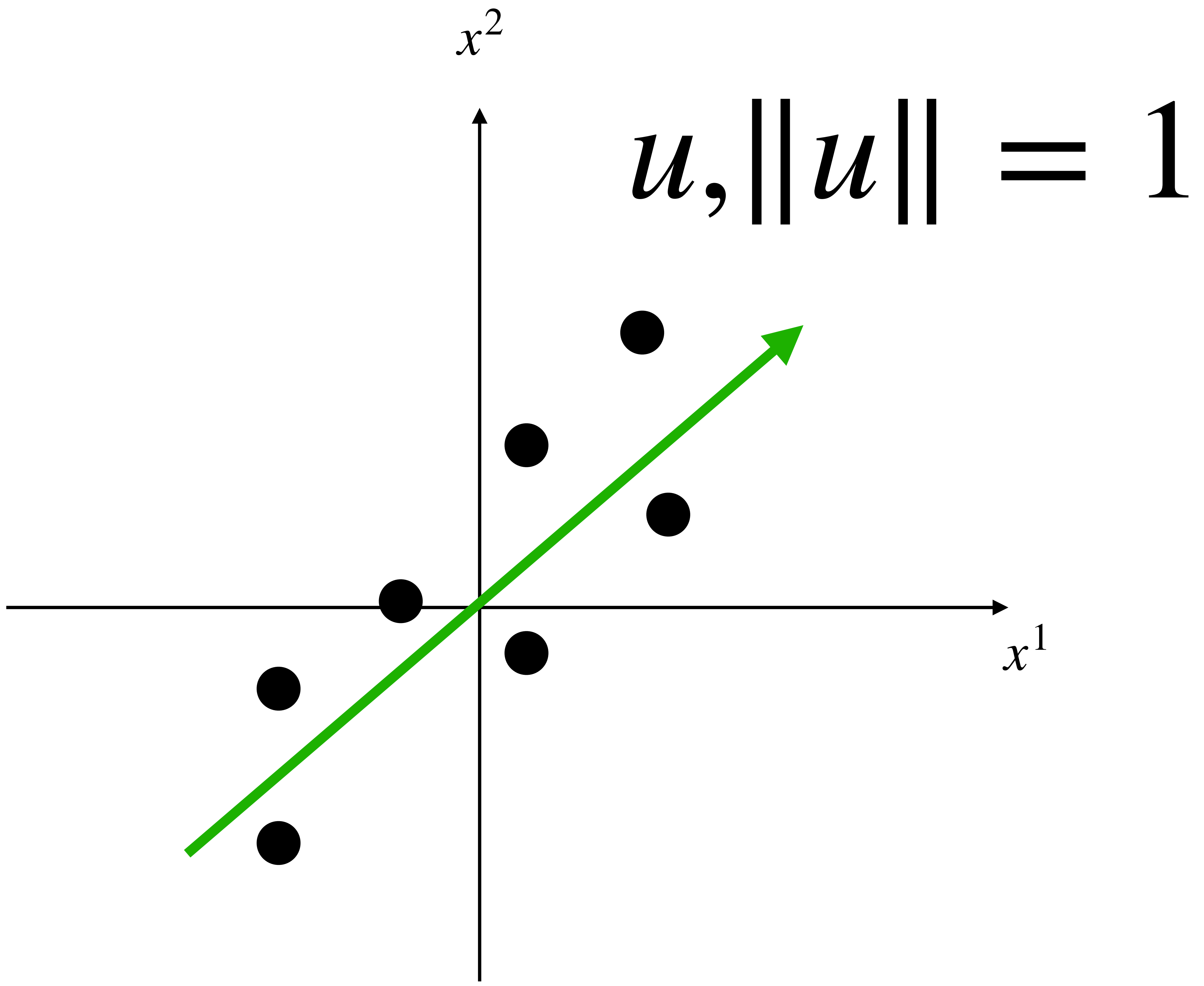
Input: the centered dateset $\mathscr{D} = \{x_1, \ldots, x_n\}$, parameter K

1. $u_1, u_2, \ldots, u_K$ = PCA($\{x_1, \ldots, x_n\}, K$)

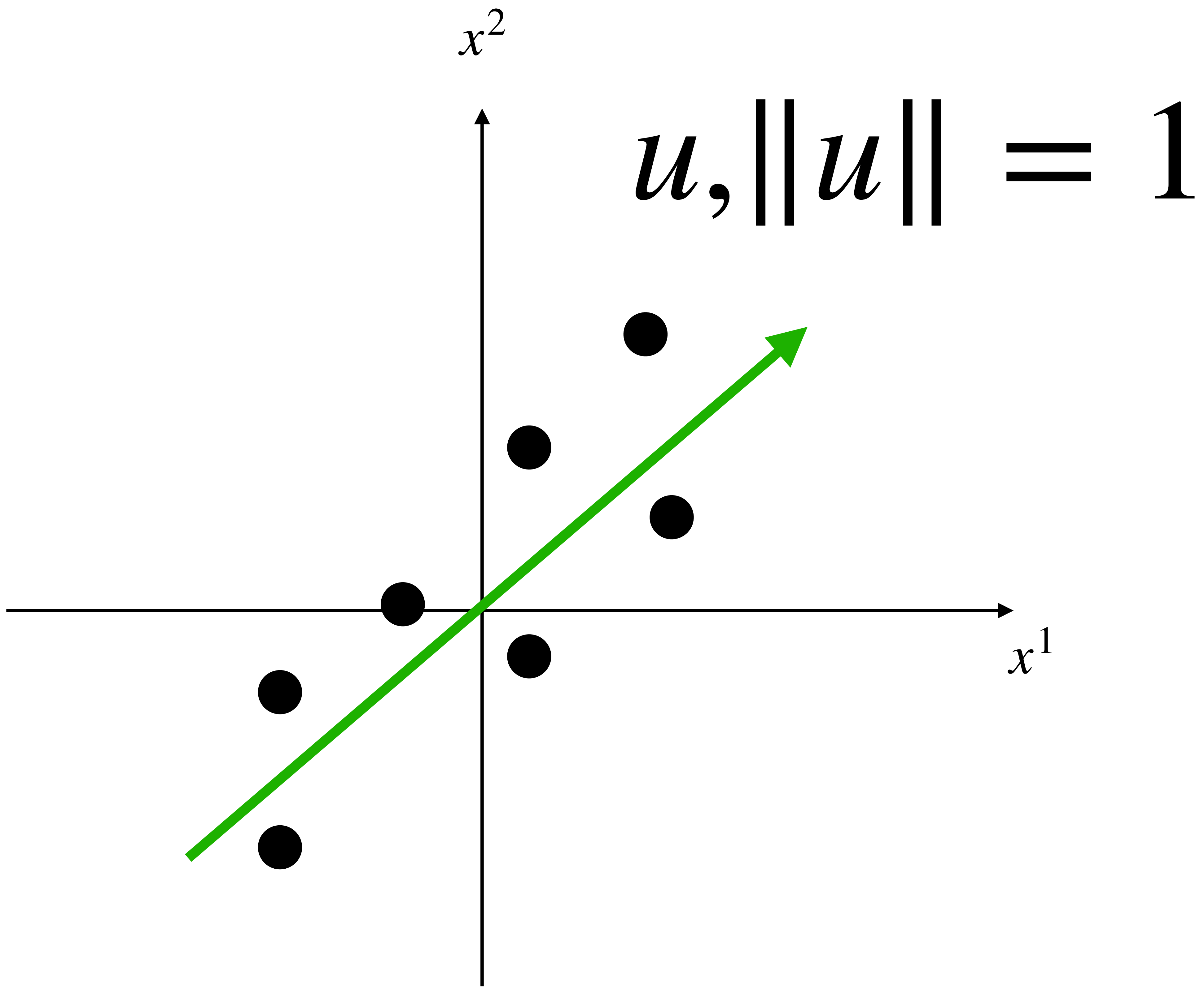
2. $\forall x \in \mathscr{D}$, compute $z = [u_1^\top x, u_2^\top x, \ldots, u_K^\top x]^\top \in \mathbb{R}^K$

Output: K-dim dataset $\mathscr{Z} = \{z_1, \ldots, z_n\}, z_i \in \mathbb{R}^K$

Think about PCA from a data re-construction perspective



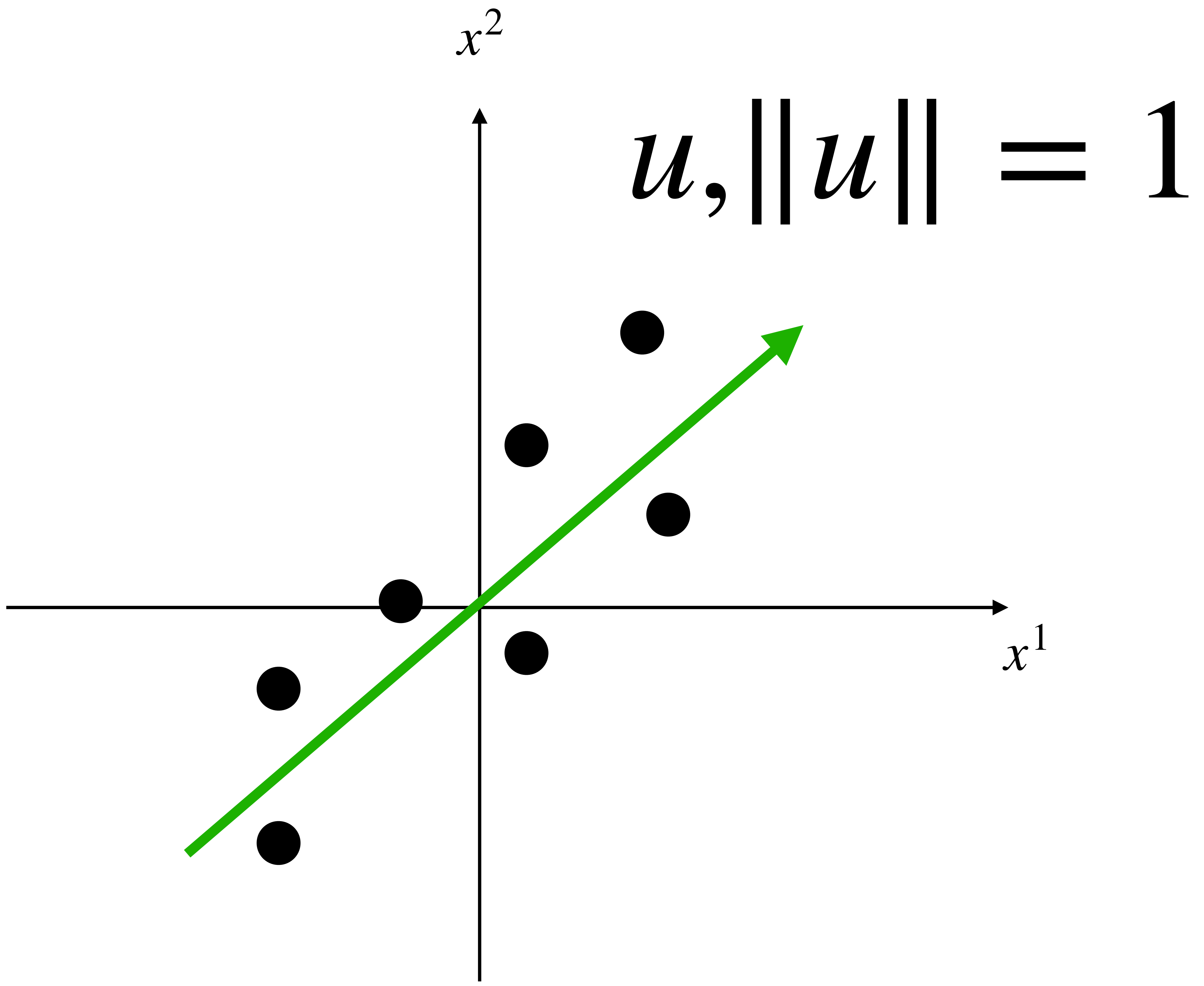
Think about PCA from a data re-construction perspective



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

(i.e., project x on u)

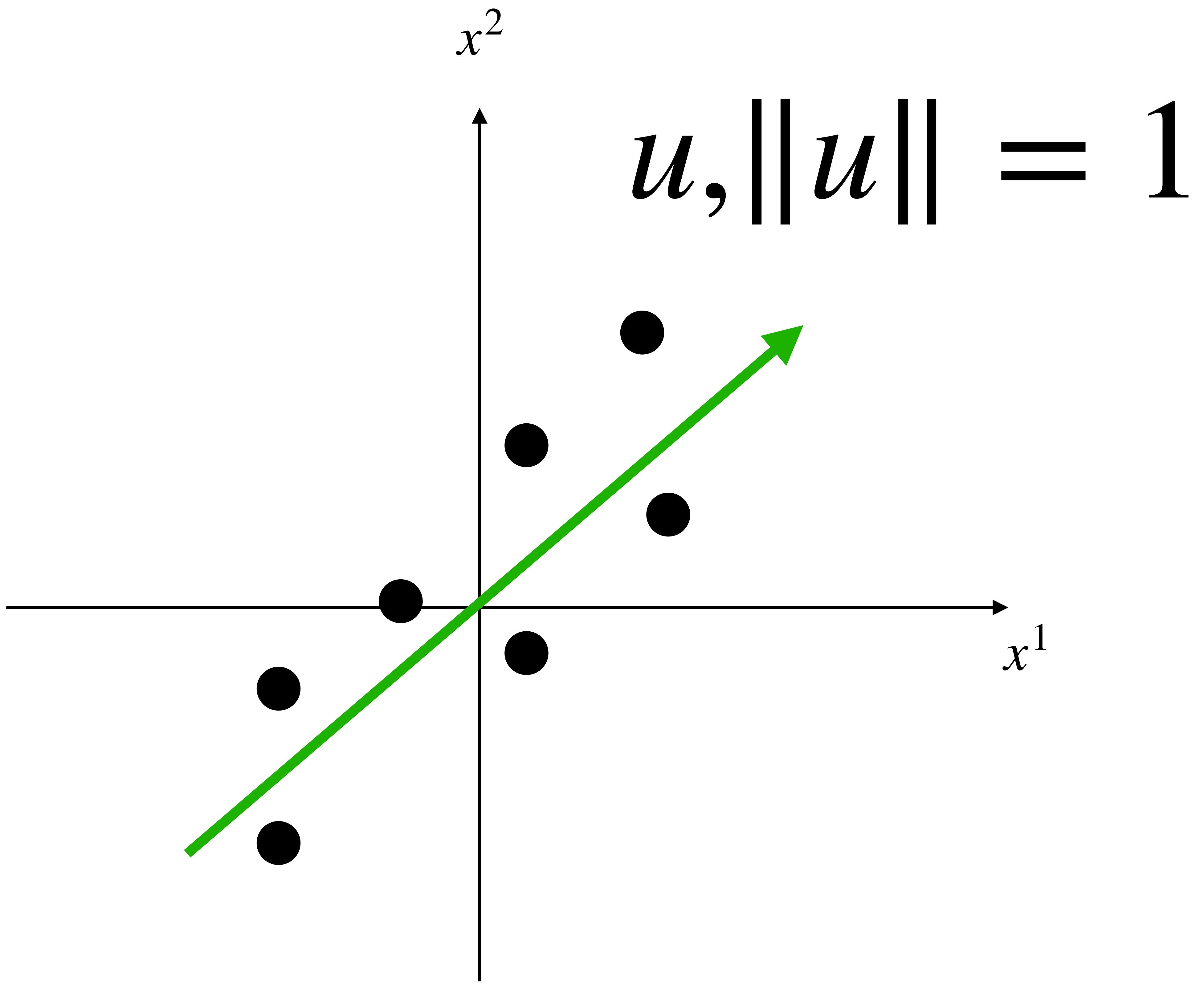
Think about PCA from a data re-construction perspective



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



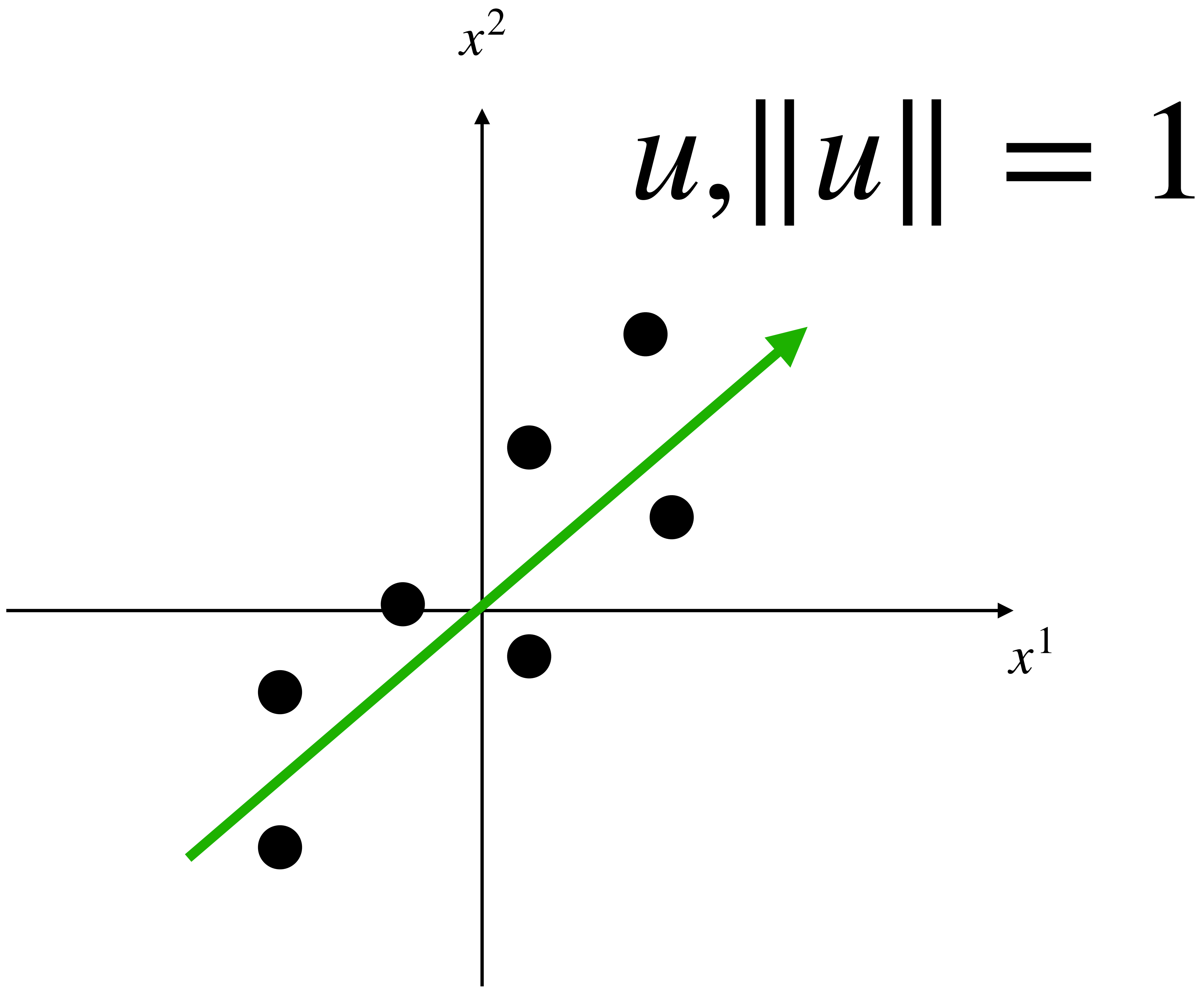
Represent x using u: $x \rightarrow (x^\top u)u$

(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



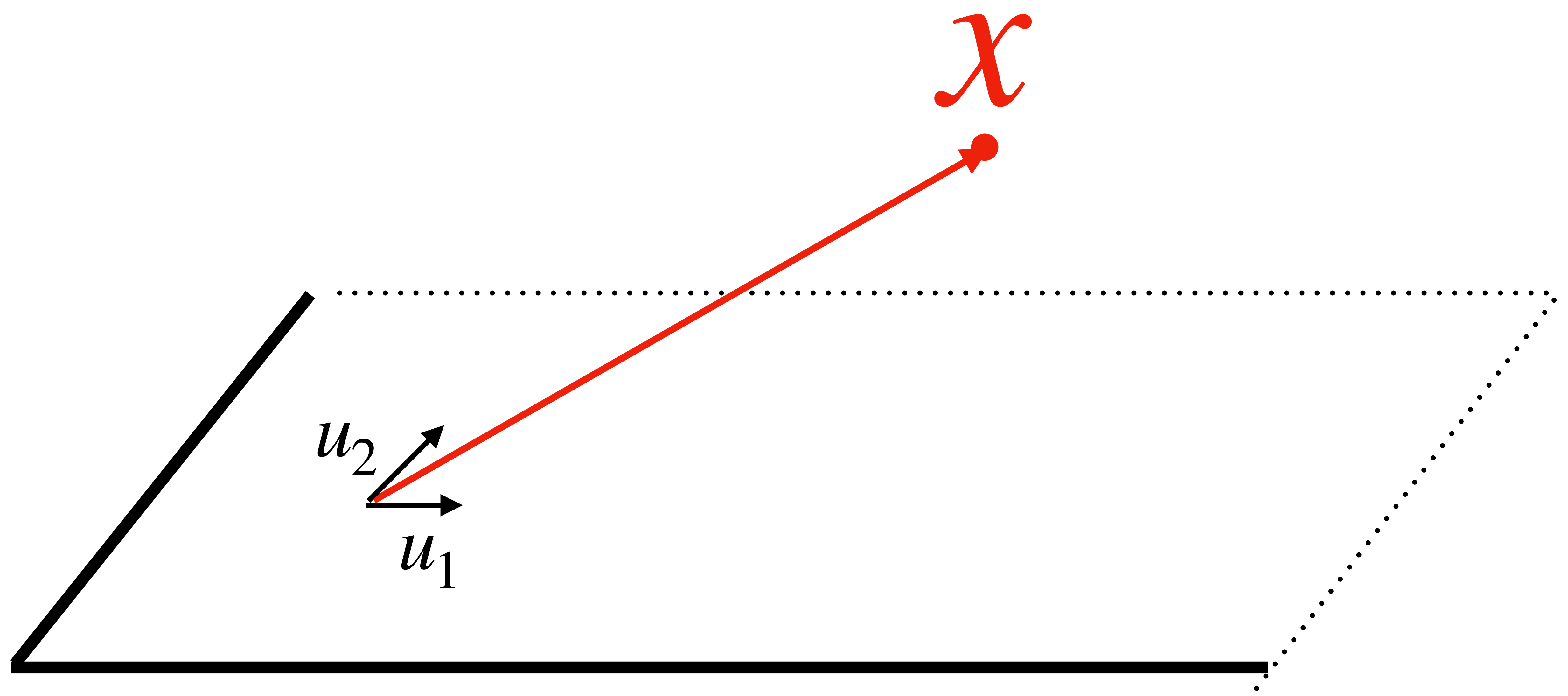
Represent x using u: $x \rightarrow (x^\top u)u$
(i.e., project x on u)

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

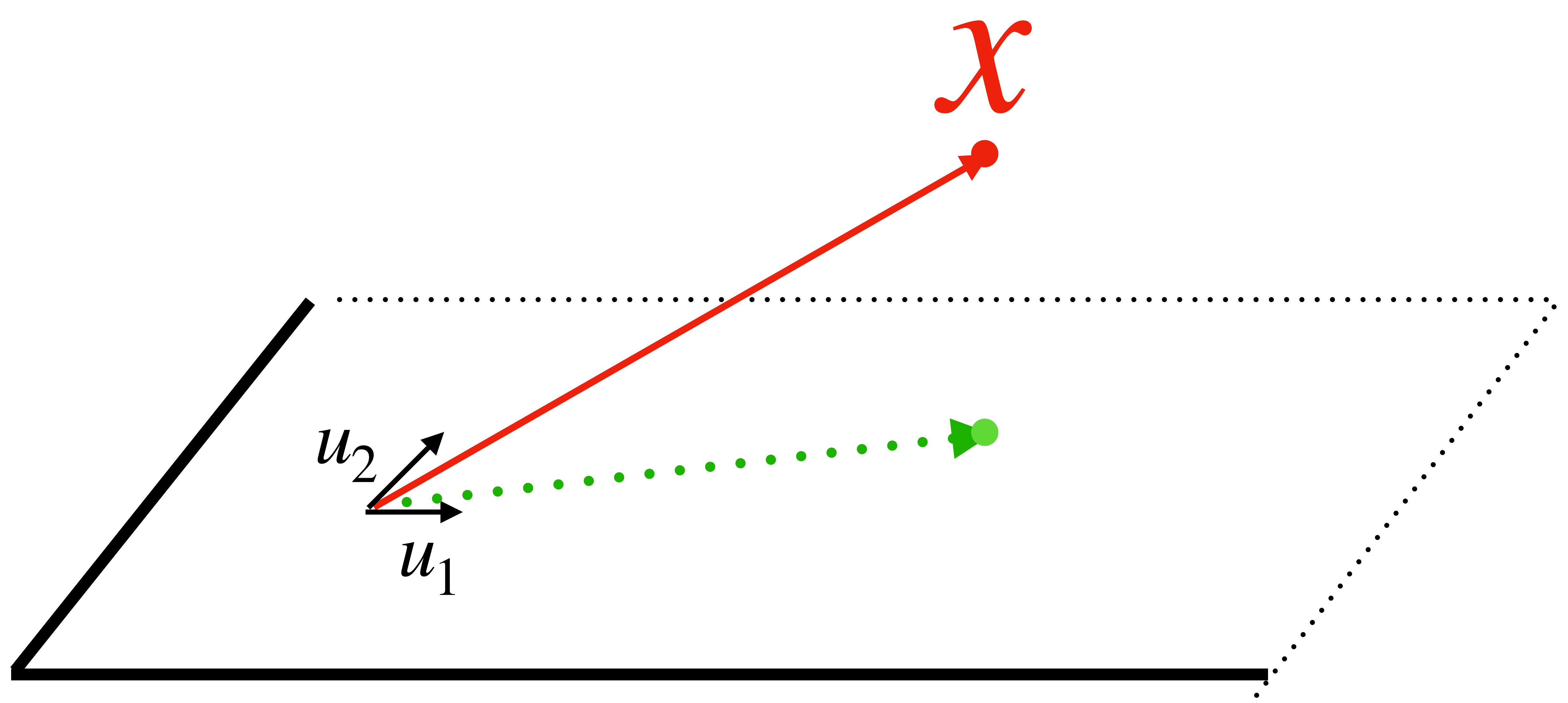
PCA first principle component procedure : find u that minimizes the total reconstruction error

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

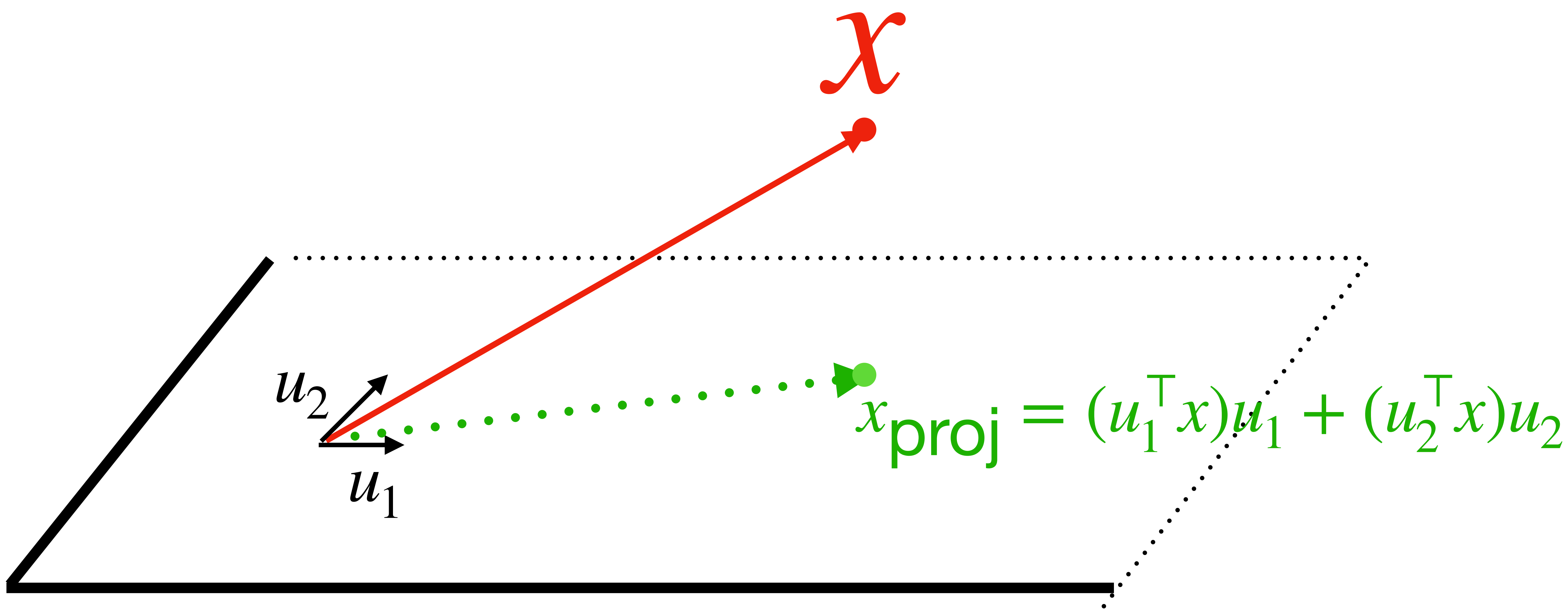
Think about PCA from a data re-construction perspective



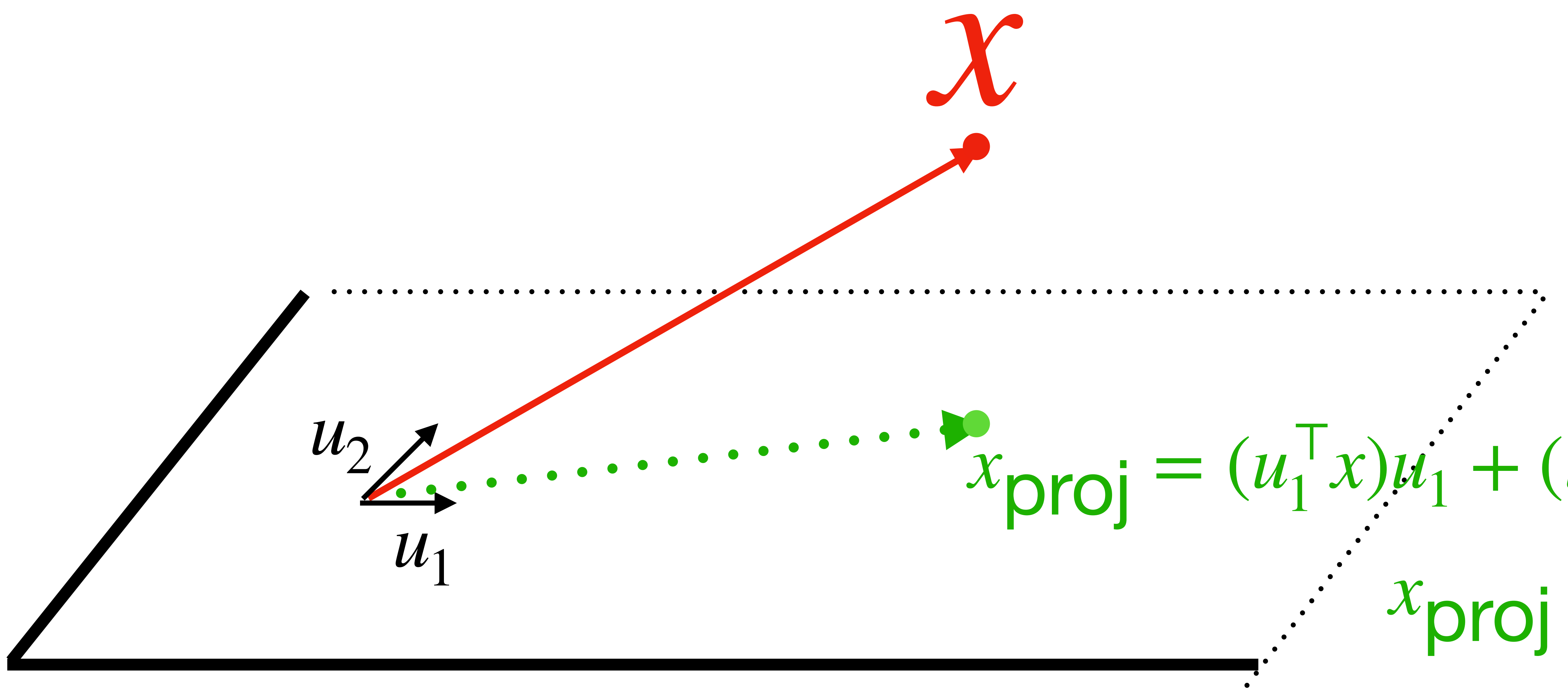
Think about PCA from a data re-construction perspective



Think about PCA from a data re-construction perspective



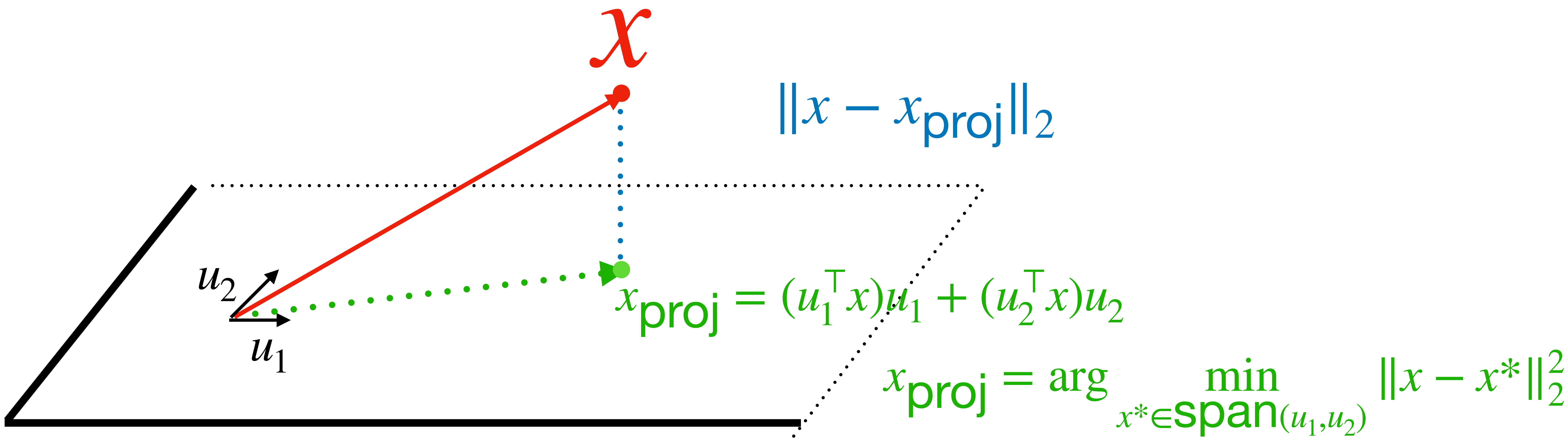
Think about PCA from a data re-construction perspective



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

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

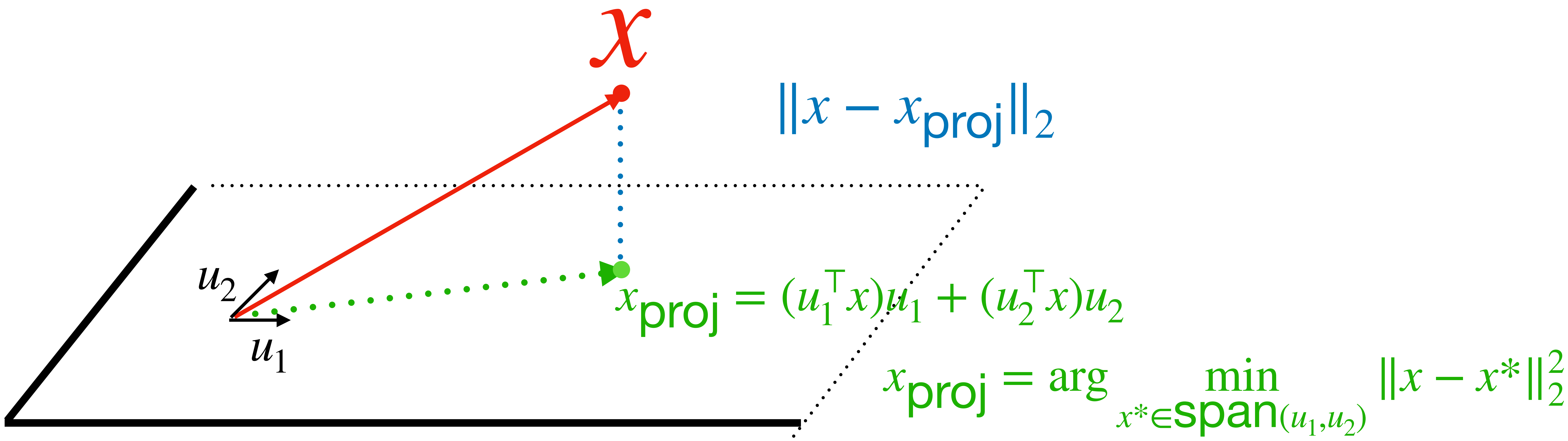
Think about PCA from a data re-construction perspective



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

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



Outline for today:

1. Intro of PCA

2. PCA via eigendecomposition

3. Example of PCA: eigenfaces

Application of PCA: Eigenfaces

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

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:

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

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:

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

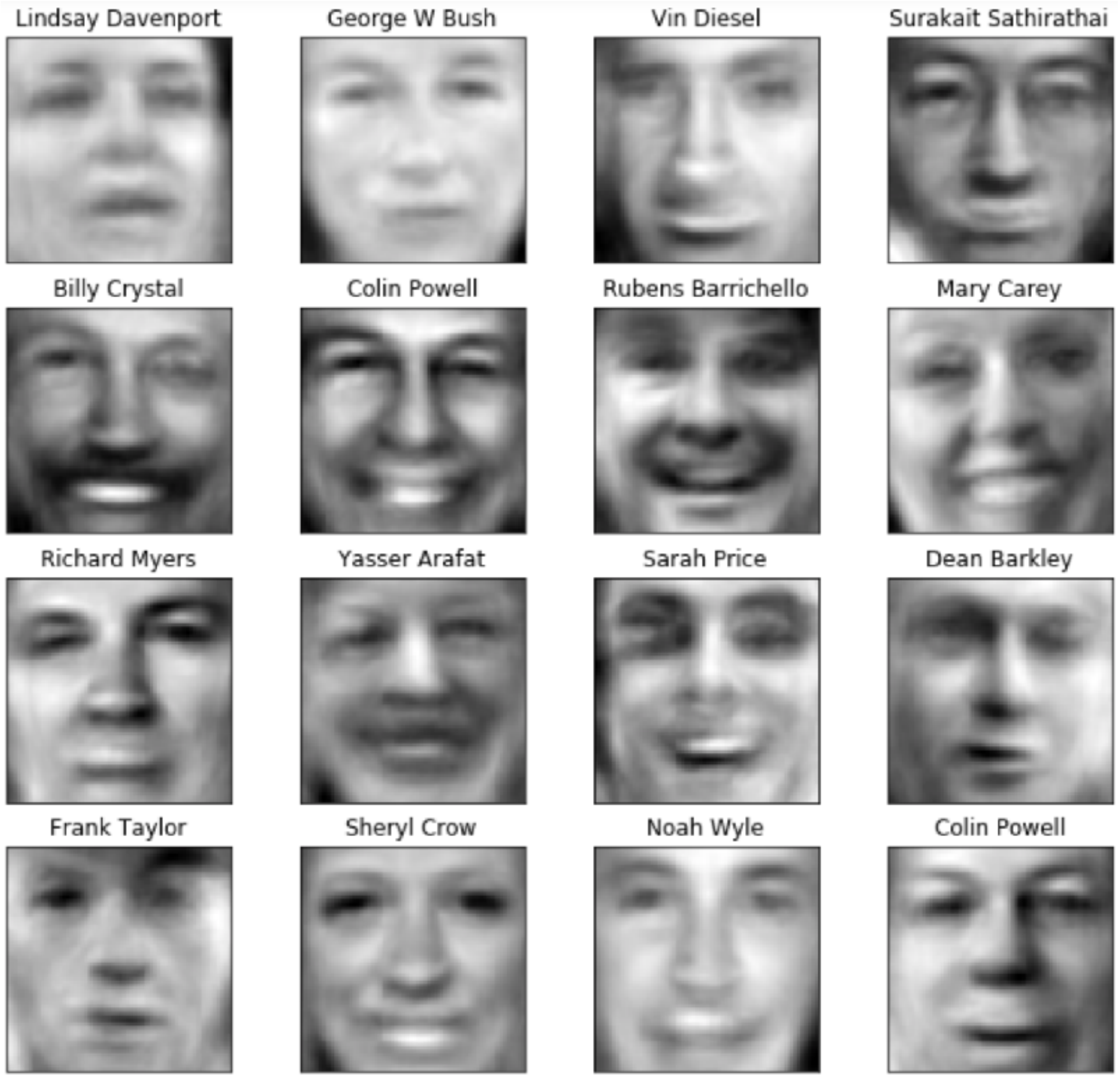
(Q: when $k \to 64^2$, we should expect $x' \to x$, why?)

Recall that PCA is about finding $u_1, u_2, \ldots, u_k$ to minimize re-construction error

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

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