Dimensionality Reduction

Lecture 12

The Curse of Dimensionality

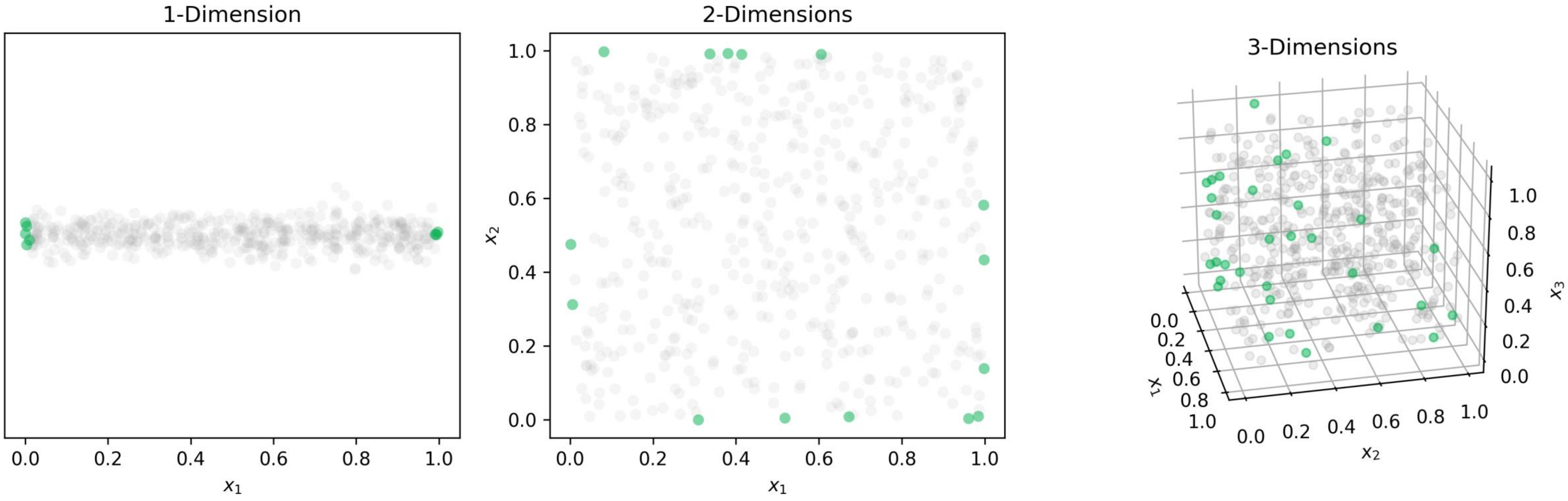
Challenge 1

In high dimensions, data become sparse

(increasing the risk of overfitting)

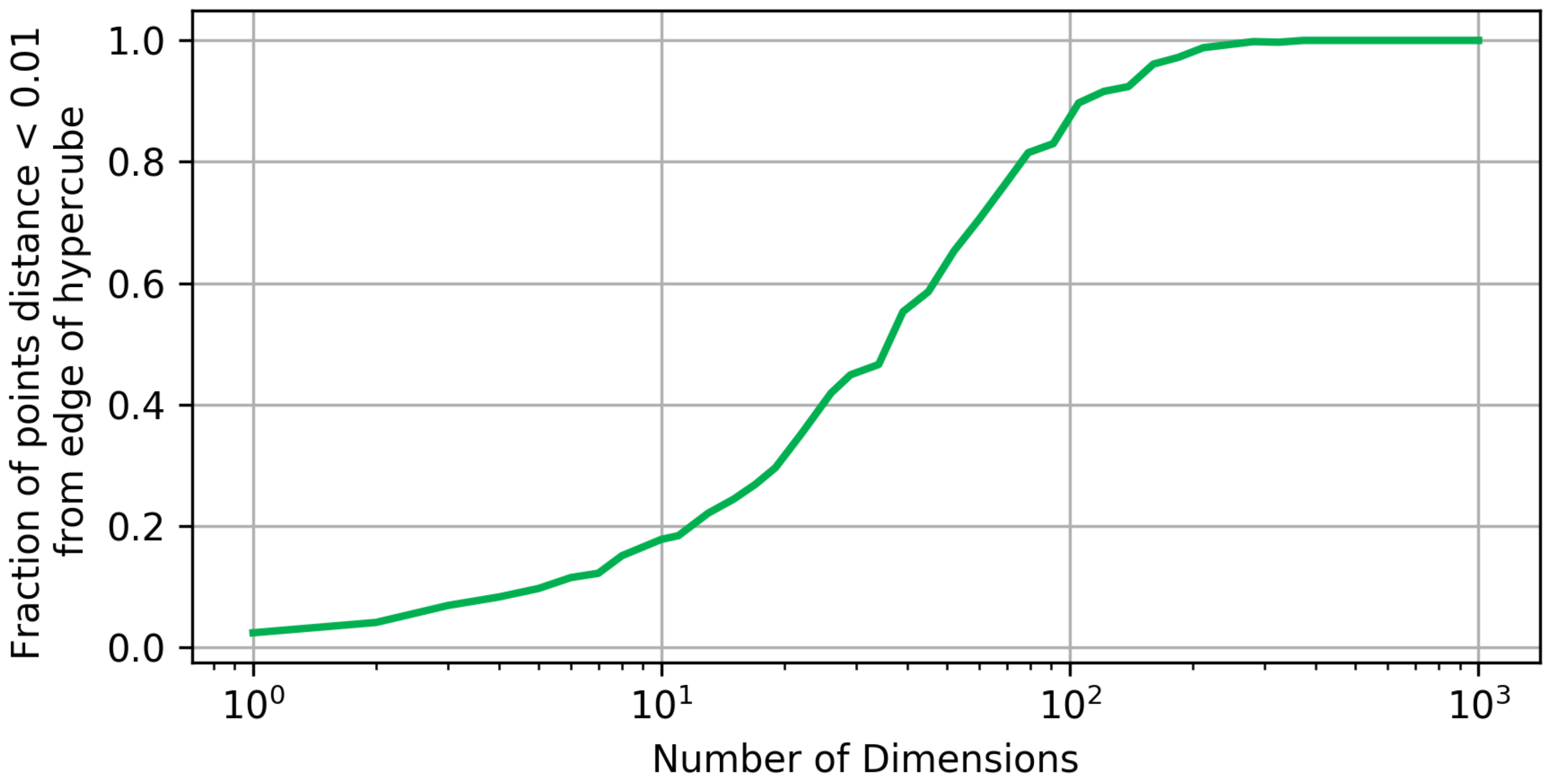
Random data points in a unit hypercube…

- Data point is a distance < 0.01 units from the edge of a unit hypercube
- All other data



Fraction of edge data = 0.016 0.030 0.064

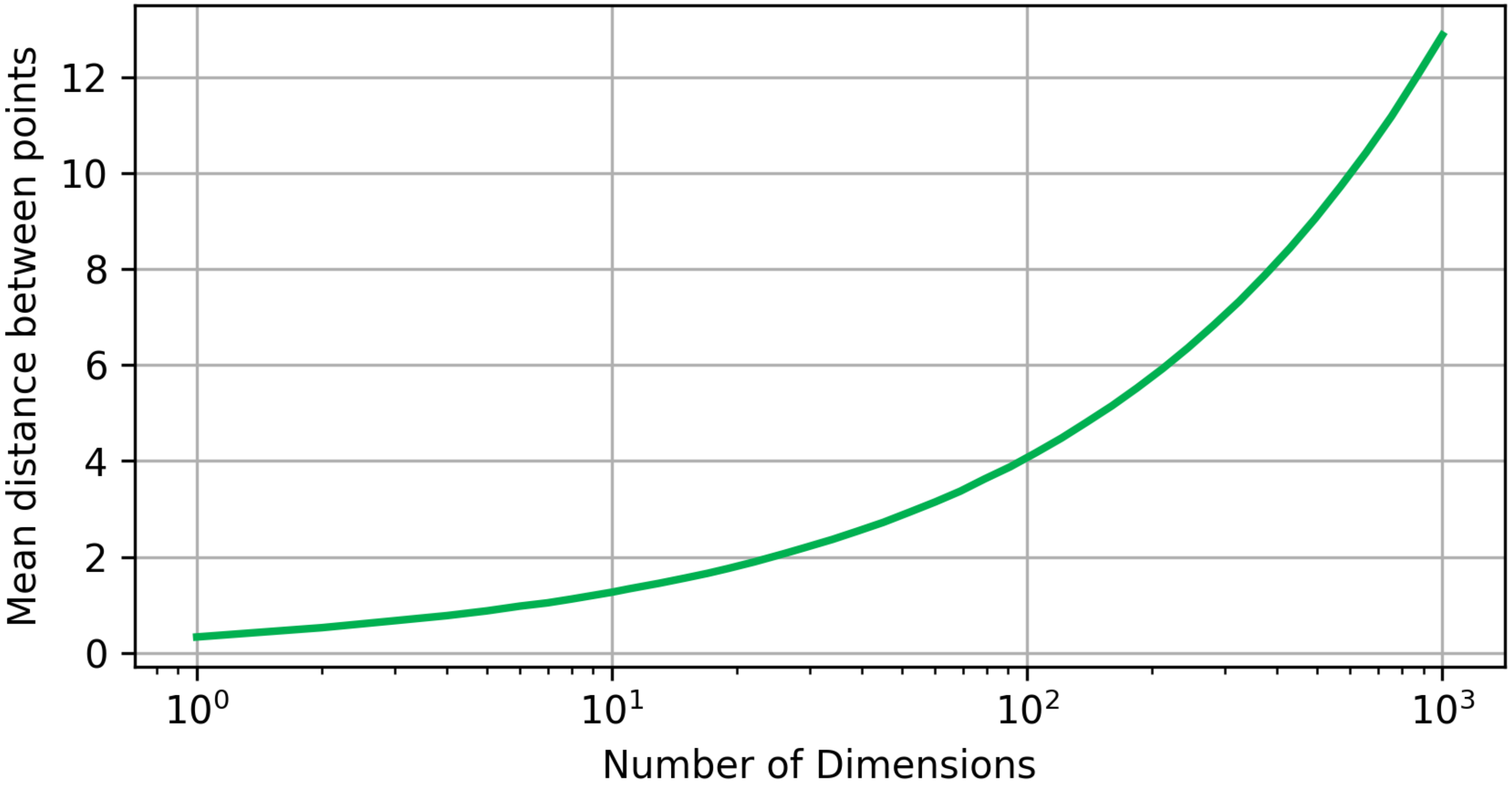
In high dimensions…



…nearly all of the high dimensional space is **far away from the center**

Note: figures constructed using 1,000 random points

In high dimensions…



…data become sparse

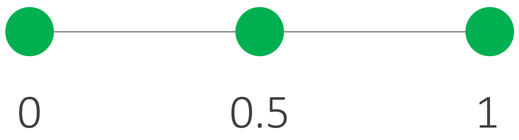
Note: figures constructed using 1,000 random points

Challenge 2

Much more data are needed for sampling higher dimensional spaces

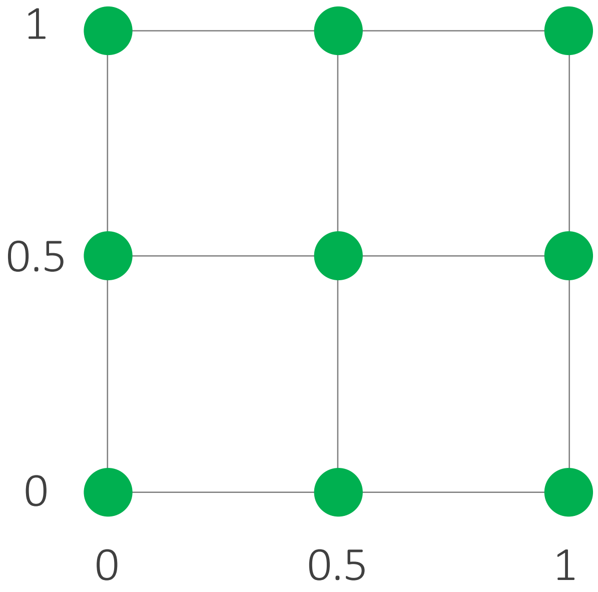
Sample a unit hypercube on a grid spaced at intervals of 0.5

D = 1



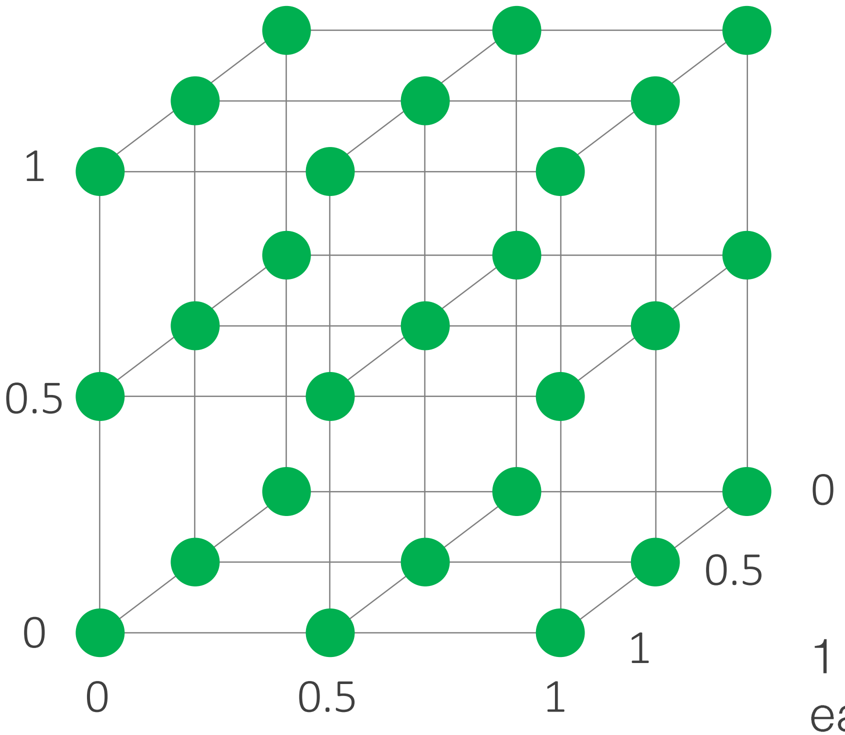
N = 3

D = 2



N = 9

D = 3



N = 27

Dim (D)	**Samples (N)**
4	81
5	243
10	59,049
100	5.2×10^{47}
300	$\mathbf{1.4 \times 10^{143}}$

1 million Googles, each with 20 exabytes of data would only be 10^{25} bytes

…it takes more data to learn in high dimensional spaces

Dimensionality Reduction

Benefits:

Simplified computation
Reduced redundancy of features
Improved numerical stability due to removed correlations

Popular approach:

Principal Components Analysis (PCA)

PCA

Before you begin: Normalize the data!

For each feature, subtract the mean and divide by the standard deviation

$$\underbrace{x}_{\text{Normalized feature}} = \frac{\overbrace{x'}^{\text{Raw, unscaled feature}} - \overline{x'}}{\sigma_{x'}}$$

$$\boldsymbol{X} = \begin{array}{c} \text{columns = features} \\ \begin{bmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{bmatrix} \end{array} \text{rows = observations}$$

We normalize each of the columns

Principal components analysis

a.k.a.
Karhunen–Loève Transform
Proper orthogonal decomposition
Hotelling transform

Transform the data from a high dimensional space to a lower dimensional subspace, while minimizing the projection error

$$A \qquad \boldsymbol{x} \qquad \rightarrow \qquad \boldsymbol{z}$$

$$\mathbb{R}^{M\times D} \qquad \mathbb{R}^{D} \qquad \mathbb{R}^{M} \qquad M < D$$

$$\begin{bmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{M1} & \cdots & u_{MD} \end{bmatrix} \qquad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_D \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_M \end{bmatrix}$$

linear transformation matrix
(this is what we want to find through PCA)

sample of data in original D-dimensional space
(this is one of N observations)

Transformed data in M-dimensional (lower dimensional) subspace

Principal components analysis

$$A$$

$$\begin{bmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{M1} & \cdots & u_{MD} \end{bmatrix} = \begin{bmatrix} - \boldsymbol{u}_1^T - \\ \vdots \\ - \boldsymbol{u}_M^T - \end{bmatrix}$$

linear transformation matrix

Each $\boldsymbol{u}_i$ represents a unit vector

The i^{th} principal component:

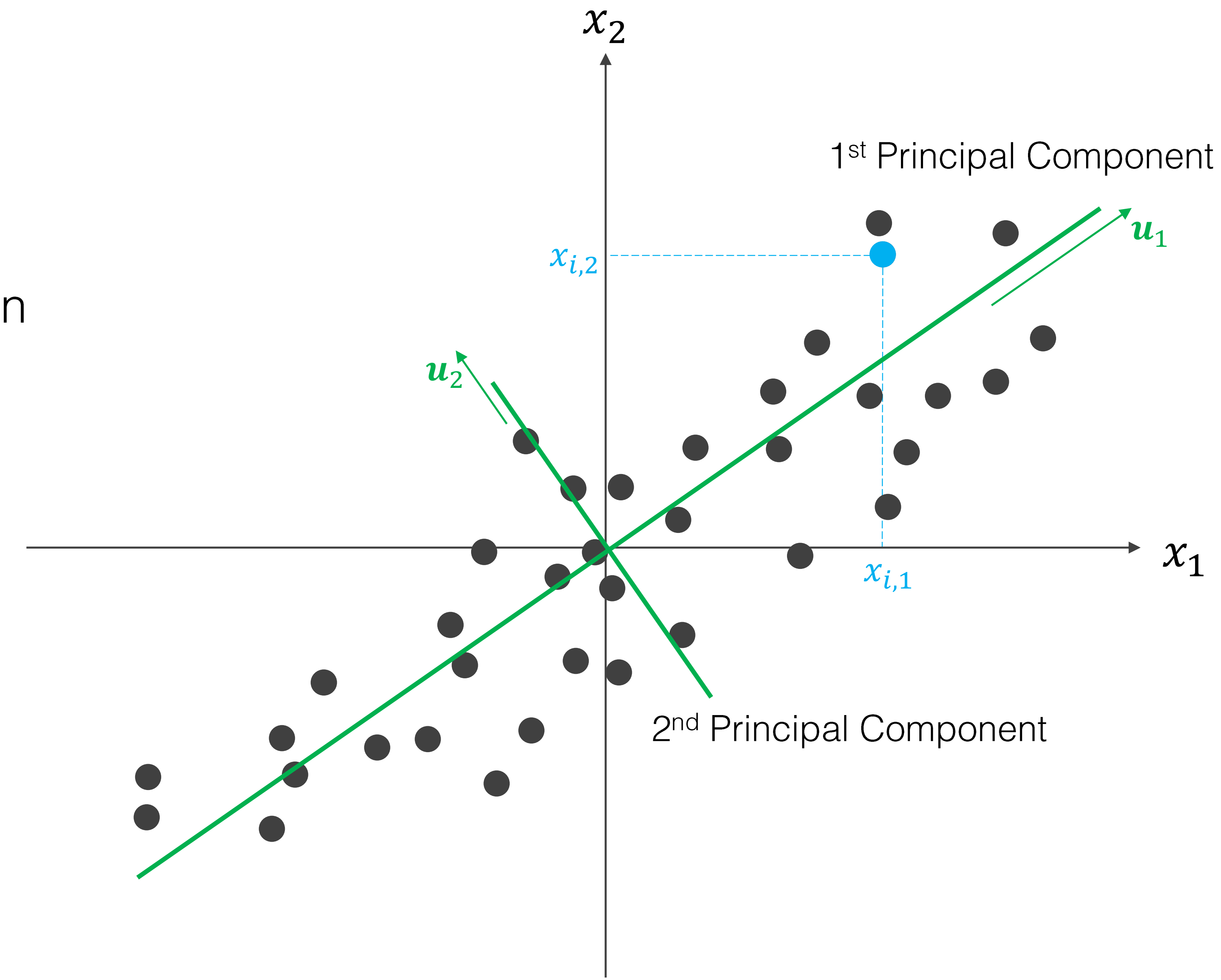
$$\underset{\text{scalar}}{\underset{\uparrow}{z_i}} = \underset{\text{unit vector}}{\underset{\uparrow}{\boldsymbol{u}_i^T}} \boldsymbol{x}$$

Since only direction matters, we assume the $\boldsymbol{u}_i$ are unit vectors

$$\boldsymbol{u}_i^T \boldsymbol{u}_i = 1$$

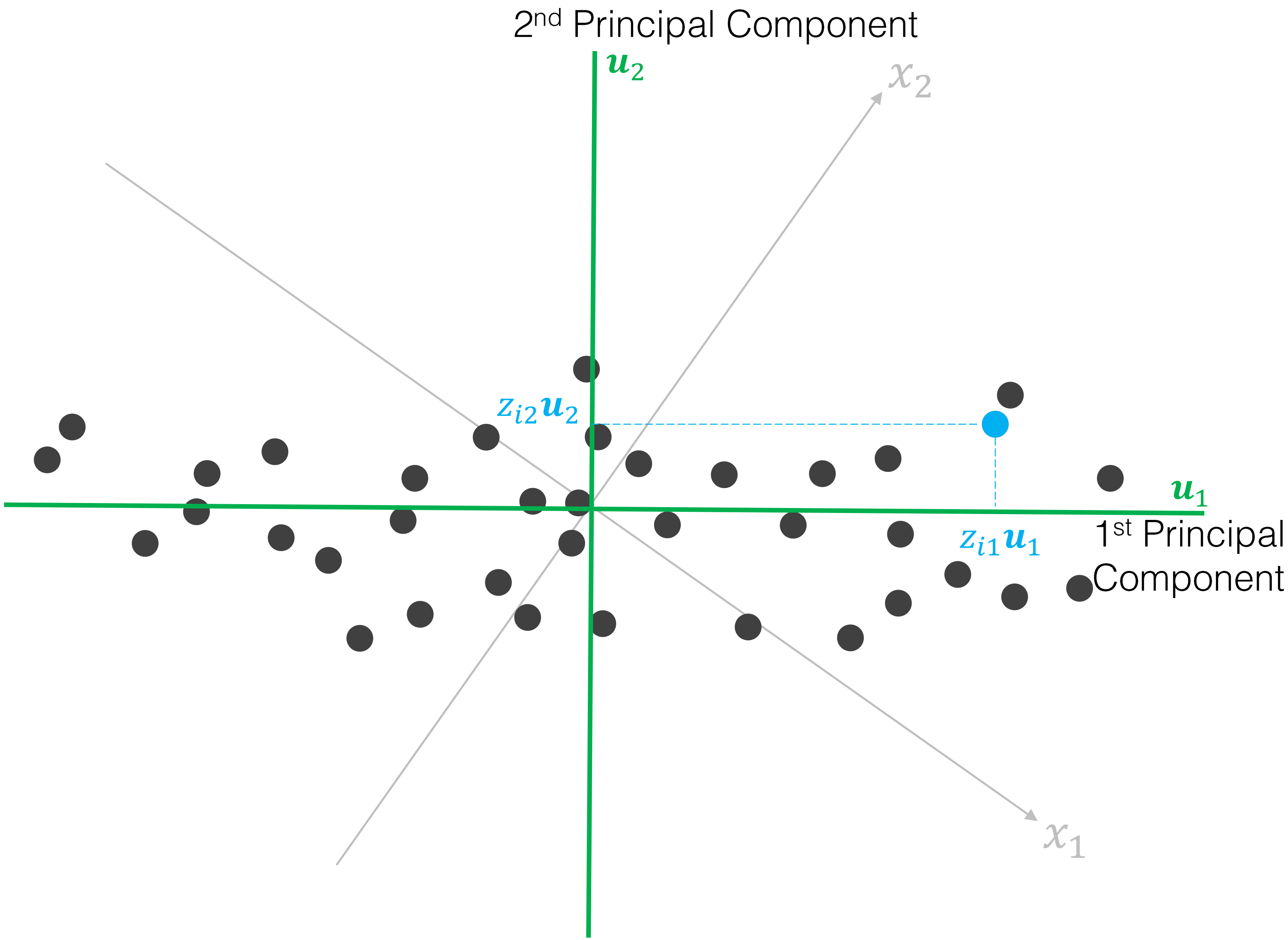
Principal Components

Maximum variance formulation



$$\boldsymbol{x}_i = \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix}$$

Reprojected Data onto Principal Components



Any point $\boldsymbol{x}_i$ can be represented as a combination of the principle components

$$\boldsymbol{x}_i = \sum_{j=1}^{D} z_{ij}\boldsymbol{u}_j$$

The $\boldsymbol{u}_j$'s are an orthogonal basis for the space $\mathbb{R}^D$

Approximating data with principal components

scalar projection

principal component direction

$$\boldsymbol{x}_j = \sum_{i=1}^{D} (\boldsymbol{x}_j^T \boldsymbol{u}_i) \boldsymbol{u}_i$$