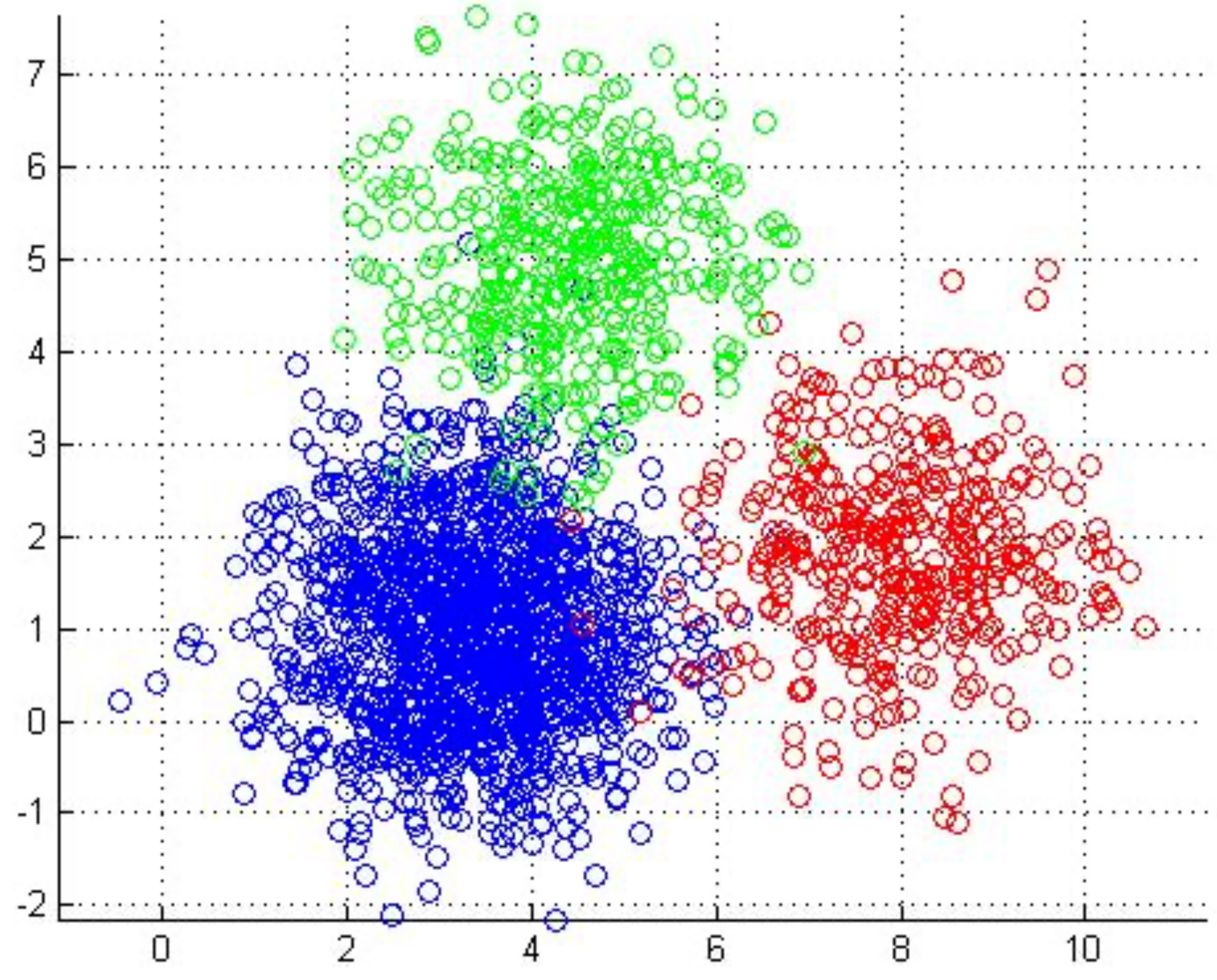
11. Dimensionality Reduction

EEC454 Introduction to Machine Learning Systems

2023 Fall, Jaeho Lee

Recap: Unsupervised Learning

- Discover useful structure of the data, using **unlabeled data**.
  - K-means clustering
  - Gaussian Mixture Models
  - Dimensionality Reduction (this week)
  - Autoencoders, GANs, Diffusion models, ...



Dealing with high-dimensional data

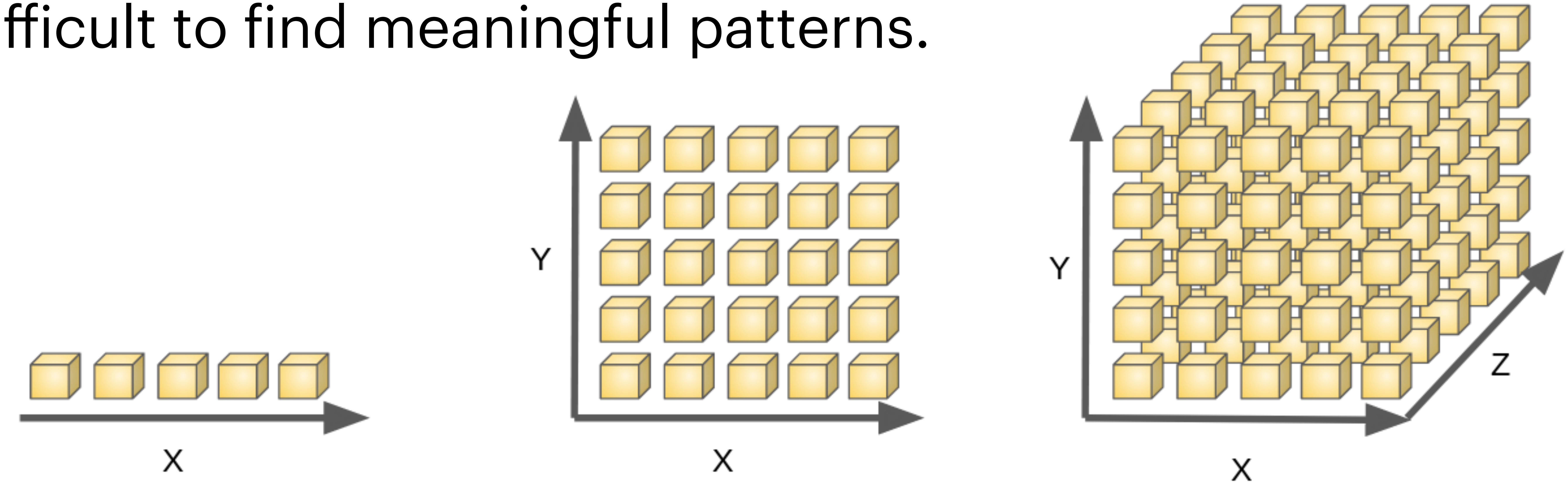
- Many datasets are extremely high-dimensional, in its raw form.
- Suppose that you are an ML engineer at Google.
  Then, you'd need to learn from these datasets:

YouTube Shorts
1920 x 1080 x 3 colors x 60 fps x 60 seconds
= 22.4 billion pixels (per video)

Gmail
1000s of words x sender info x receiver info x (images...)
= millions~billions real numbers (per mail)

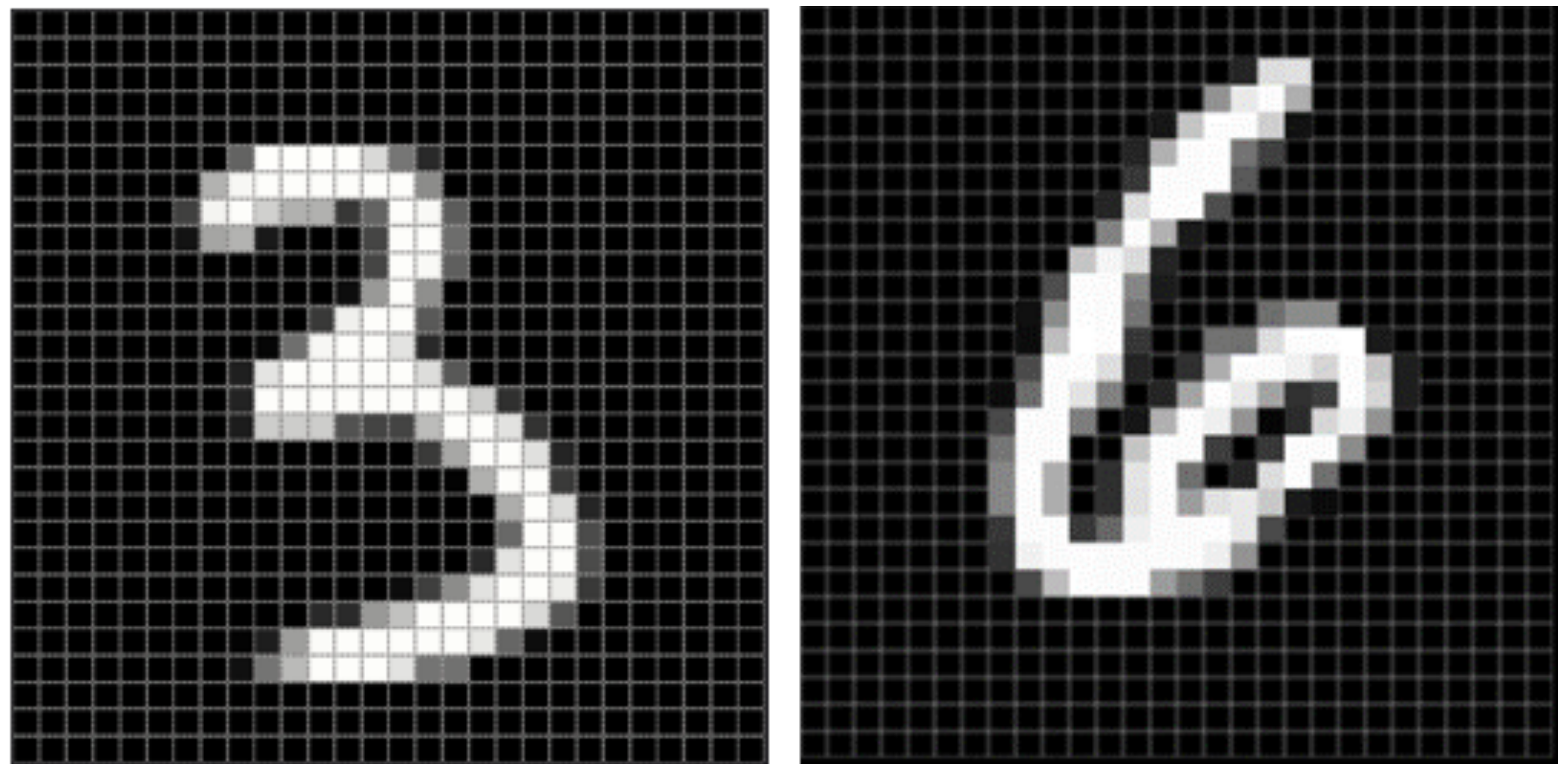
Curse of Dimensionality

- Higher-dimensional data are nasty to do ML on.
  - More computation.
  - Higher chance of noise.
  - Difficult to visualize (for human insight)
  - Difficult to find meaningful patterns.

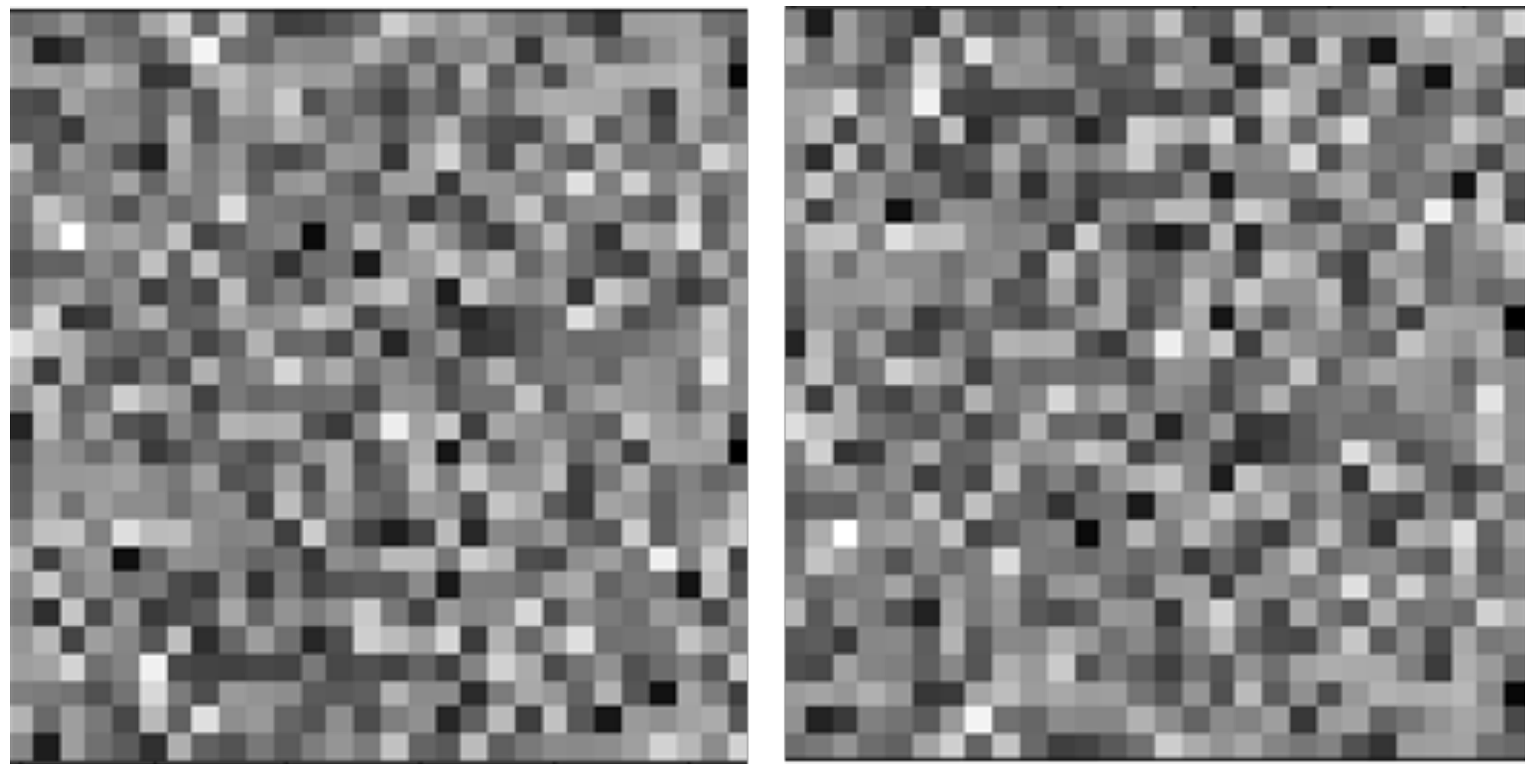


Dimensionality: Nominal vs. True

- But do we really need all dimensions?
  - Example. Handwritten Digit Recognition (MNIST, 28x28 image)



only looks like this



... and not like this

- That is, we may not need to **fully utilize** $\mathbb{R}^{28\times 28} = \mathbb{R}^{784}$.

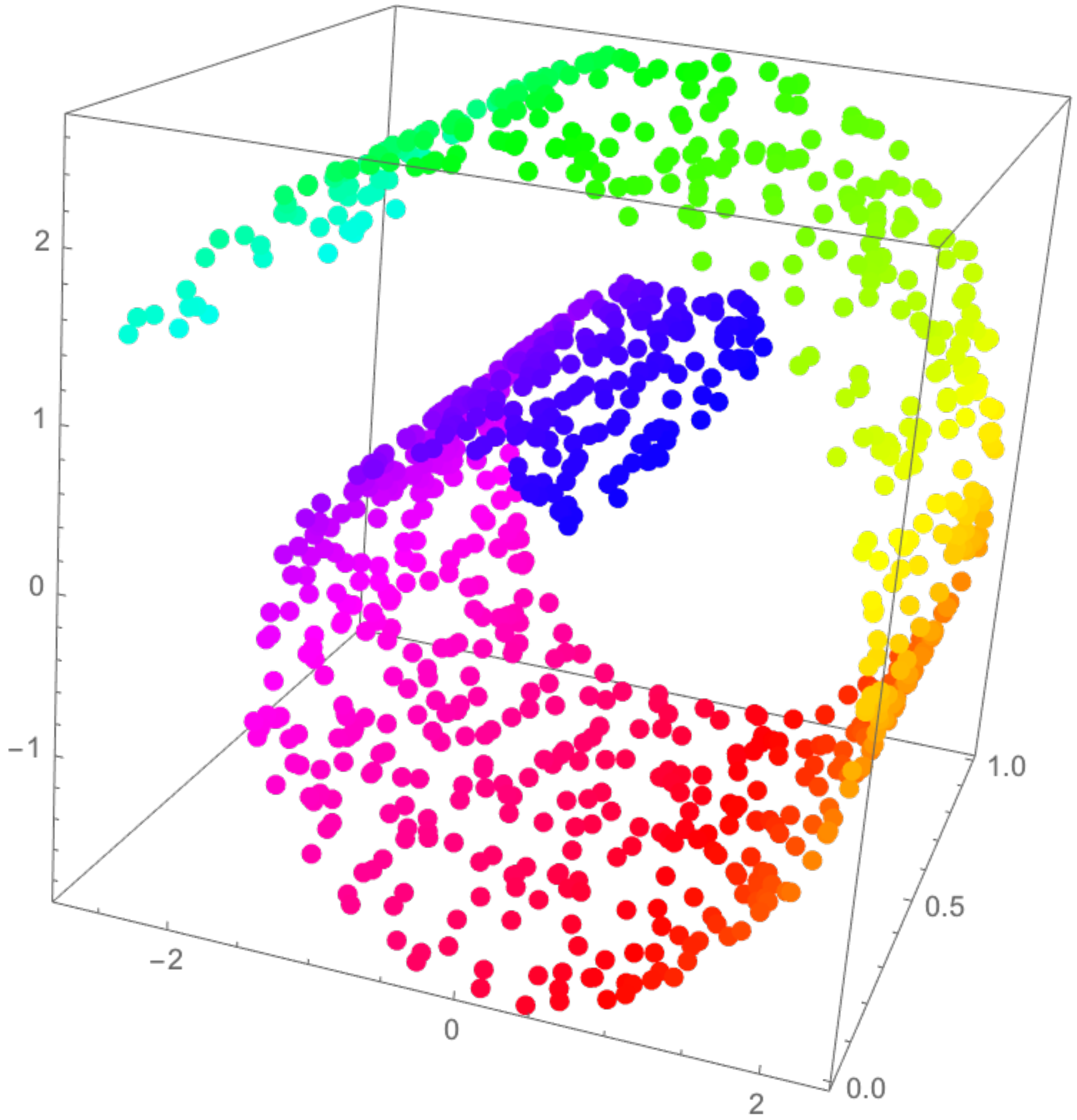
Dimensionality: Nominal vs. True

Hypothesis
There is a low-dimensional subspace (or submanifold) in the high-d space where the real data lies on.

Important. *Ignore small "noise" in each datum!*

Dimensionality Reduction
Finding these high-d -> low-d mapping.

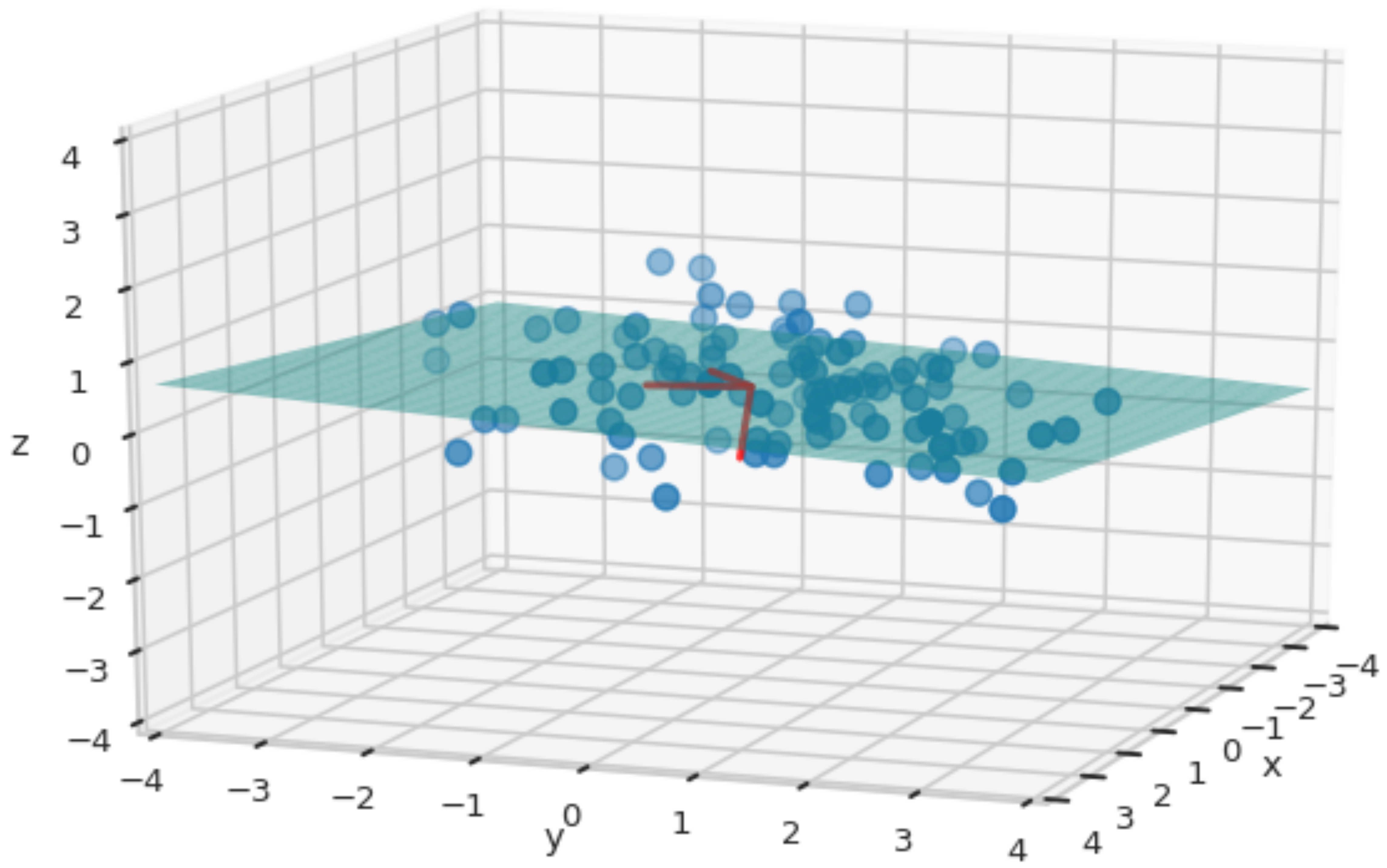
Note. *No need for labels!*



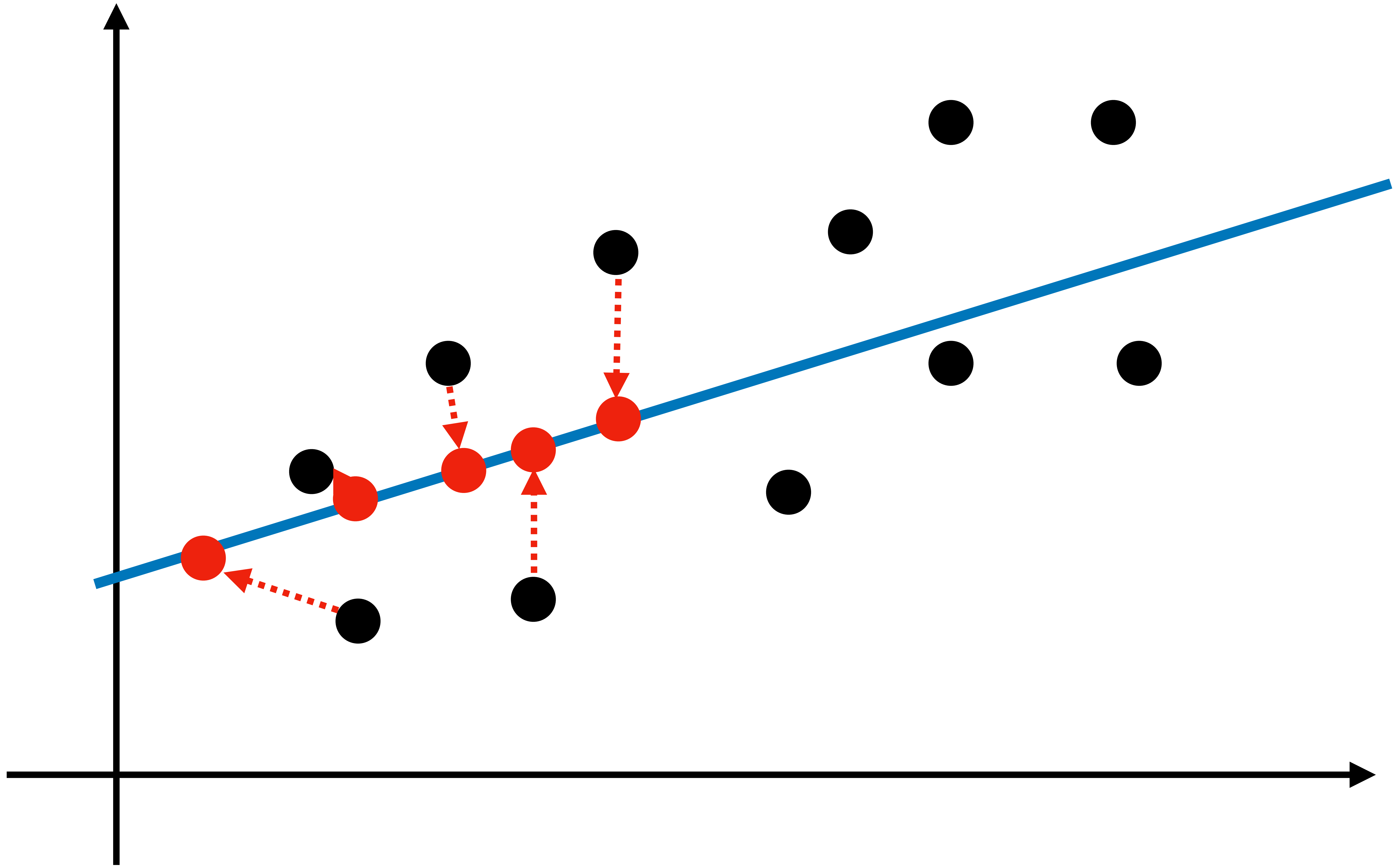
Principal Component Analysis

Principal Component Analysis

- Dimensionality reduction using a **affine subspace** of the original space
    - Invented by Karl Pearson (1909)
    - Many aliases, e.g., Karhunen-Loève Transform

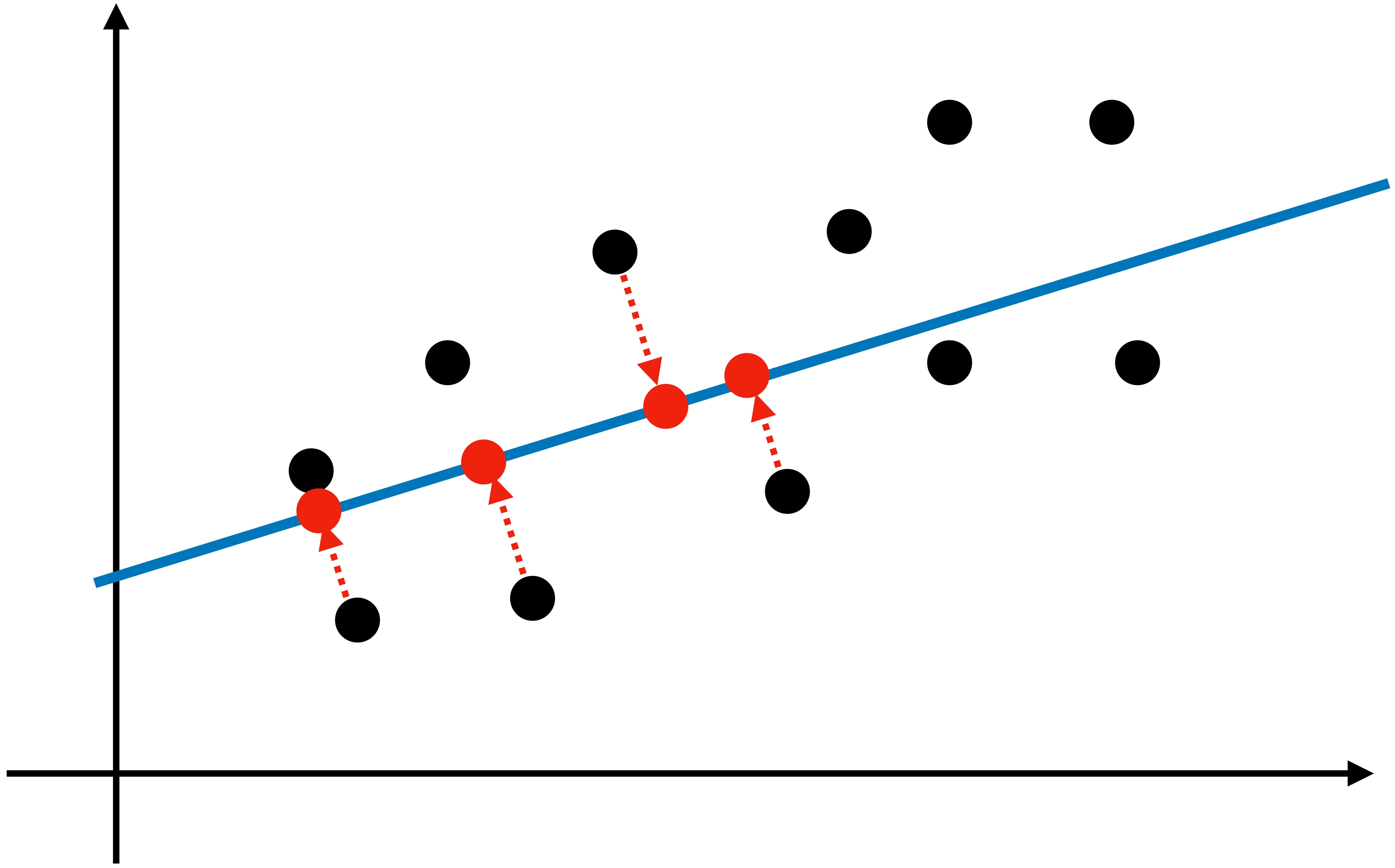


Suppose that we are given a 2D dataset—here, we want to find a **1D subspace** and a **mapping**, s.t. mapped data has **nice properties**.

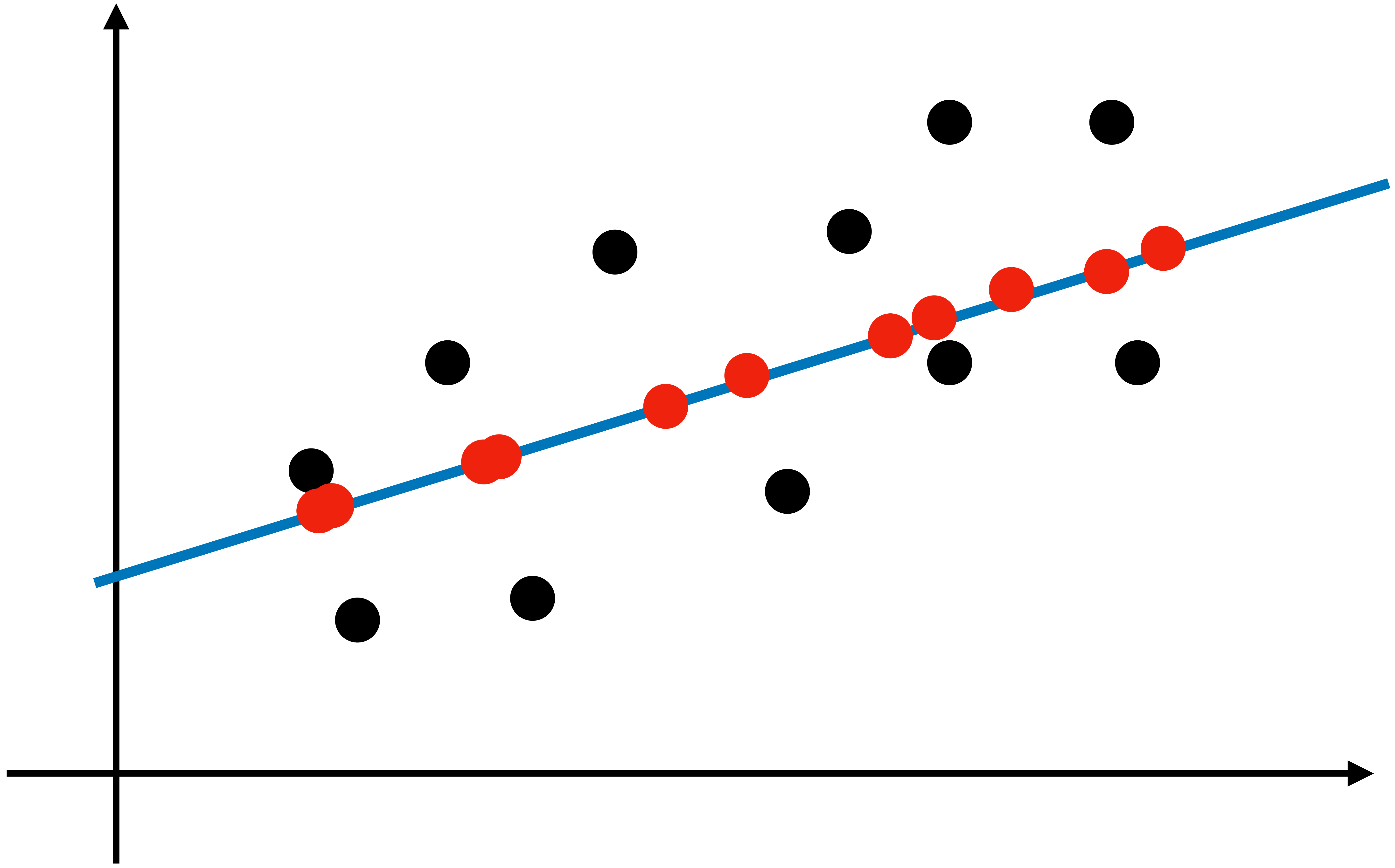


Suppose that we are given a 2D dataset—here, we want to find a **1D subspace** and a **mapping**, s.t. mapped data has **nice properties**.

simplify ⇒ only consider (orthogonal) projections to the subspace.

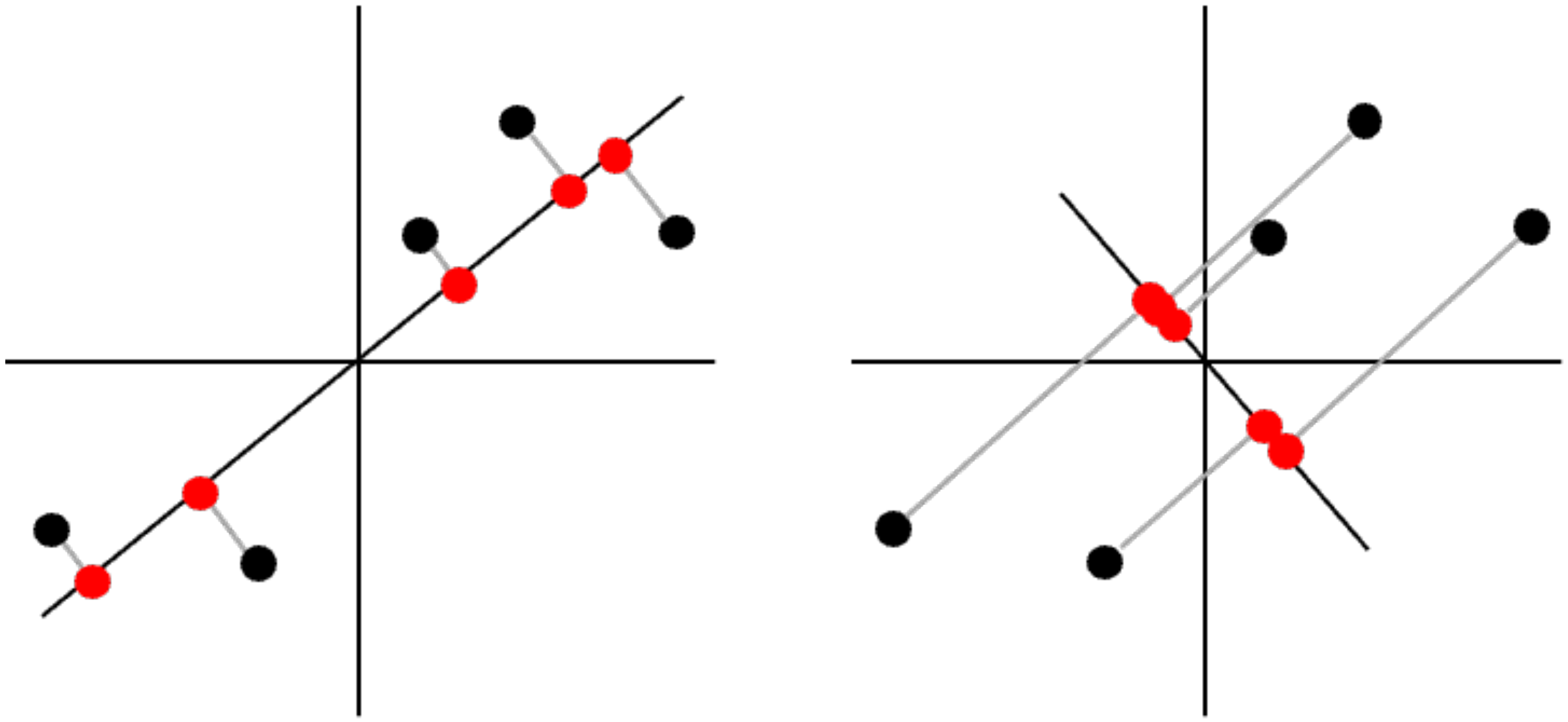


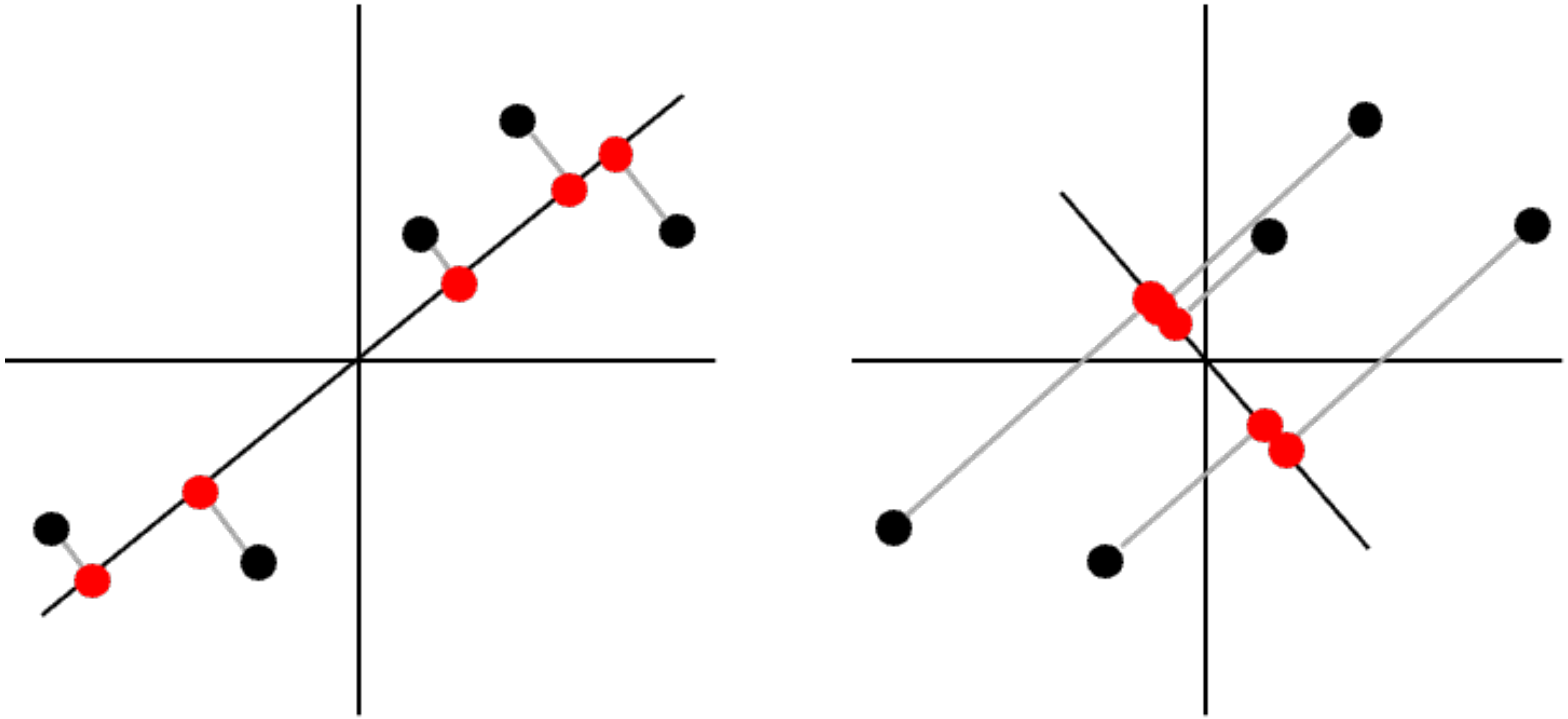
Suppose that we are given a 2D dataset—here, we want to find a **1D subspace**, s.t. the projected data has **nice properties**.



The Spirit

- We want to **preserve the information** as much as possible.
  - **Question.** Which projection contains more information?





- **Answer.** **Left!**
  A. Projected points are more widely spread.
  B. Original points (●) are closer to their projections (●)

(we will see that A and B are equivalent)

What PCA does, abstractly.

Suppose that we have the dataset $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$.

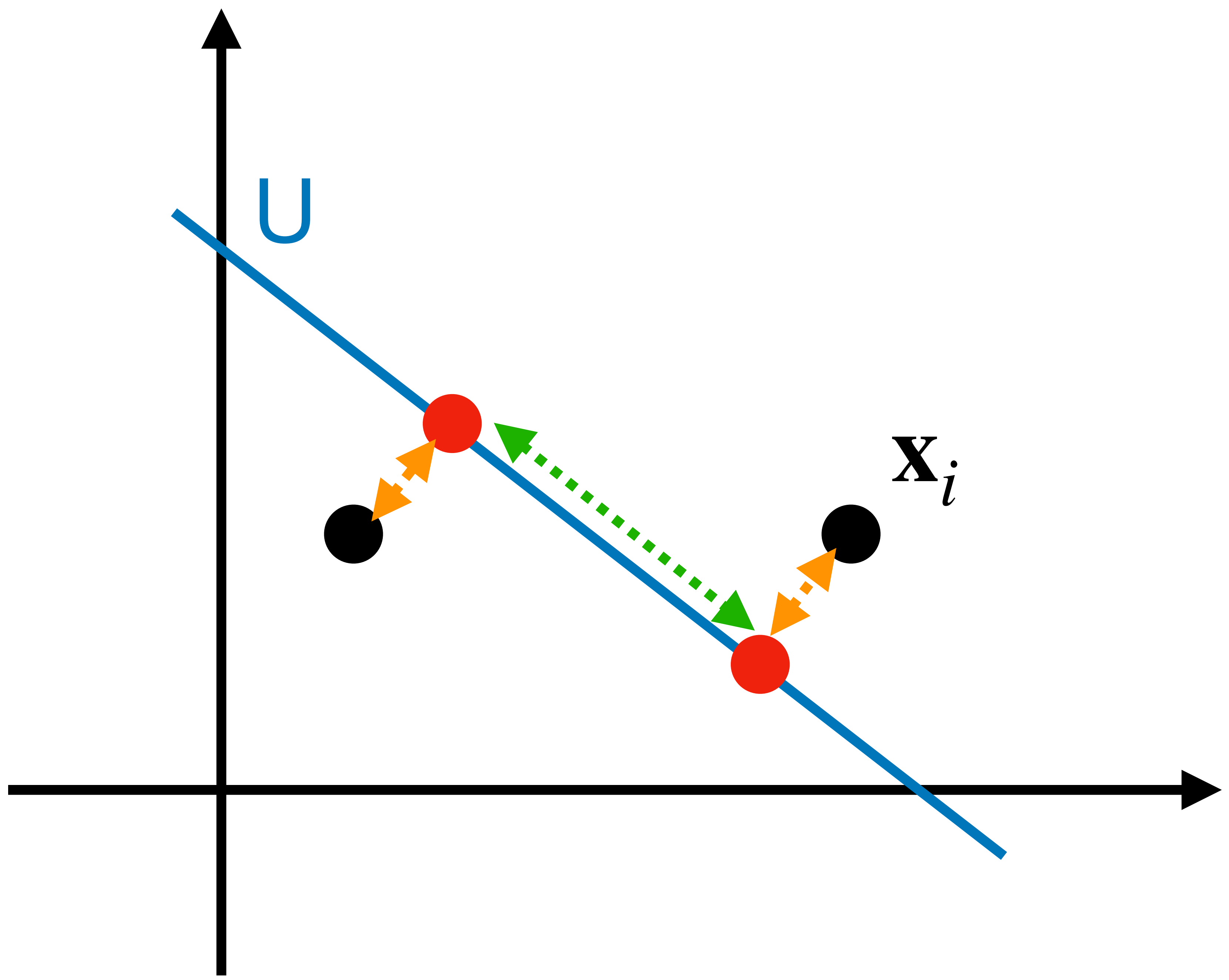
Goal. Find the k-dimensional subspace U of $\mathbb{R}^d$ such that:

- The projections has the **maximum variance**:

$$\max_{\mathsf{U}} \mathrm{Var}(\pi_{\mathsf{U}}(\mathbf{x}_1), \ldots, \pi_{\mathsf{U}}(\mathbf{x}_n))$$

- The **distortion** from projection is **minimized**:

$$\min_{\mathsf{U}} \sum_{i=1}^{n} \|\mathbf{x}_i - \pi_{\mathsf{U}}(\mathbf{x}_i)\|_2^2$$

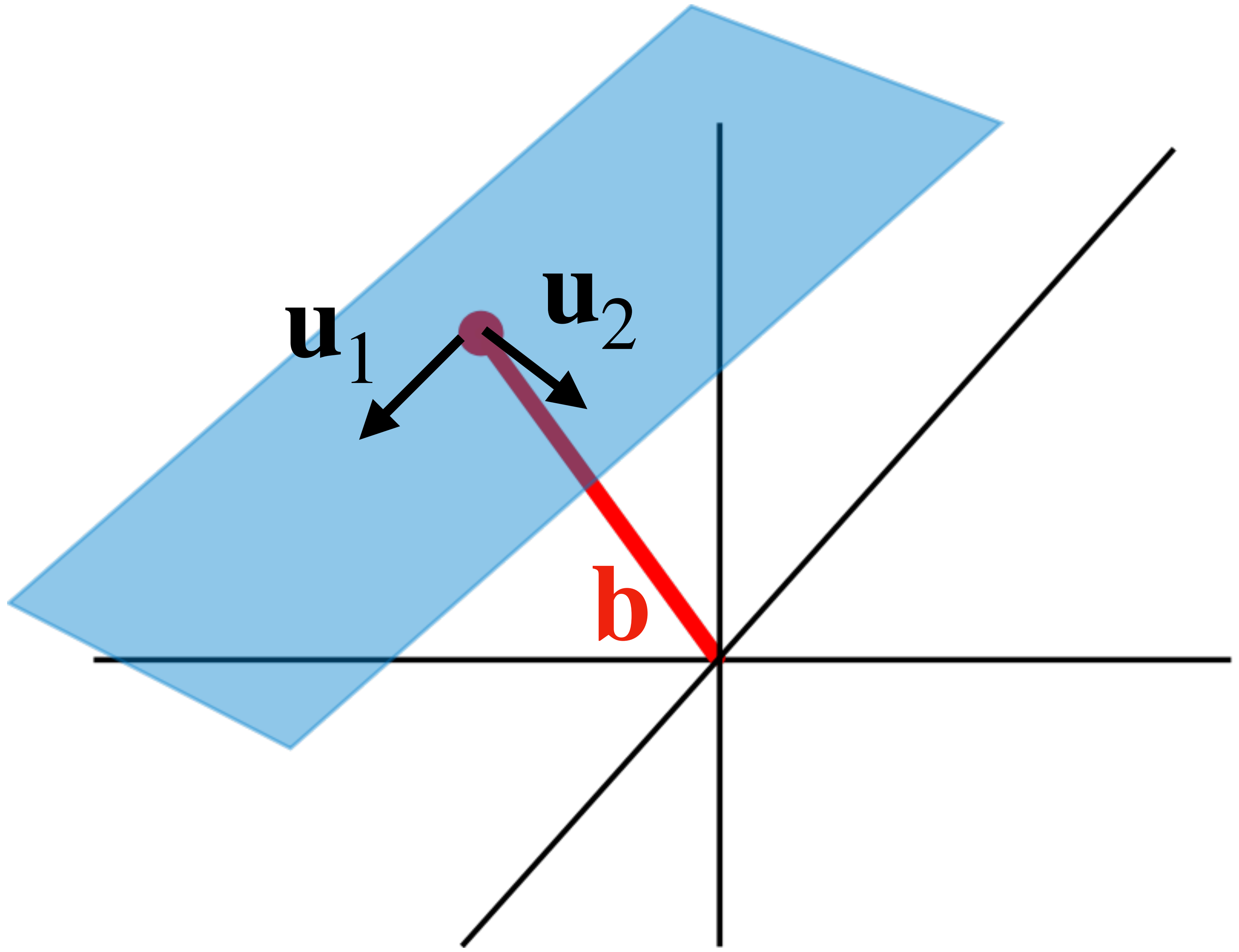


PCA as a Variance Maximization

Formalism: Affine Subspace

- A k-dimensional affine subspace $\mathsf{U} \subset \mathbb{R}^d$ can be characterized by its orthonormal bases $\mathbf{u}_1, \ldots, \mathbf{u}_k \in \mathbb{R}^d$ and an orthogonal bias $\mathbf{b} \in \mathbb{R}^d$ as

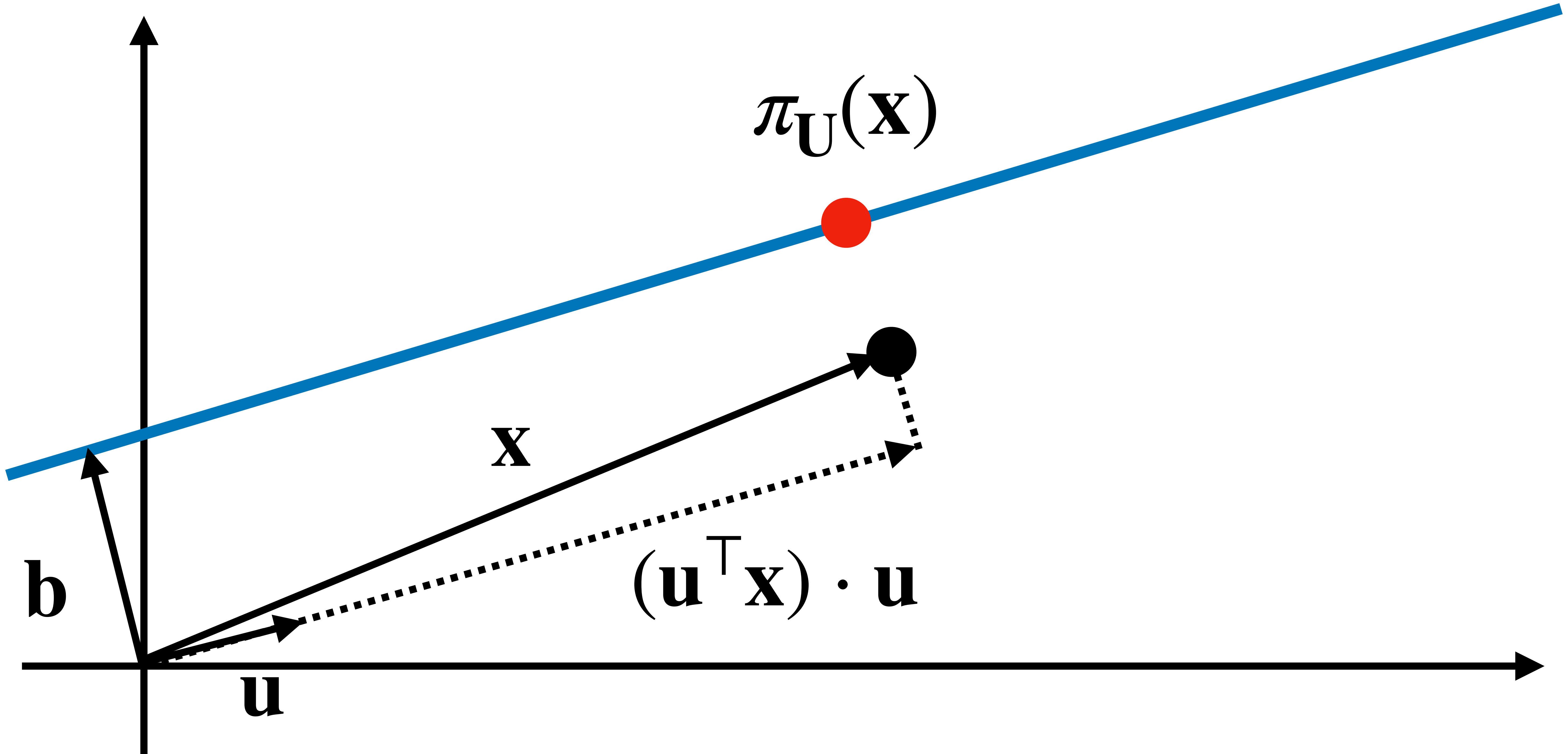
$$\mathsf{U} = \{a_1\mathbf{u}_1 + \cdots + a_k\mathbf{u}_k + \mathbf{b} \; : \; a_i \in \mathbb{R}\}$$



Formalism: Projection

- A projection of a vector $\mathbf{x} \in \mathbb{R}^d$ to an affine subspace U is

$$\pi_{\mathsf{U}}(\mathbf{x}) = \sum_{i=1}^{k} (\mathbf{u}_i^\top \mathbf{x}) \cdot \mathbf{u}_i + \mathbf{b}$$



Formalism: Projection

- This can be neatly written as a matrix form:

$$\pi_{\mathsf{U}}(\mathbf{x}) = \sum_{i=1}^{k} (\mathbf{u}_i^\top \mathbf{x}) \cdot \mathbf{u}_i + \mathbf{b}$$

$$= \left(\sum_{i=1}^{k} \mathbf{u}_i \mathbf{u}_i^\top \right) \mathbf{x} + \mathbf{b}$$

$$=: \mathbf{U}\mathbf{x} + \mathbf{b}$$

a $d \times d$ matrix with the rank k

Formalism: Projection

- The projection matrix has some useful properties.
  - $\mathbf{U}^{\top} = \mathbf{U}$
  - $\mathbf{U}^{\top}\mathbf{U} = \mathbf{U}$

(check by yourself!)

Variance maximization as a quadratic opt.

- Now, let's start looking into the variance maximization.

- We want to maximize the variance of the projected points, i.e.,

$$\mathrm{Var}\left(\mathbf{U}\mathbf{x}_1 + \mathbf{b}, \ldots, \mathbf{U}\mathbf{x}_n + \mathbf{b}\right)$$

- Because a constant term does not affect variance, this is equal to

$$\mathrm{Var}\left(\mathbf{U}\mathbf{x}_1, \ldots, \mathbf{U}\mathbf{x}_n\right)$$

Variance maximization as a quadratic opt.

$$\mathrm{Var}\Big(\mathbf{U}\mathbf{x}_1, \ldots, \mathbf{U}\mathbf{x}_n\Big)$$

- The mean of the $\{\mathbf{U}\mathbf{x}_i\}$ is $\mathbf{U}\bar{\mathbf{x}}$, where $\bar{\mathbf{x}}$ is the mean of $\{\mathbf{x}_i\}$.
- Thus, the variance is equal to

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n} \|\mathbf{U}(\mathbf{x}_i - \bar{\mathbf{x}})\|_2^2 &= \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{U}^\top \mathbf{U}(\mathbf{x}_i - \bar{\mathbf{x}}) \\ &= \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{U}(\mathbf{x}_i - \bar{\mathbf{x}})\end{aligned}$$

Variance maximization as a quadratic opt.

$$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{U}(\mathbf{x}_i - \bar{\mathbf{x}})$$

- By definition of $\mathbf{U}$, we can re-write the above as

$$\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}(\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{u}_j\mathbf{u}_j^\top(\mathbf{x}_i - \bar{\mathbf{x}})$$

$$= \sum_{j=1}^{k}\mathbf{u}_j^\top \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top\right)}_{=\text{ sample covariance matrix } S}\mathbf{u}_j$$

= sample covariance matrix S

(positive-semidefinite)

Variance maximization as a quadratic opt.

- Thus, PCA is solving the **quadratic optimization**

$$\max_{\mathbf{u}_1,\ldots,\mathbf{u}_k} \sum_{j=1}^{k} \mathbf{u}_j^\top \mathbf{S} \mathbf{u}_j$$

subject to the **constraints**

$$\mathbf{u}_i^\top \mathbf{u}_j = \begin{cases} 1 & \cdots & i = j \\ 0 & \cdots & i \neq j \end{cases}.$$

Solving the quadratic optimization (k=1)

- Let us take a closer look at the problem.

$$\max_{\mathbf{u}_1,\ldots,\mathbf{u}_k} \sum_{j=1}^{k} \mathbf{u}_j^\top \mathbf{S} \mathbf{u}_j, \qquad \text{subject to} \qquad \mathbf{u}_i^\top \mathbf{u}_j = \mathbf{1}\{i = j\}$$

- Consider the simplest case where $k = 1$, i.e.,

$$\max_{\mathbf{u}} \mathbf{u}^\top \mathbf{S} \mathbf{u}, \qquad \text{subject to} \qquad \|\mathbf{u}\|_2 = 1$$

- We see that the $\mathbf{u}$ should be the eigenvector **of $\mathbf{S}$ corresponding to the largest eigenvalue** (i.e., the principal component) why?

Why principal component?

(Version 1) Routine answer

To solve the constrained optimization

$$\max_{\mathbf{u}} \mathbf{u}^\top \mathbf{S}\mathbf{u}, \qquad \text{subject to} \qquad \|\mathbf{u}\|_2 = \mathbf{u}^\top \mathbf{u} = 1,$$

consider the Lagrangian relaxation

$$\max_{\mathbf{u}} \mathbf{u}^\top \mathbf{S}\mathbf{u} + \alpha(1 - \mathbf{u}^\top \mathbf{u}).$$

The critical point is at the point $\mathbf{S}\mathbf{u} = \alpha\mathbf{u}$ holds (i.e., eigenvectors).

Choosing the principal coefficient maximizes the value of $\mathbf{u}^\top \mathbf{S}\mathbf{u}$

Why principal component?

(Version 2) If you don't like Lagrangian... (difficult to extend to k=2)

Let $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ be the unit-norm eigenvectors of $\mathbf{S}$,
with eigenvalues $(\lambda_1, \ldots, \lambda_d)$ in the descending order.

Any choice of $\mathbf{u}$ can be written as a *mixture of eigenvectors*

$$\mathbf{u} = w_1\mathbf{e}_1 + \cdots + w_d\mathbf{e}_d$$

with the weights $w_1^2 + \cdots + w_d^2 = 1$. (energy in each direction, with total budget 1)

Why principal component?

The system $\mathbf{S}$ scales each eigenvectors, i.e.,

$$\begin{aligned}\mathbf{Su} &= \mathbf{S}(w_1\mathbf{e}_1 + \cdots + w_d\mathbf{e}_d) \\ &= w_1\mathbf{S}\mathbf{e}_1 + \cdots + w_d\mathbf{S}\mathbf{e}_d \\ &= w_1\lambda_1\mathbf{e}_1 + \cdots + w_d\lambda_d\mathbf{e}_d\end{aligned}$$

Thus, we have

$$\mathbf{u}^\top\mathbf{Su} = w_1^2\lambda_1 + \cdots + w_d^2\lambda_d.$$

Optimal choice. Assign all weights to w_1, i.e., $\mathbf{u} = \mathbf{e}_1$.

The Next Component

- Now, consider the case where $k = 2$.

$$\max_{\mathbf{u}_1,\mathbf{u}_2} \mathbf{u}_1^\top \mathbf{S}\mathbf{u}_1 + \mathbf{u}_2^\top \mathbf{S}\mathbf{u}_2, \qquad \text{subject to } \|\mathbf{u}_1\| = \|\mathbf{u}_2\| = 1, \mathbf{u}_1^\top \mathbf{u}_2 = 0$$

- View this as a nested optimization problem

$$\max_{\|\mathbf{u}_1\|=1} \left(\mathbf{u}_1^\top \mathbf{S}\mathbf{u}_1 + \max_{\|\mathbf{u}_2\|=1, \mathbf{u}_2 \perp \mathbf{u}_1} \left(\mathbf{u}_2^\top \mathbf{S}\mathbf{u}_2 \right) \right).$$

- Then, take a look at the inner maximization problem.

$$\max_{\|\mathbf{u}_2\|=1, \mathbf{u}_2 \perp \mathbf{u}_1} \left(\mathbf{u}_2^\top \mathbf{S}\mathbf{u}_2 \right)$$

The Next Component

- The Lagrangian of the inner maximization becomes

$$\mathbf{u}_2^\top \mathbf{S}\mathbf{u}_2 + \alpha \cdot (1 - \mathbf{u}_2^\top \mathbf{u}_2) - \beta \cdot (\mathbf{u}_1^\top \mathbf{u}_2)$$

- The critical point condition is where:

$$\mathbf{S}\mathbf{u}_2 = \alpha \mathbf{u}_2 + \frac{\beta}{2}\mathbf{u}_1$$

- Multiplying $\mathbf{u}_1^\top$ on both sides, we get

$$\underbrace{\mathbf{u}_1^\top \mathbf{S}\mathbf{u}_2}_{=0} = \alpha \underbrace{\mathbf{u}_1 \mathbf{u}_2}_{=0} + \frac{\beta}{2}$$

... and thus $\beta = 0$

The Next Component

- Plugging in $\beta = 0$, we get

$$\mathbf{S}\mathbf{u}_2 = \alpha \mathbf{u}_2$$

- Thus, we should also select $\mathbf{u}_2$ as an eigenvector.
    - Selecting $\mathbf{u}_1$, $\mathbf{u}_2$ as eigenvectors for top-2 eigenvalues is optimal.

PCA, with k principal components

- Similarly, we can select the affine subspace spanned by

$$\{\mathbf{e}_1, \ldots, \mathbf{e}_k\},$$

where $\mathbf{e}_1, \ldots, \mathbf{e}_k$ are k principal components of the sample covariance matrix $\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$.

  - This can be done by performing SVD on the data matrix

$$\mathbf{X} = [\mathbf{x}_1 - \bar{\mathbf{x}} \mid \cdots \mid \mathbf{x}_n - \bar{\mathbf{x}}] = \mathbf{U}\Sigma\mathbf{V}^\top$$

and selecting the columns of $\mathbf{U}$ for top-k singular values.

Cheers

- ***Next up.*** PCA as minimum reconstruction error, Kernel PCA, t-SNE, …