Dimensionality Reduction (2)

EECE454 Intro. to Machine Learning Systems

Fall 2024

Recap

- **PCA.** Projecting data to an affine subspace spanned by principal components

  (top-k eigenvectors of data covariance matrix)

  - Projection can be done by $\mathbf{x} \mapsto \mathbf{U}\mathbf{x} + \mathbf{b}$

  - Derived as a solution of variance maximization:

$$\max_{\mathsf{U}} \mathrm{Var}\left(\{\pi_{\mathsf{U}}(\mathbf{x}_i)\}_{i=1}^{n}\right)$$

projection of $\mathbf{x}_i$ on the affine subspace U

Recap

- **PCA.** Projecting data to an affine subspace spanned by principal components

  (top-k eigenvectors of data covariance matrix)

  - Projection can be done by $\mathbf{x} \mapsto \mathbf{U}\mathbf{x} + \mathbf{b}$

  - Derived as a solution of variance maximization:

$$\max_{\mathsf{U}} \mathrm{Var}\left(\{\pi_{\mathsf{U}}(\mathbf{x}_i)\}_{i=1}^{n}\right)$$

- **Today.** Variance maximization = Distortion minimization

  - Gives us a natural way to determine $\mathbf{b}$

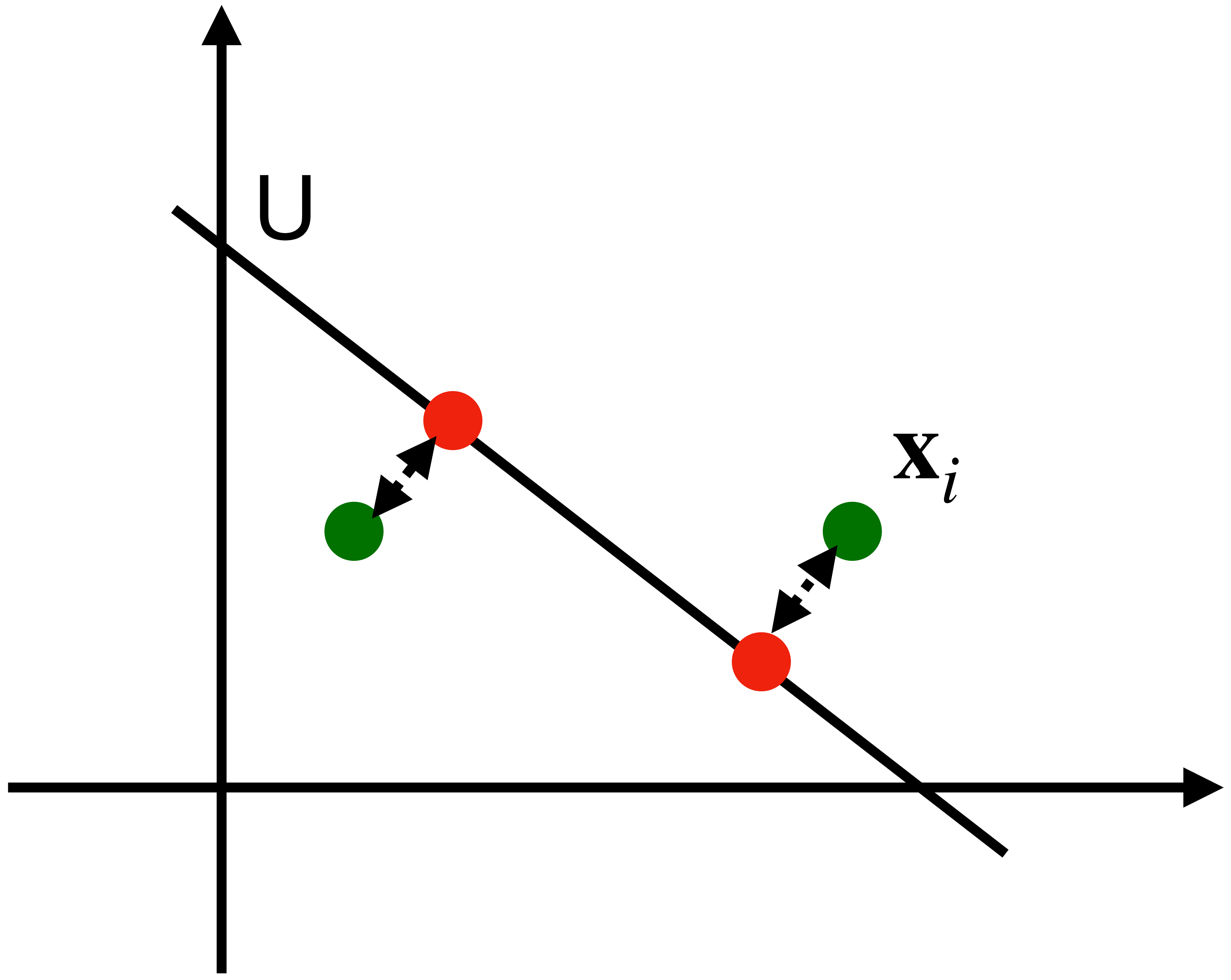
  - Explains why “projection” should be considered as our mapping to the subspace

PCA: Distortion minimization

Distortion minimization

- Here's the perspective:

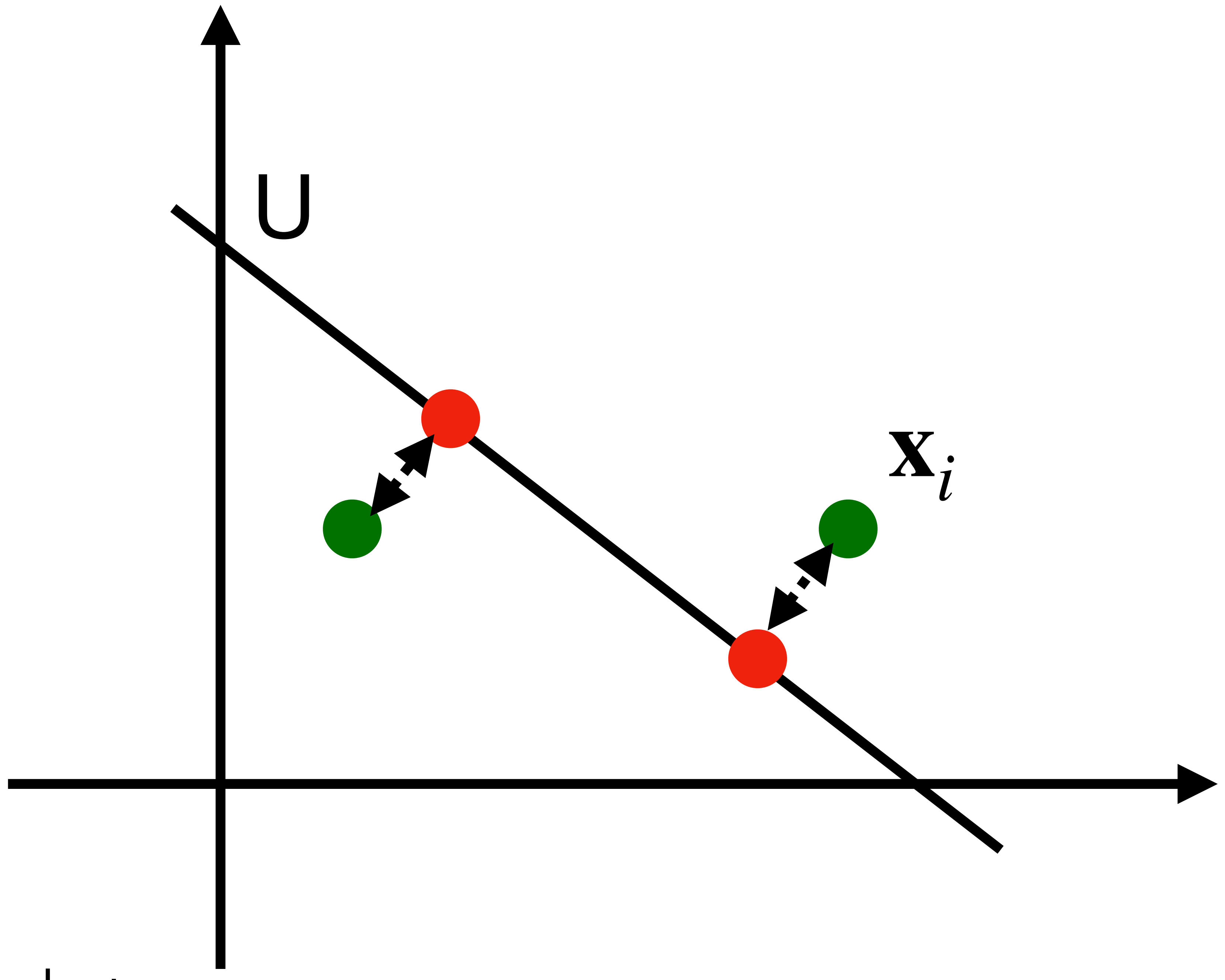
  "If the projected point is close to the original point,
  maybe it did not loose too much original information"



Distortion minimization

- Here's the perspective:

  "If the projected point is close to the original point,
  maybe it did not loose too much original information"



- In fact, this is quite natural—
  - Suppose that we use some predictor $f(\cdot)$ on the projected data
  - Then, we have

$$f(\mathbf{x}) - f(\pi_{\mathsf{U}}(\mathbf{x})) \leq \mathrm{Lip}(f) \cdot \|\mathbf{x} - \pi_{\mathsf{U}}(\mathbf{x})\|$$

(here, $\mathrm{Lip}(f) = \sup_{x \neq y} |f(\mathbf{x}) - f(\mathbf{y})| / \|\mathbf{x} - \mathbf{y}\|$ is the "Lipschitz constant")

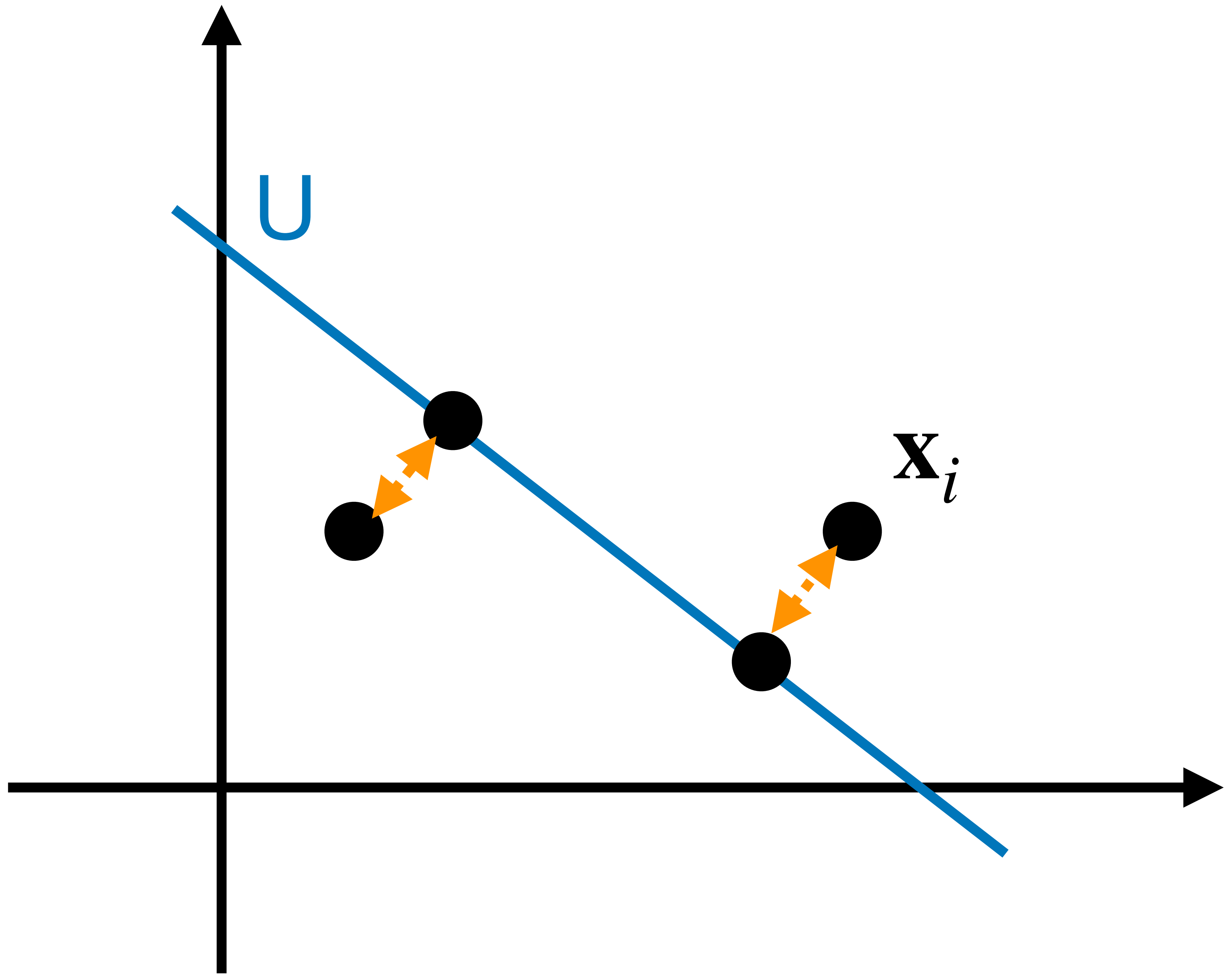
Formally…

- Formally, we try to find an **affine subspace**

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

such that the **mean squared distortion** of data, incurred by projection, is minimized:

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



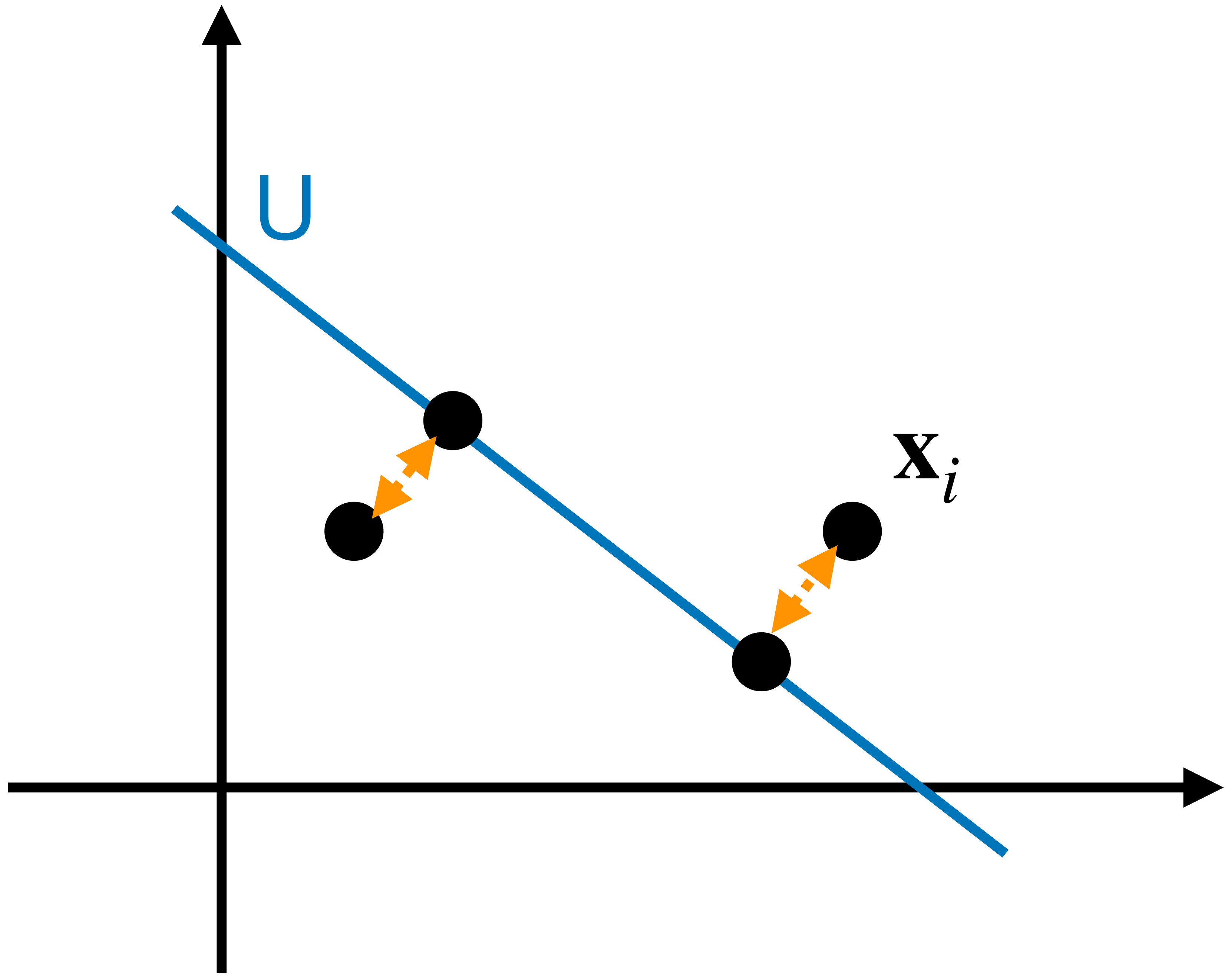
Formally...

- Formally, we try to find an **affine subspace**

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

such that the **mean squared distortion** of data, incurred by projection, is minimized:

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



- Using the definition of projection, this is:

$$\min_{\mathbf{U},\mathbf{b}} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{U}\mathbf{x}_i - \mathbf{b}\|^2$$

Formally…

- Then, we can proceed as

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \pi_{\mathsf{U}}(\mathbf{x}_i)\|^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\|\mathbf{x}_i\|^2 + \|\mathbf{b}\|^2 - \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top \mathbf{x}_i + 2\mathbf{b}^\top \mathbf{U}\mathbf{x}_i\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 \; + \; \|\mathbf{b}\|^2 - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top \bar{\mathbf{x}} + 2\mathbf{b}^\top \mathbf{U}\bar{\mathbf{x}}$$

- Separating out irrelevant terms, we get

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \min_{\mathbf{U},\mathbf{b}}\left(\|\mathbf{b}\|^2 - \frac{1}{n}\sum \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top \bar{\mathbf{x}} + 2\mathbf{b}^\top \mathbf{U}\bar{\mathbf{x}}\right)$$

Formally...

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \min_{\mathbf{U},\mathbf{b}}\left(\|\mathbf{b}\|^2 - \frac{1}{n}\sum \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top\bar{\mathbf{x}} + 2\mathbf{b}^\top\mathbf{U}\bar{\mathbf{x}}\right)$$

- Minimizing with respect to $\mathbf{b}$, we get $\mathbf{b}^* = \bar{\mathbf{x}} - \mathbf{U}\bar{\mathbf{x}}$

Formally...

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \min_{\mathbf{U},\mathbf{b}}\left(\|\mathbf{b}\|^2 - \frac{1}{n}\sum \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top\bar{\mathbf{x}} + 2\mathbf{b}^\top\mathbf{U}\bar{\mathbf{x}}\right)$$

- Minimizing with respect to $\mathbf{b}$, we get $\mathbf{b}^* = \bar{\mathbf{x}} - \mathbf{U}\bar{\mathbf{x}}$
  - Plugging in, we get:

$$\left(\frac{1}{n}\sum \|\mathbf{x}_i\|^2 - \bar{\mathbf{x}}^\top\bar{\mathbf{x}}\right) + \min_{\mathbf{U}}\left(\bar{\mathbf{x}}^\top\mathbf{U}\bar{\mathbf{x}} - \frac{1}{n}\sum \mathbf{x}_i^\top\mathbf{U}\mathbf{x}_i\right)$$

$$= \mathrm{Var}\left(\{\mathbf{x}_i\}_{i=1}^n\right)$$

$$= -\sum_{j=1}^{k} \mathbf{u}_j^\top \mathbf{S}\mathbf{u}_j$$

Formally...

$$\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i\|^2 + \min_{\mathbf{U},\mathbf{b}}\left(\|\mathbf{b}\|^2 - \frac{1}{n}\sum \mathbf{x}_i^\top \mathbf{U}\mathbf{x}_i - 2\mathbf{b}^\top\bar{\mathbf{x}} + 2\mathbf{b}^\top\mathbf{U}\bar{\mathbf{x}}\right)$$

- Minimizing with respect to $\mathbf{b}$, we get $\mathbf{b}^* = \bar{\mathbf{x}} - \mathbf{U}\bar{\mathbf{x}}$
  - Plugging in, we get:

$$\left(\frac{1}{n}\sum\|\mathbf{x}_i\|^2 - \bar{\mathbf{x}}^\top\bar{\mathbf{x}}\right) + \min_{\mathbf{U}}\left(\bar{\mathbf{x}}^\top\mathbf{U}\bar{\mathbf{x}} - \frac{1}{n}\sum \mathbf{x}_i^\top\mathbf{U}\mathbf{x}_i\right)$$

  - Rephrasing, we arrive at:

$$\min_{\mathsf{U}}\frac{1}{n}\sum_{i=1}^{n}\|\mathbf{x}_i - \pi_{\mathsf{U}}(\mathbf{x}_i)\|^2 = \mathrm{Var}(\{\mathbf{x}_i\}) - \max_{\mathbf{U}}\left(\sum_{j=1}^{k}\mathbf{u}_j\mathbf{S}\mathbf{u}_j\right)$$

exactly what we solved for
variance maximization problem

Applications of PCA

Face recognition

- **Goal.** Identify specific person, based on facial image
  - Robust to glasses, lighting, ...
  - Using 256 x 256 pixels is difficult!

Face recognition

- **Goal.** Identify specific person, based on facial image
    - Robust to glasses, lighting, ...
    - Using 256 x 256 pixels is difficult!
- **Idea.** Build one PCA database for the whole dataset (eigenface)
    - Classify based on weights $(\mathbf{u}_1^\top \mathbf{x}, \ldots, \mathbf{u}_k^\top \mathbf{x})$
    - Advantages. Rapid recognition, tracking, reconstruction ...

Face recognition

- **Goal.** Identify specific person, based on facial image
  - Robust to glasses, lighting, ...
  - Using 256 x 256 pixels is difficult!
- **Idea.** Build one PCA database for the whole dataset (eigenface)
  - Classify based on weights $(\mathbf{u}_1^\top \mathbf{x}, \ldots, \mathbf{u}_k^\top \mathbf{x})$
  - Advantages. Rapid recognition, tracking, reconstruction ...
  - Shortcomings. Requires same size
    Sensitive to angles
    Needs "centering" of the face ...

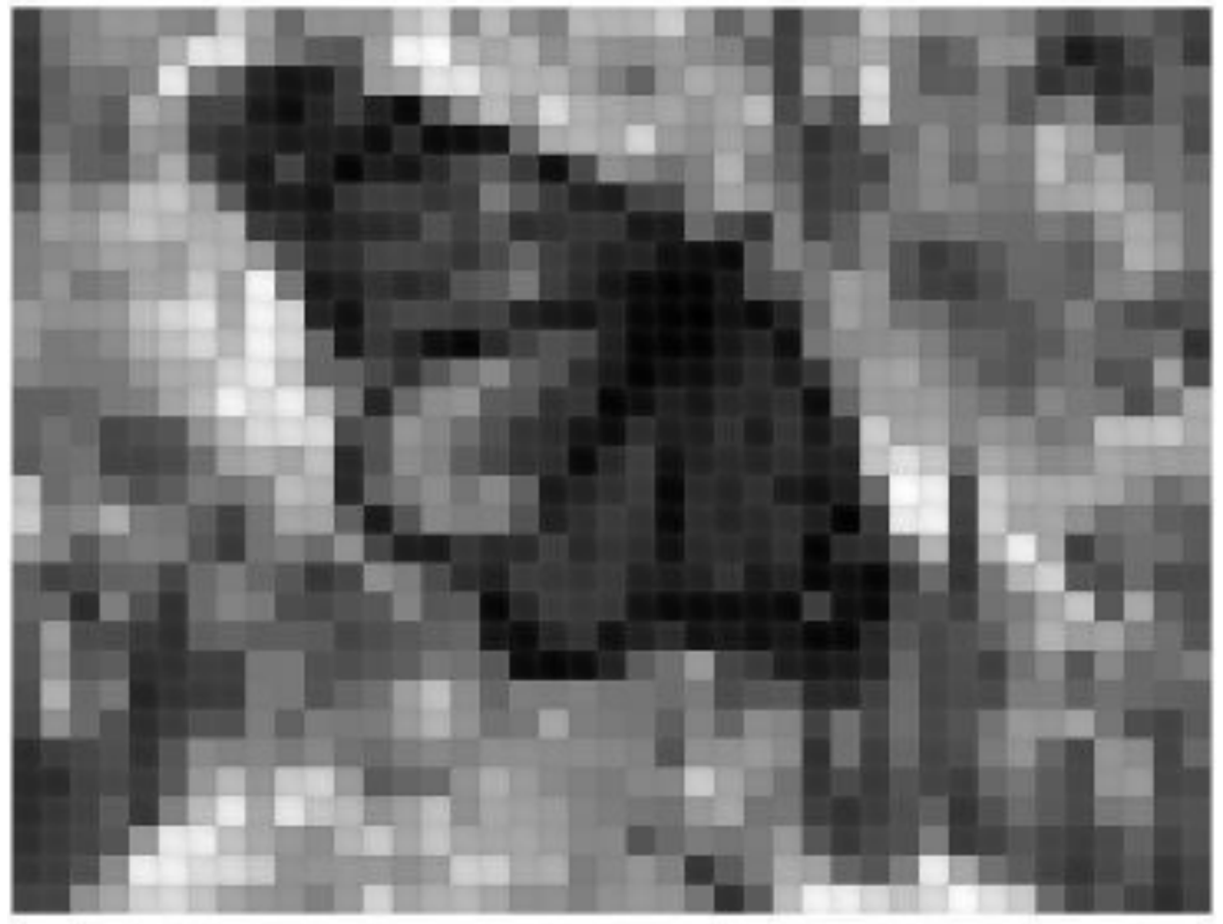
Image Compression

- **Goal.** Represent an image using less dimensions
- **Idea.** Do the following
  - Divide each image into 12×12 patches
  - Perform PCA and select top-k directions
  - Save the codes $(\mathbf{u}_1^\top \mathbf{x}, \ldots, \mathbf{u}_k^\top \mathbf{x})$ for each patch (requires saving the "codebook" $\mathbf{u}_1, \ldots, \mathbf{u}_k$)

144-dimension
(full)

60-dimension

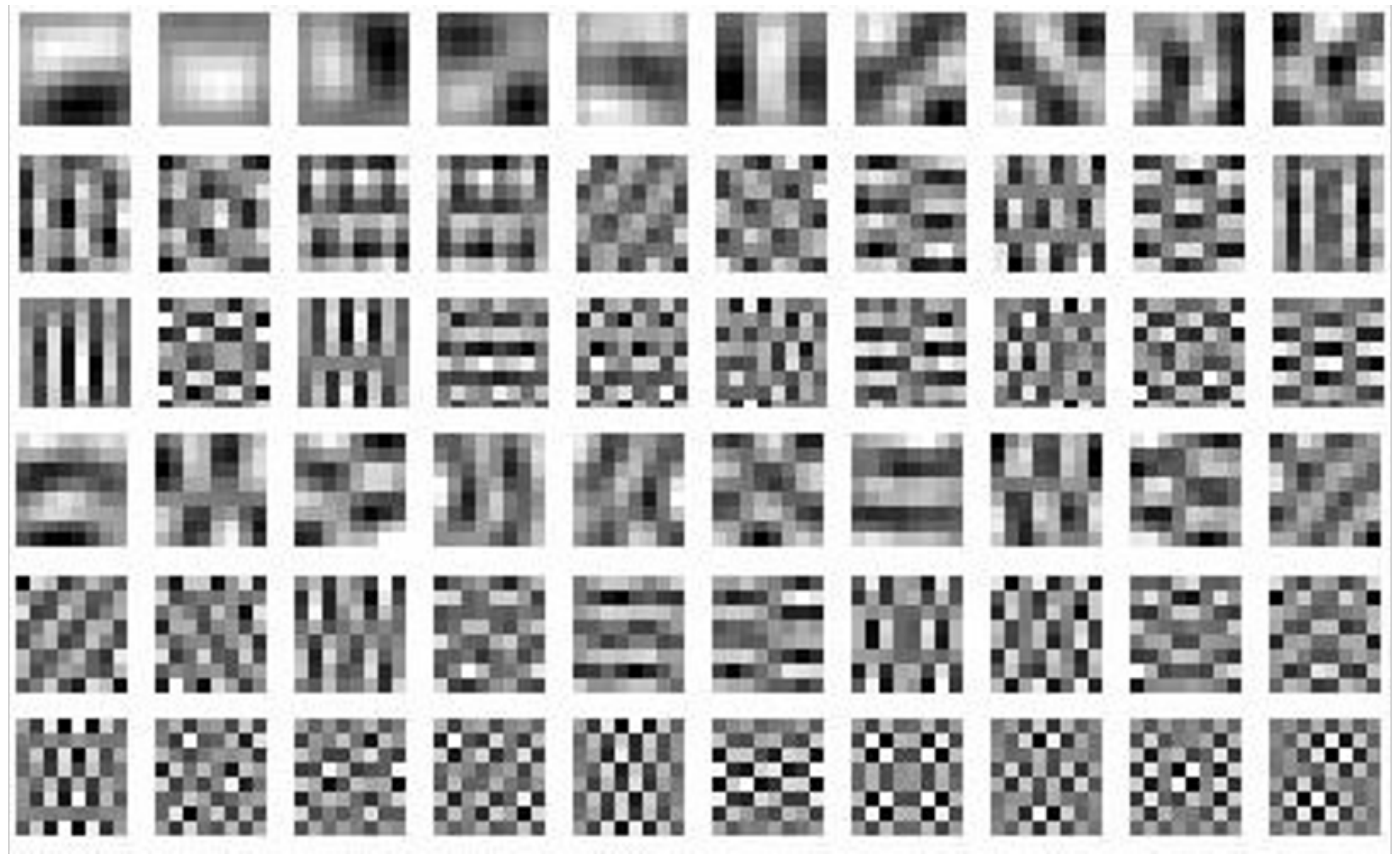
6-dimension



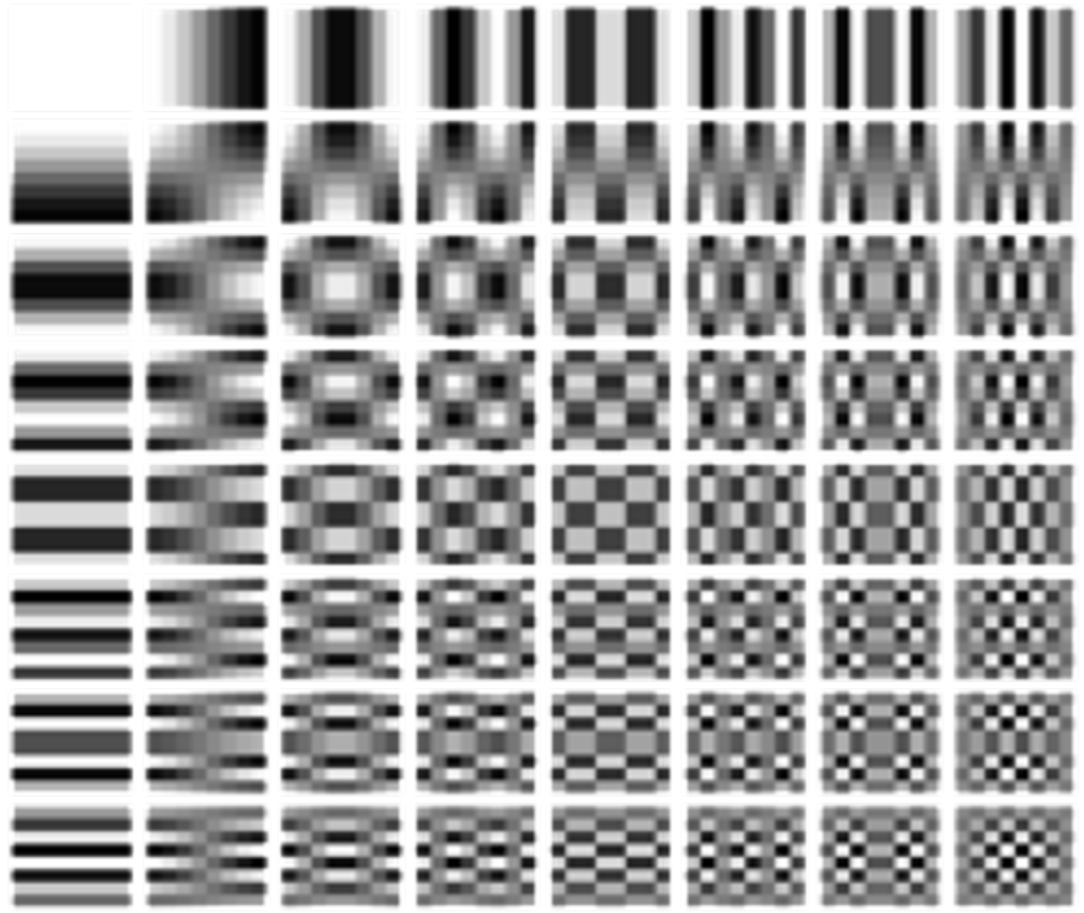
1-dimension

Image Compression

- Represent an image using less dimensions
- **Idea.** Do the following
  - Divide each image into 12×12 patches
  - Perform PCA and select top-k directions
  - Save the codes $(\mathbf{u}_1^\top \mathbf{x}, \ldots, \mathbf{u}_k^\top \mathbf{x})$ for each patch
- Note.
  - Interestingly, the eigenvectors look similar to discrete cosine transforms (DCTs), used in JPEG
  - Has some noise filtering effect



Eigenvectors

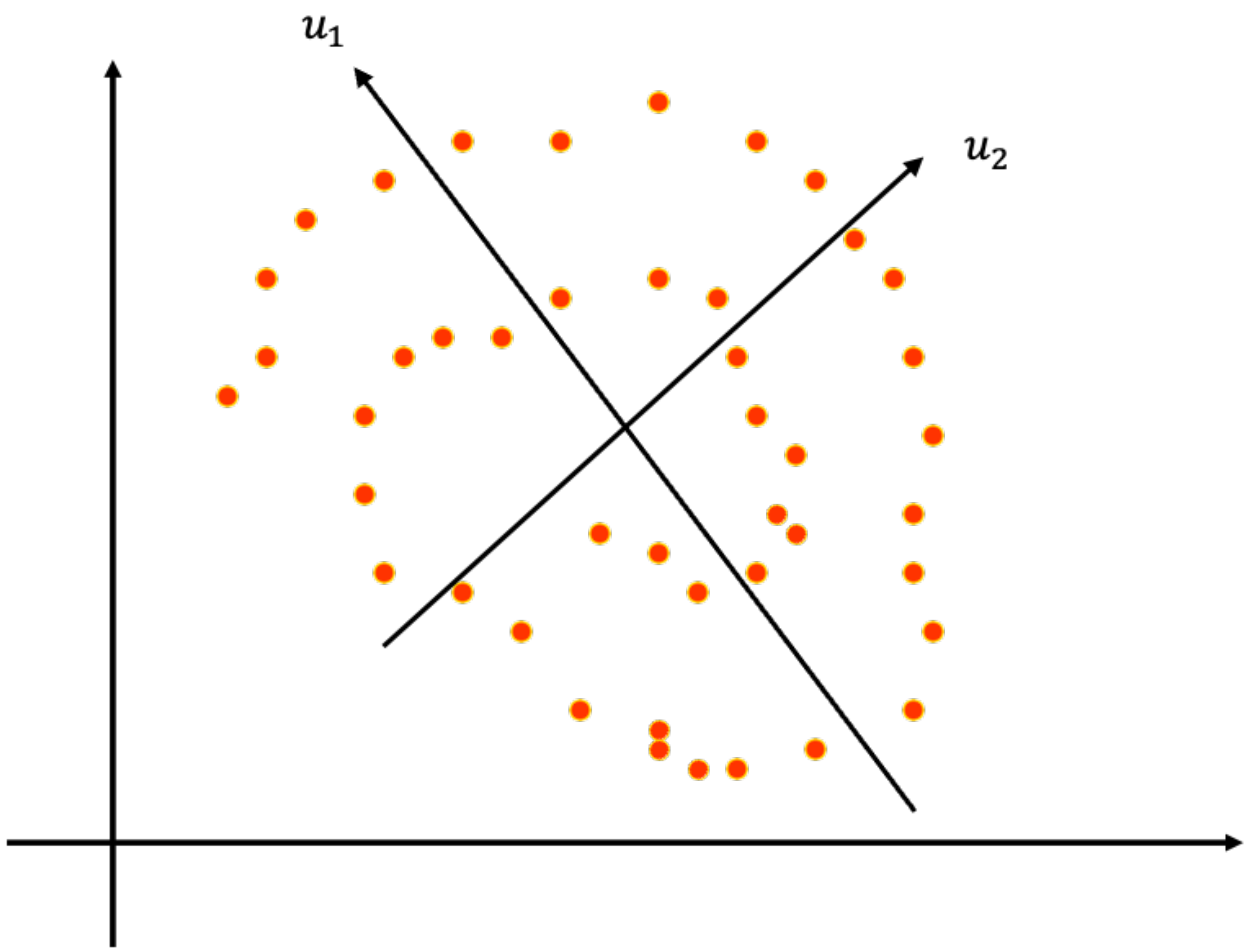


DCT bases

Limitations of PCA

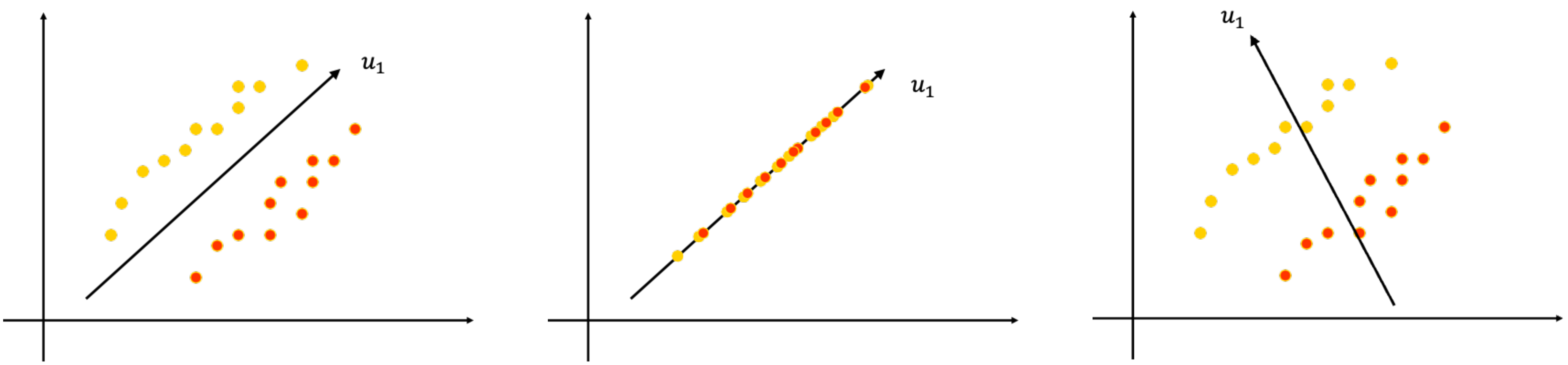
Failure modes

- Difficult to capture nonlinear datasets



Failure modes

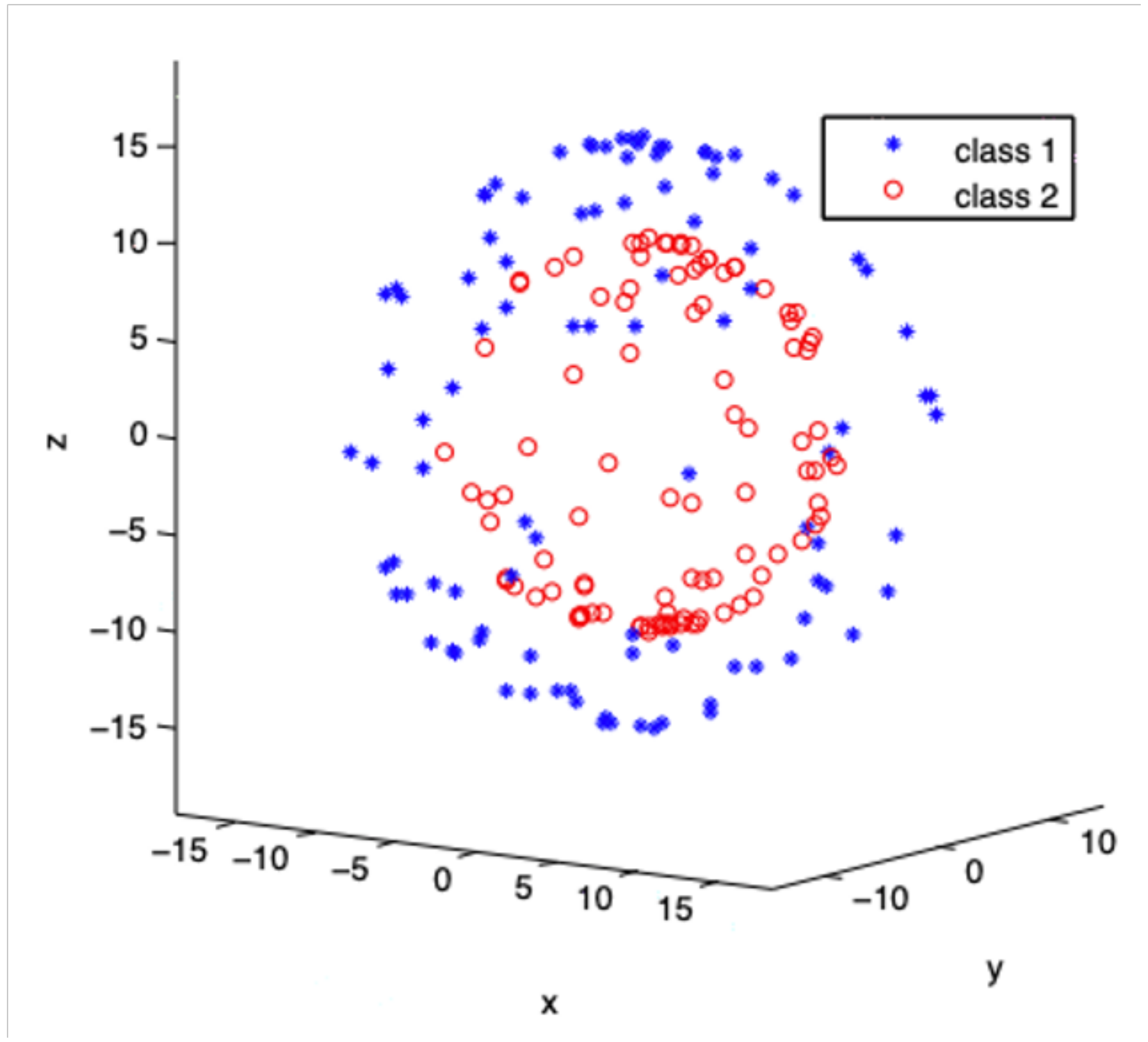
- Difficult to capture nonlinear datasets
- Does not account for class labels



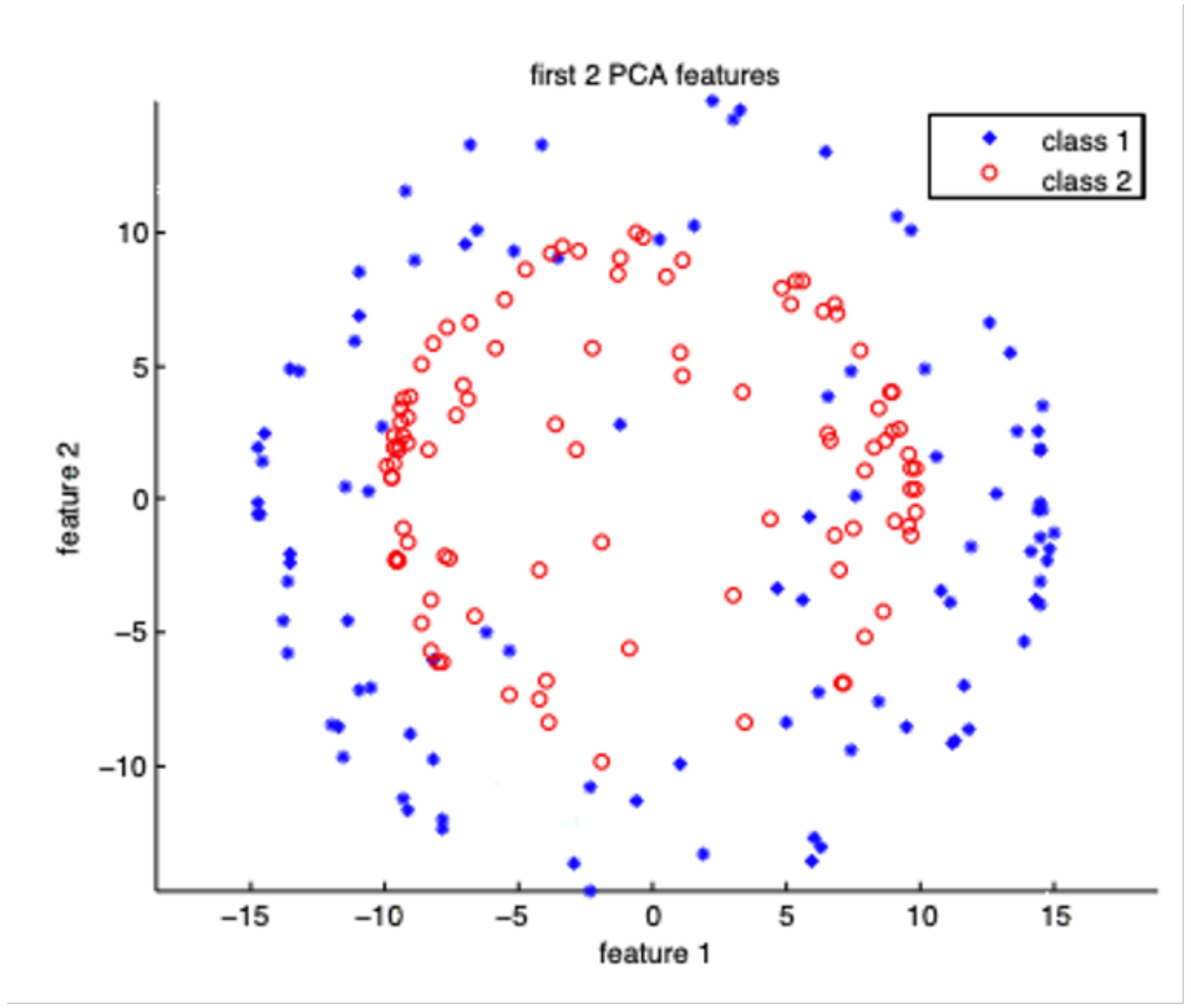
Advanced methods

Kernel PCA

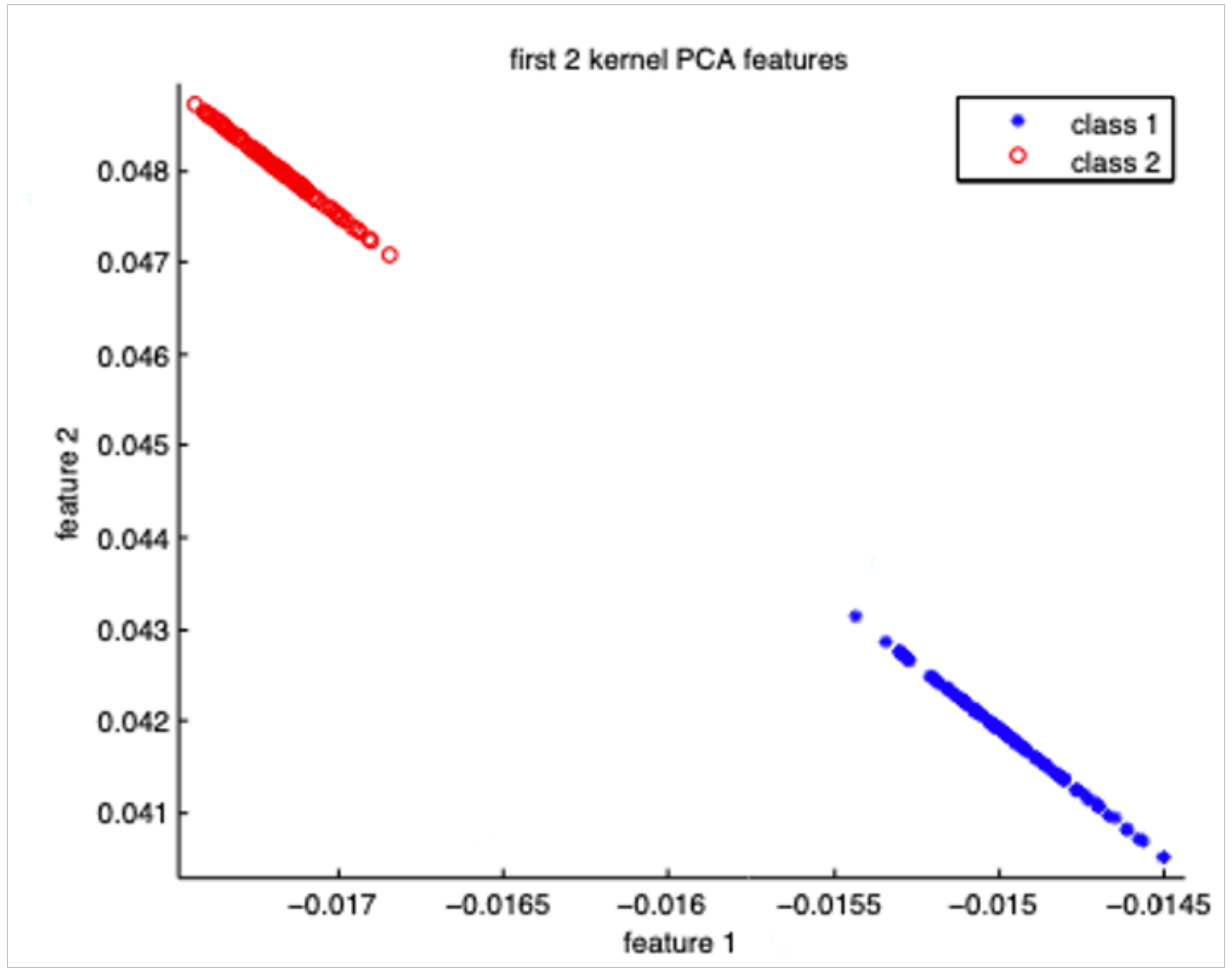
- **Idea.** Perform PCA for $\boldsymbol{\Phi}(\mathbf{x})$, not $\mathbf{x}$
  - Requires careful hyperparameter tuning & validation



Spherical Data



No Kernel

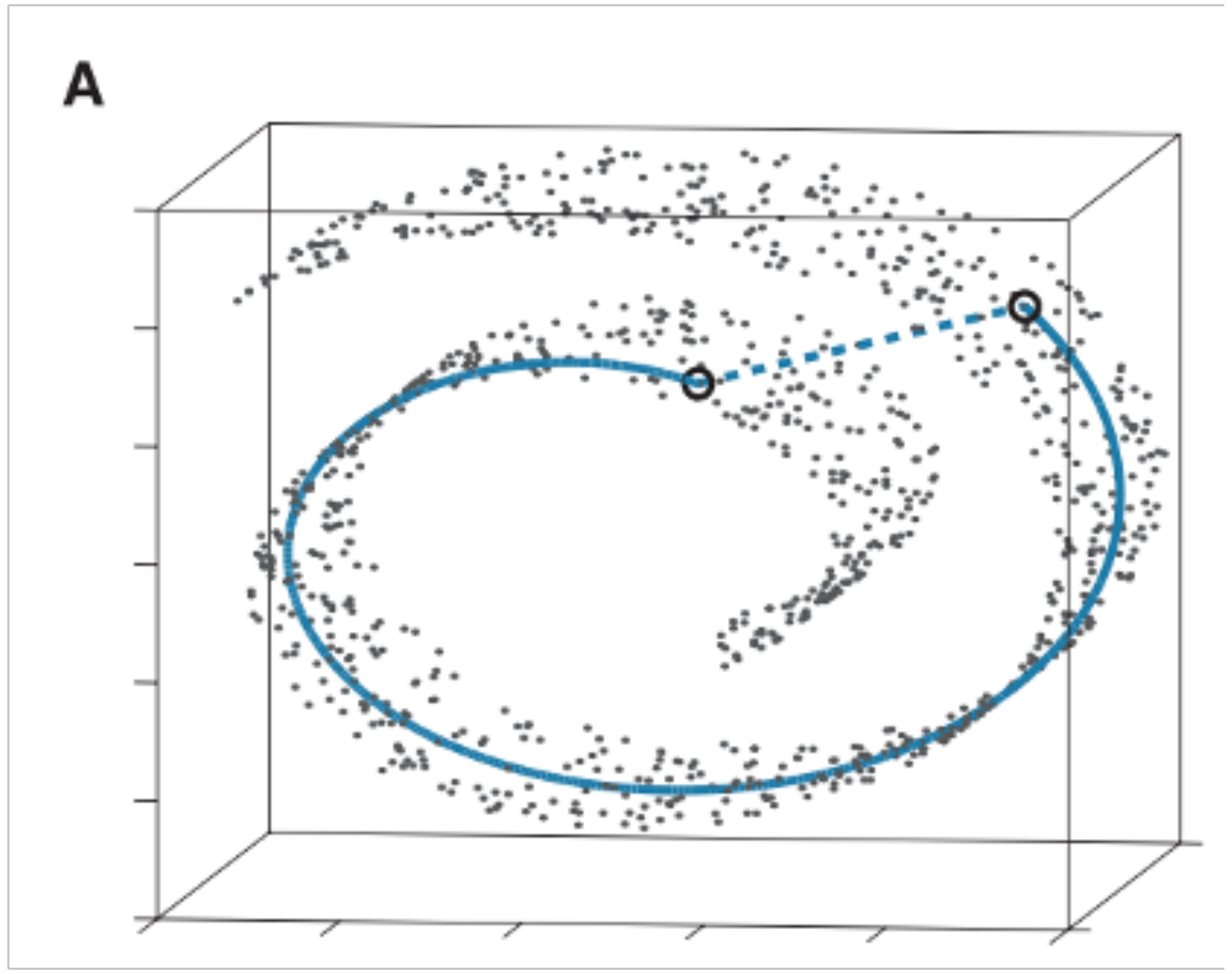


Gaussian Kernel ($\sigma = 20$)

Isomap

- **Goal.** Embed each data to low-dimensional space, so that

distance on the manifold = distance on the embedded space

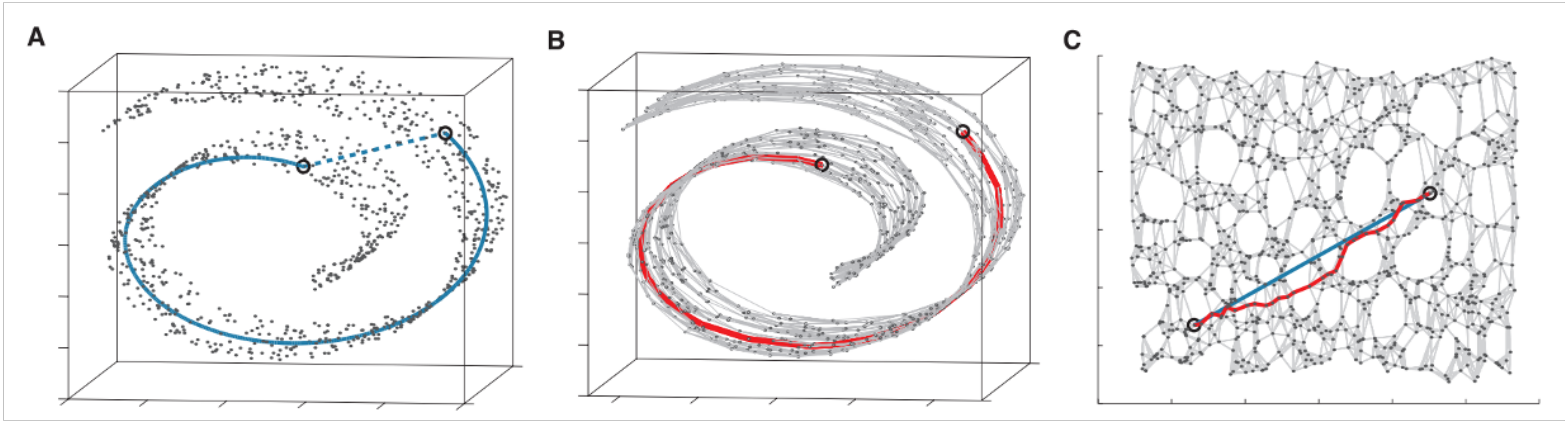


Isomap

- **Goal.** Embed each data to low-dimensional space, so that

  distance on the manifold = distance on the embedded space

- **Idea.** Build a graph of points, by connecting each point to k-nearest neighbors
  - Measure pairwise distance as the graph distance (use, e.g., Dijkstra's algorithm)



Isomap

- **Goal.** Embed each data to low-dimensional space, so that

  distance on the manifold = distance on the embedded space

- **Idea.** Build a graph of points, by connecting each point to k-nearest neighbors
  - Measure pairwise distance as the graph distance (use, e.g., Dijkstra's algorithm)
  - Then, use **MDS (multi-dimensional scaling)** to construct low-dimensional embedding
    - Rough idea. Translate pairwise distances $D \in \mathbb{R}^{n \times n}$ into something that looks like a sample covariance, via

$$-\frac{1}{2}HDH^\top, \qquad \text{where} \quad H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top \qquad \text{(called double centering)}$$

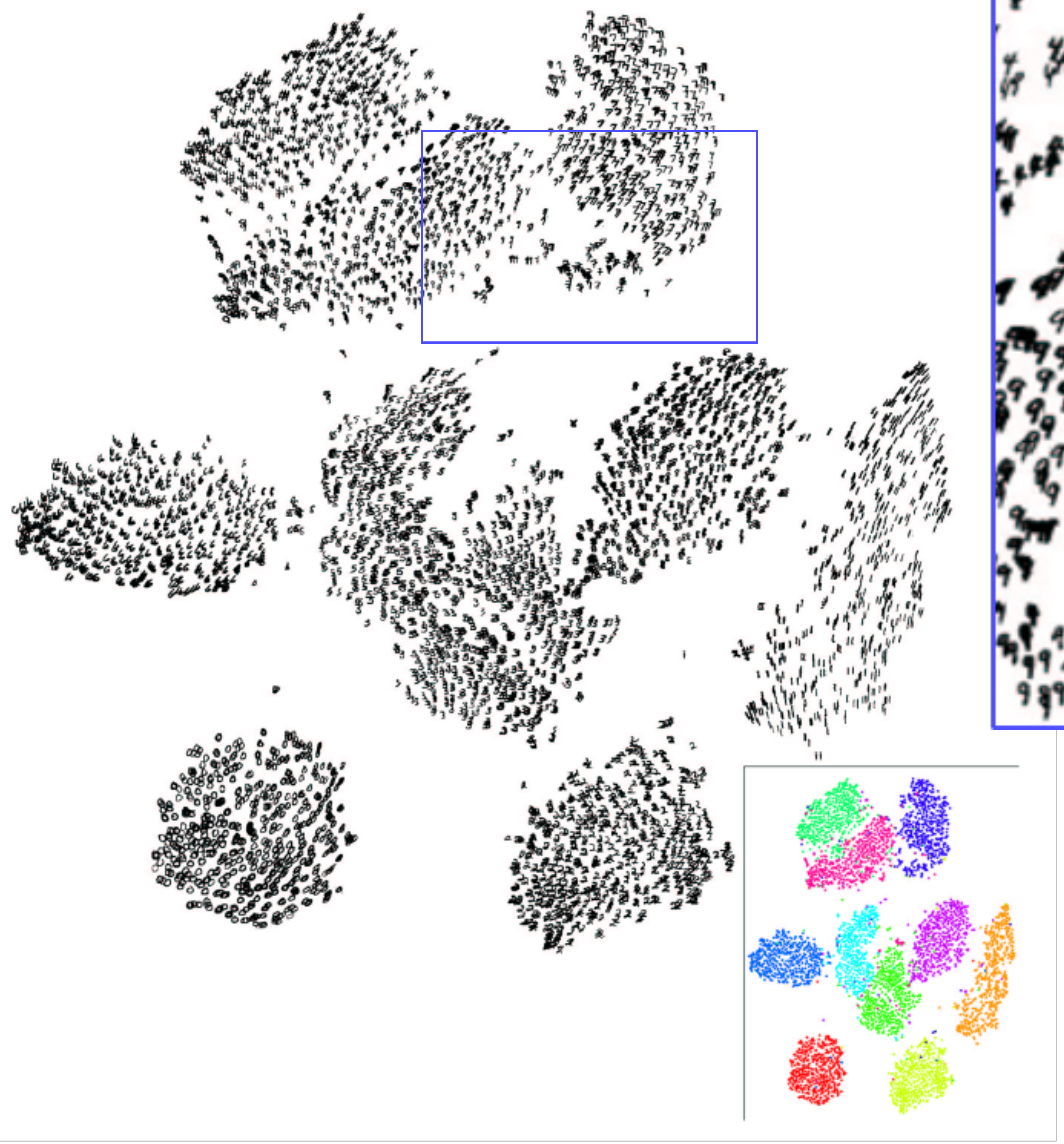
Then, perform PCA with it.

t-SNE

- Similar to Isomap, we preserve some distance
- **Idea.** Encode neighbor information as a probability distribution

$$p_i(j) = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)}{\sum_{k\neq i}\exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma^2)}$$

Then, find a low-dimensional embedding such that $\mathrm{dist}(p_i, p_j) \approx \mathrm{dist}(\mathbf{z}_i, \mathbf{z}_j)$



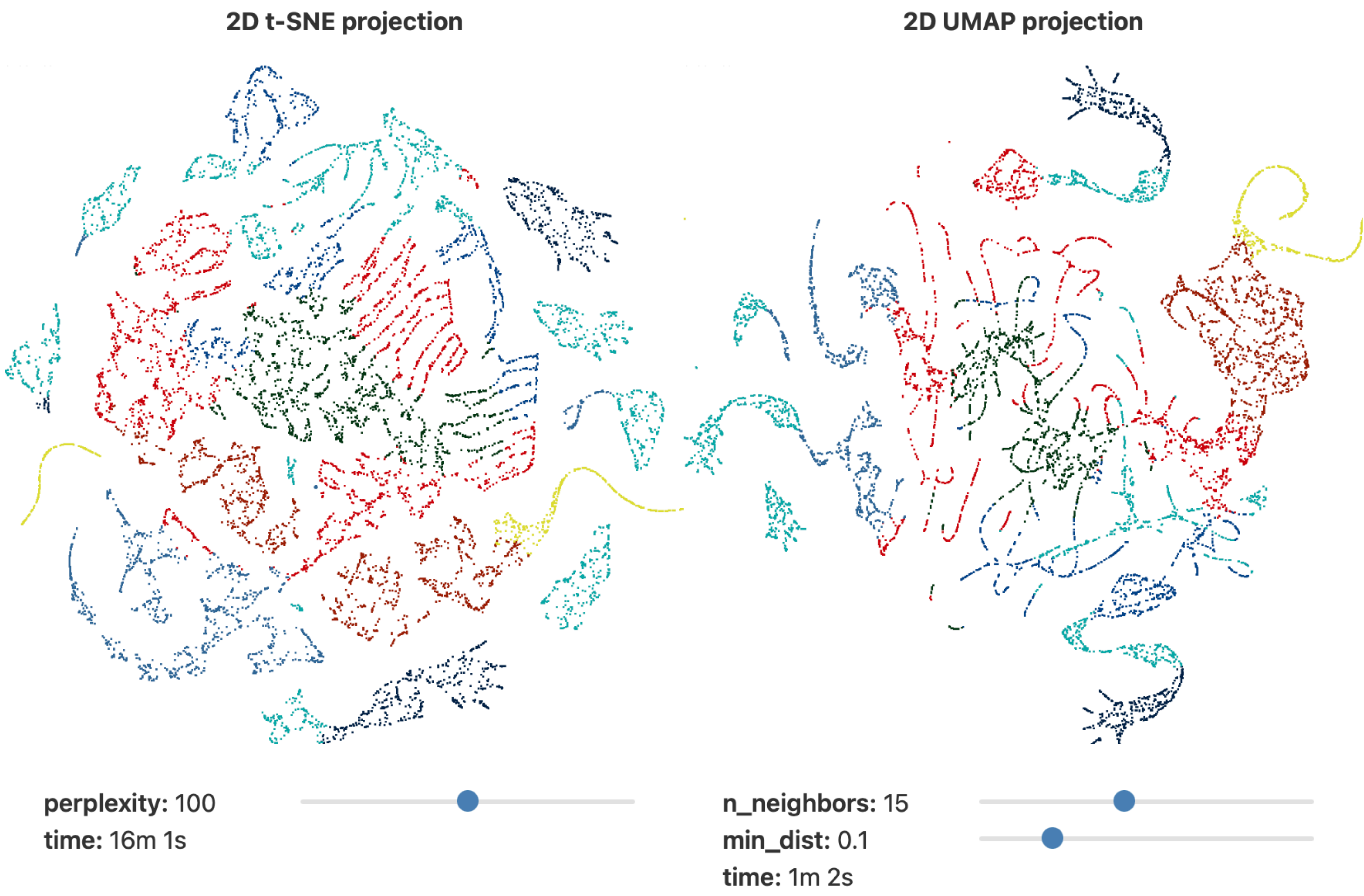
MNIST embeddings of t-SNE

(requires computing pairwise distances of 60,000 samples)


ICLR 2017–2024 submissions (n=24,347)
Coloured by year
Coloured by topic
2024
2017
offline reinf
multi-agent
imitation
reinforcement
causal
bandit
game
continual
meta-learning
physics
reasoning
language
question
recurrent
optimization
federated
vision-lang
word
autoencode
object-
privacy
bayesian
fair
attentio
audio
video
backdoor
network
3d
adversarial
diffusion
semi-sup
contrastive
generative adv
pruning
domain
time series
architecture
out-of-dis
protein
anomaly
distillation
graph
All ICLR submissions over eight years, scraped using the
OpenReview API. Titles+abstracts in TF-IDF representation,
SVD to 100 components, visualized with t-SNE. Papers are
assigned labels based on specific words present in their titles.
Dmitry Kobak, @hippopedoid

UMAP

- An elaborate and faster version of Isomap
  - Useful material: https://pair-code.github.io/understanding-umap/



Wrapping up

- **This week**
    - Dimensionality reduction
    - Principal component analysis
        - Basic maths on projection
        - PCA as variance maximization
        - PCA as distortion minimization
        - Applications and limitations
    - Modern versions

Cheers