UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Fall 2024)

Machine Learning

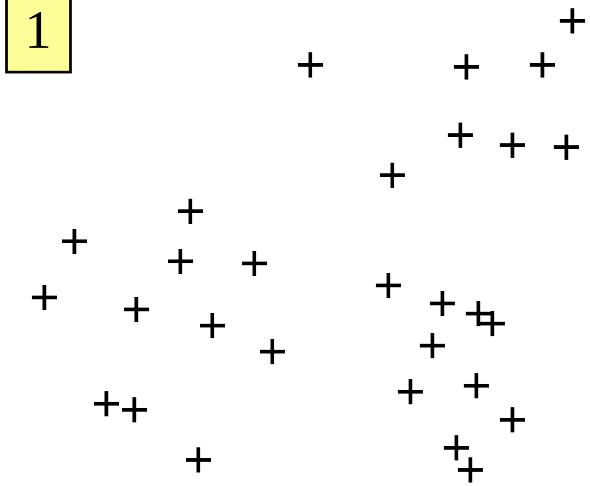
Lecture 19: Dimension Reduction

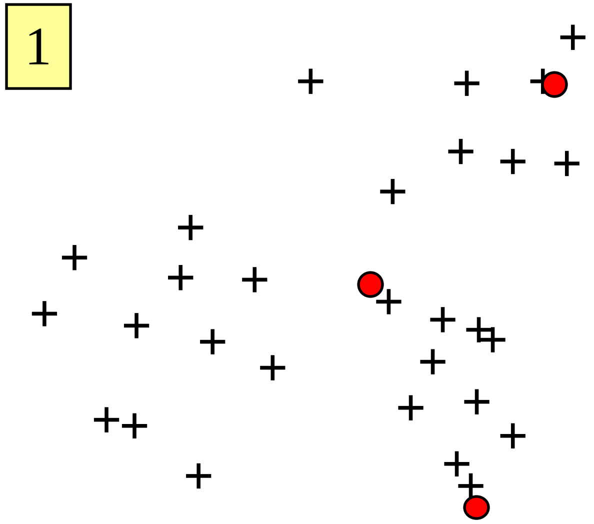
Chong Liu

Assistant Professor of Computer Science

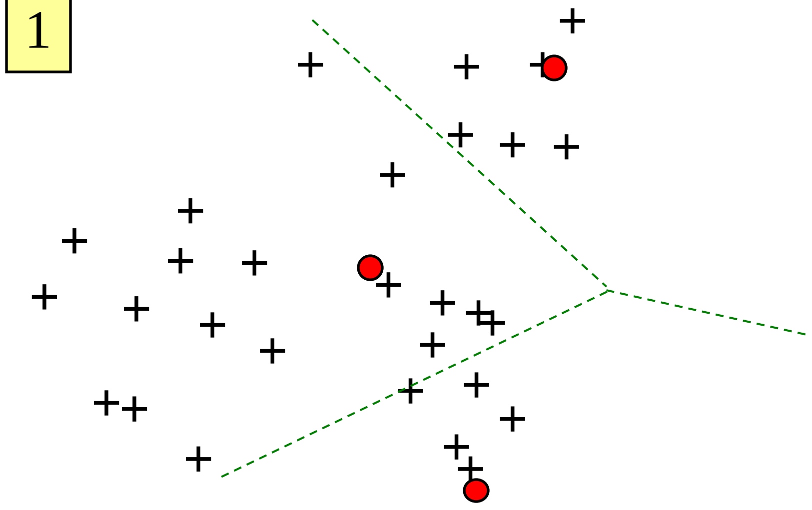
Nov 19, 2024

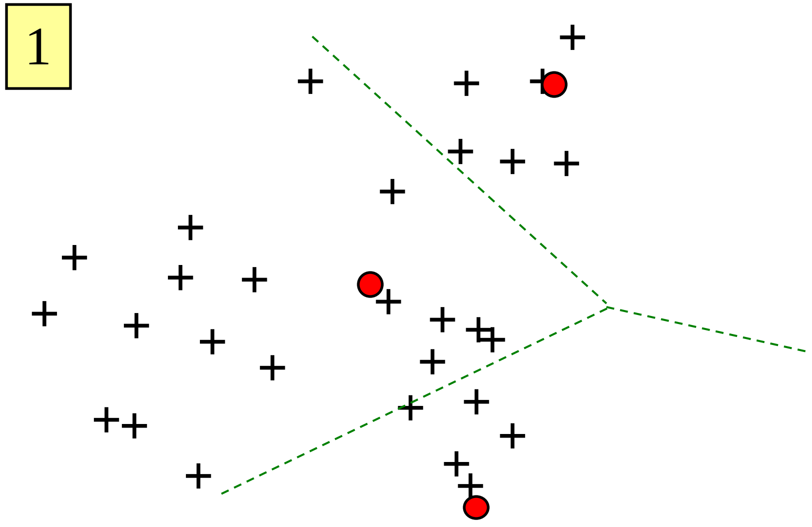
1

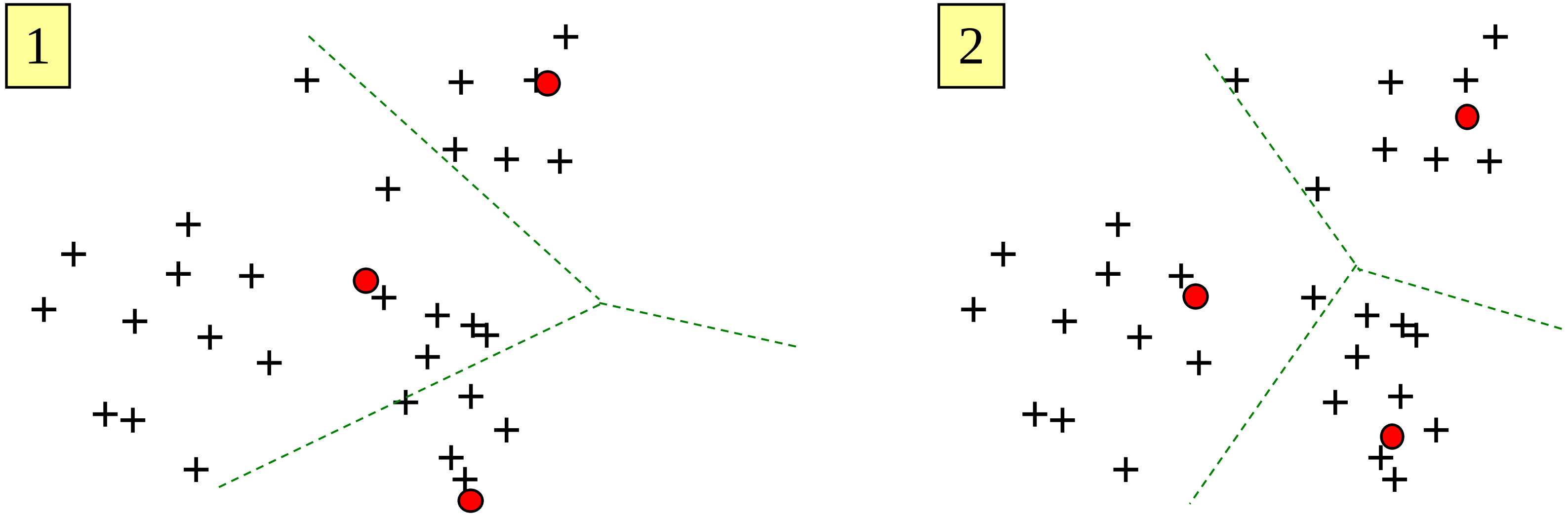



1

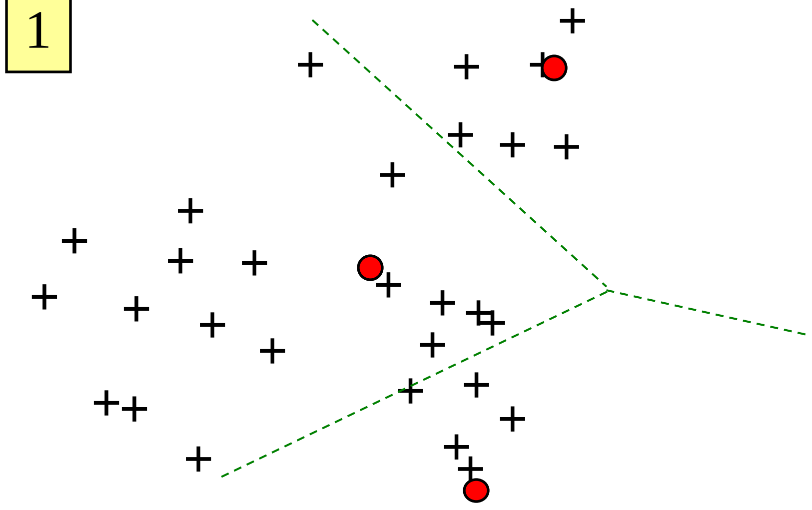
1



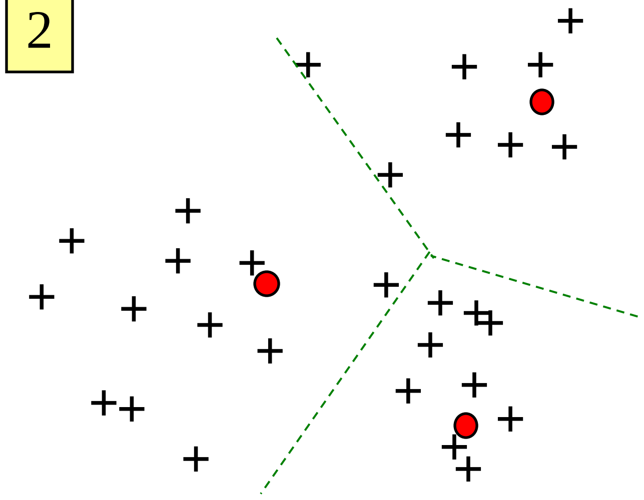

1
2


1
2

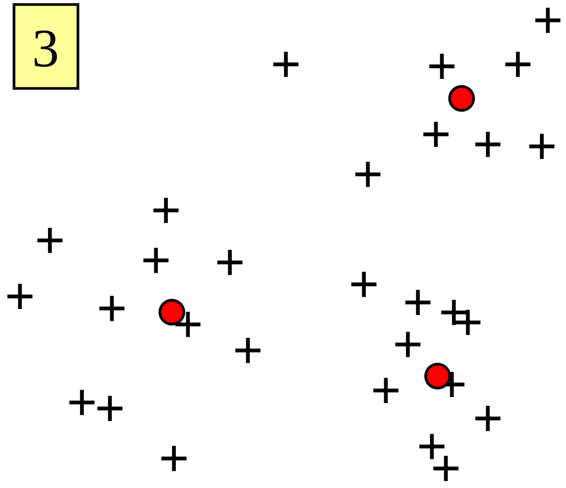
1



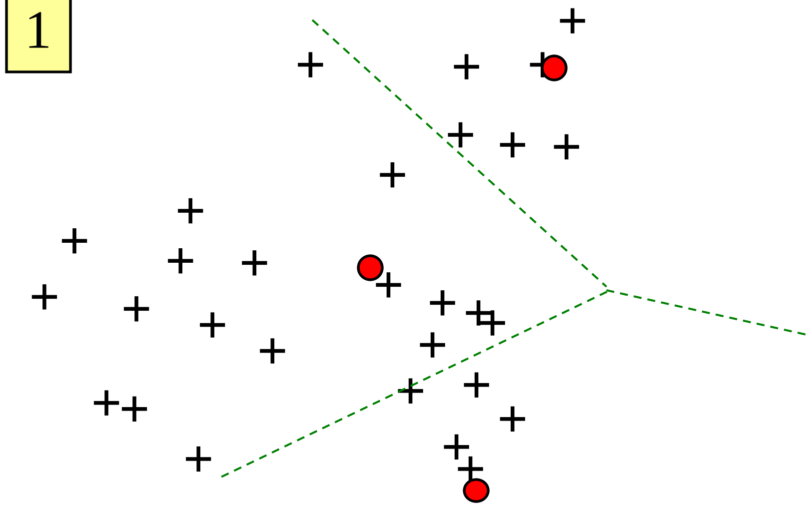
2



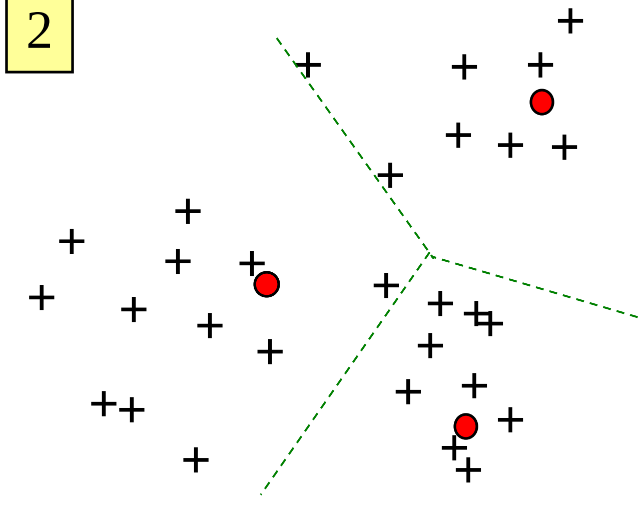
3



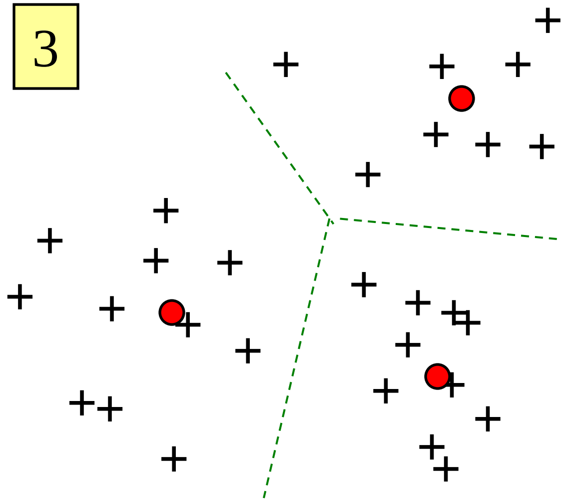
1

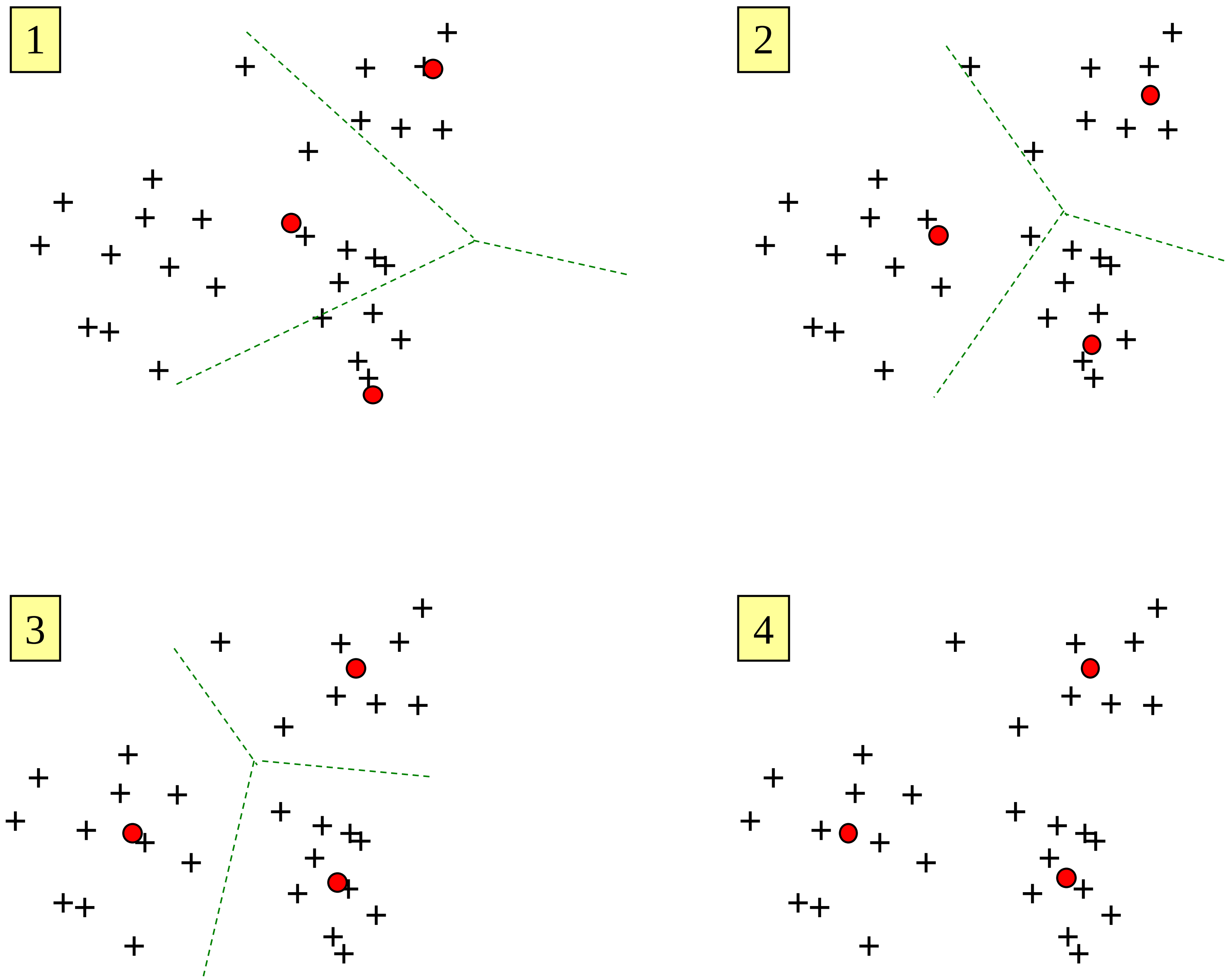


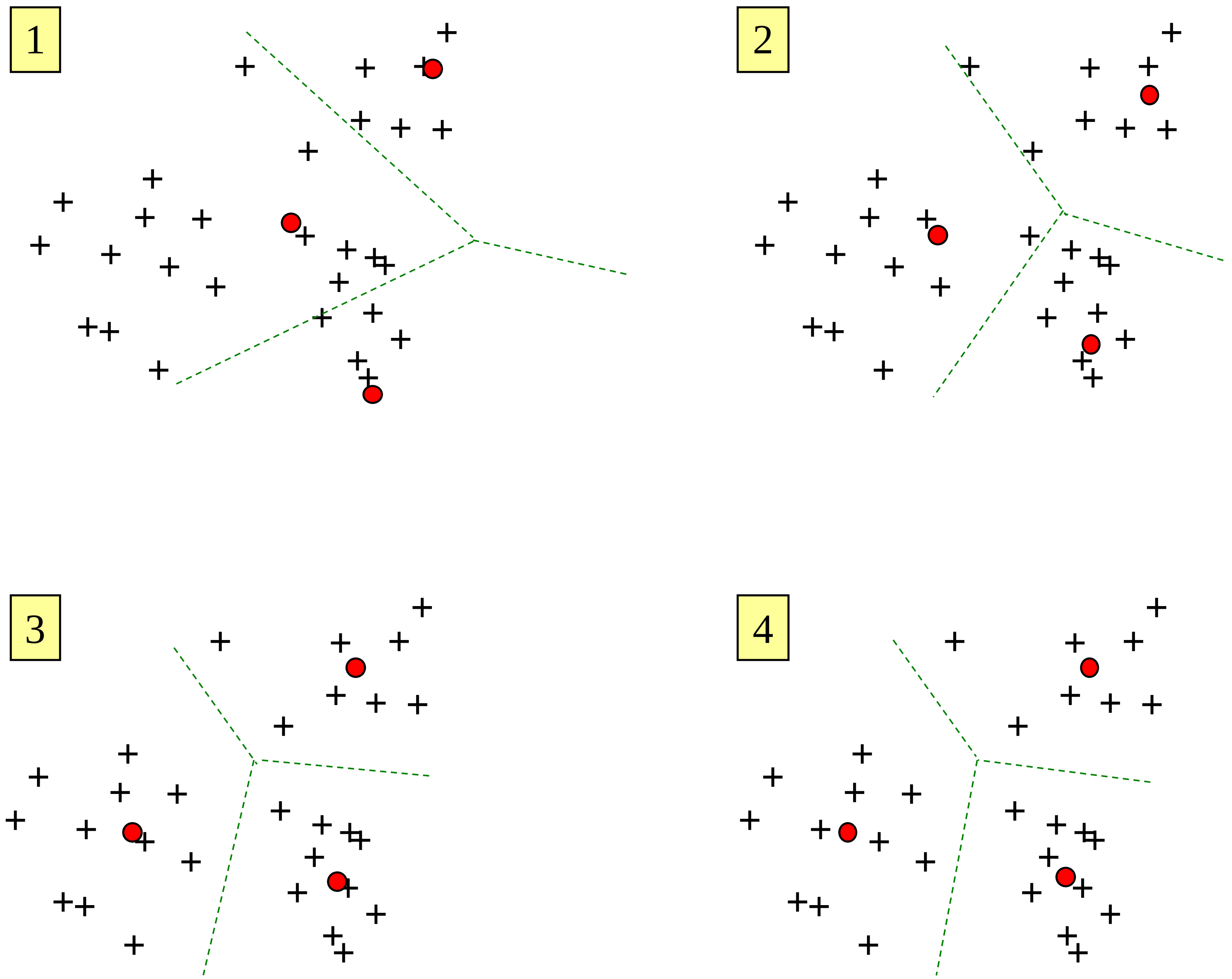
2

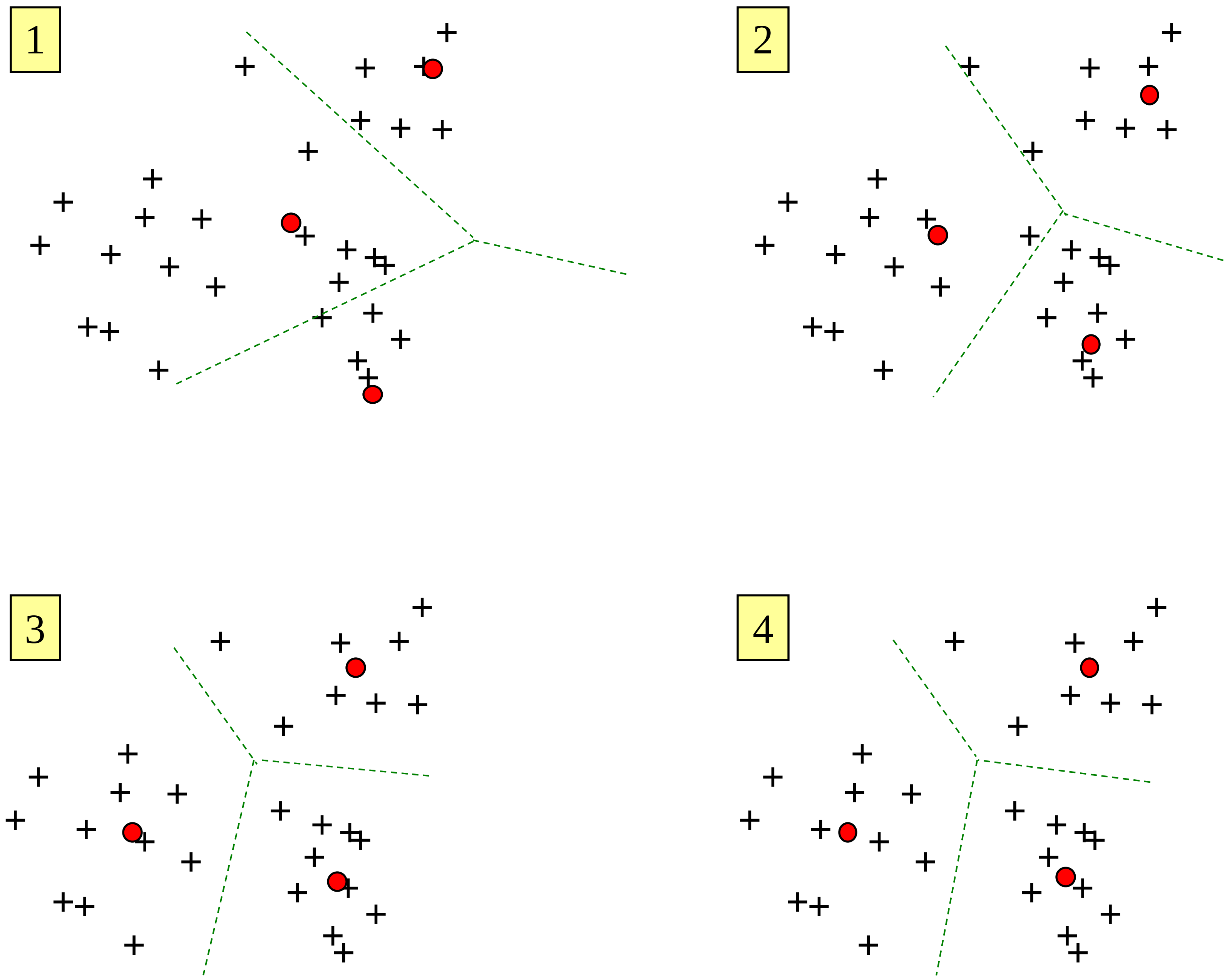


3




1
2
3
4


1
2
3
4



No change – finished!

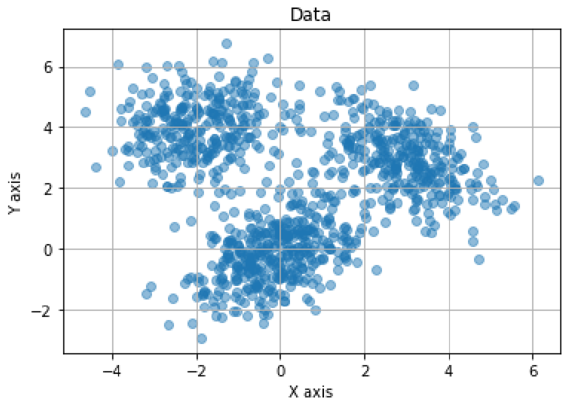
Recap: Clustering

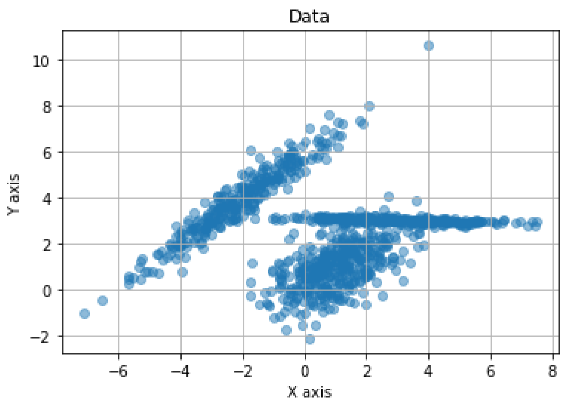
- K-means algorithm
  - Assign hard labels to data points
  - How does it work?
    - Alternating makes updates
    - Which distance function to use?
    - How many cluster centers (centroids) to choose?
    - How to initialize the centroids?

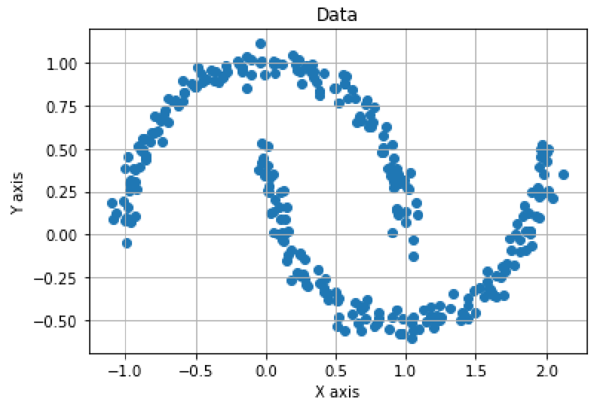
- Gaussian mixture models
  - Assign soft labels to data points
  - A probabilistic model for clustering

Recap: Two broad categories of unsupervised learning (1) Clustering **(2) Dimension reduction**

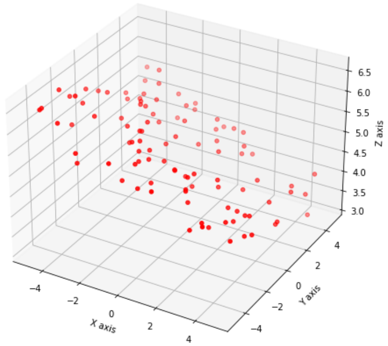
- Clustering aims at finding a partition of the data that makes sense.

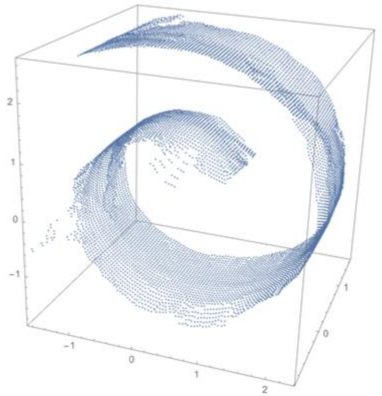






- Dimension reduction aims at identifying a more compact representation of data

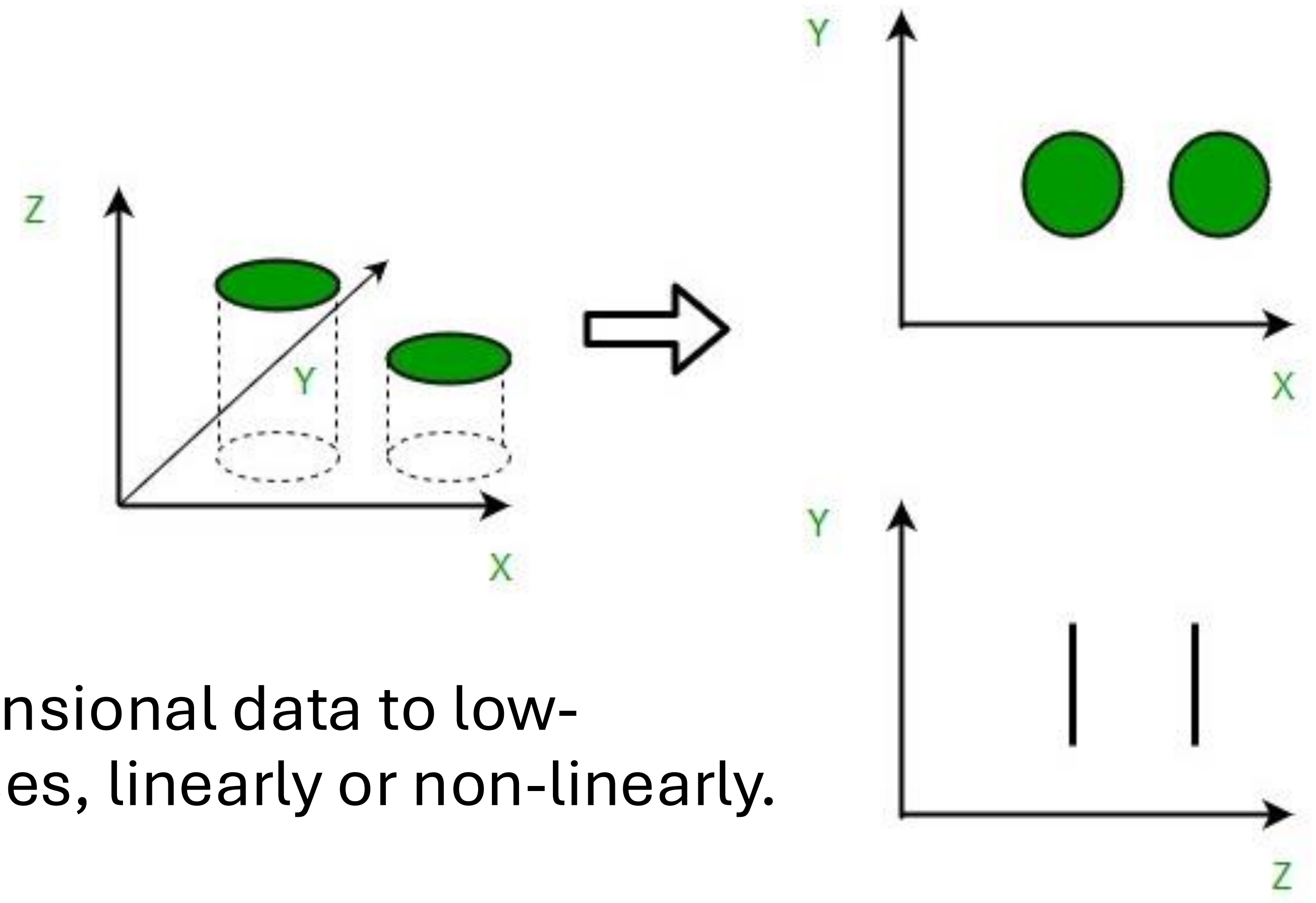




Today

- Why dimension reduction?
- Linear dimension reduction
    - Principal Component Analysis (PCA) algorithm
- Non-linear dimension reduction

What is dimension reduction?



- Project high-dimensional data to low-dimensional spaces, linearly or non-linearly.

Why dimension reduction

- Computation and memory efficiency
  - Reduce file size

- Statistical efficiency (fewer features to learn):
  - “Curse of dimensionality”

- Fewer features are easier to understand. It can help identifying hidden causes factors.

- Often data are high-dimensional but the physics mandate that they should be lying on a low-dimensional subspace.

Linear dimension reduction

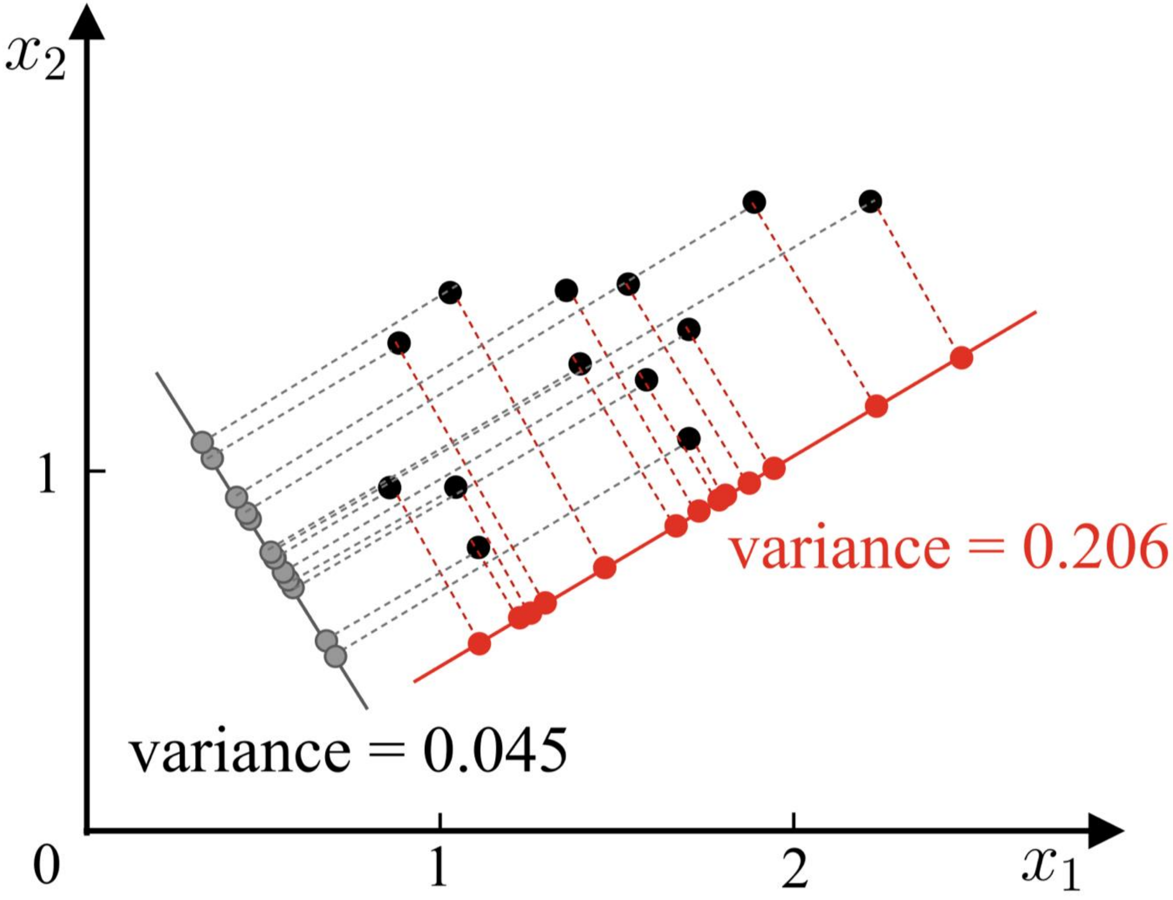
- Input: $X \in R^{d \times m}$
  - Number of data points: m
  - Number of features: d
- Output: $X' = WX \in R^{d' \times m}$
  - Projection matrix $W \in R^{d' \times d}$
  - Discussion: Does a random W works as a valid projection matrix?

- What is a good X'?
  - As long as it makes sense to your task
  - A good low-dimensional representation that is good for further process (e.g., classification)

Example: from 2-d to 1-d

- Which project direction should we choose?
  - Red, gray, or ?

- Max-variance is the best choice!
  - Data is distributed sparsely
  - Easier for further process

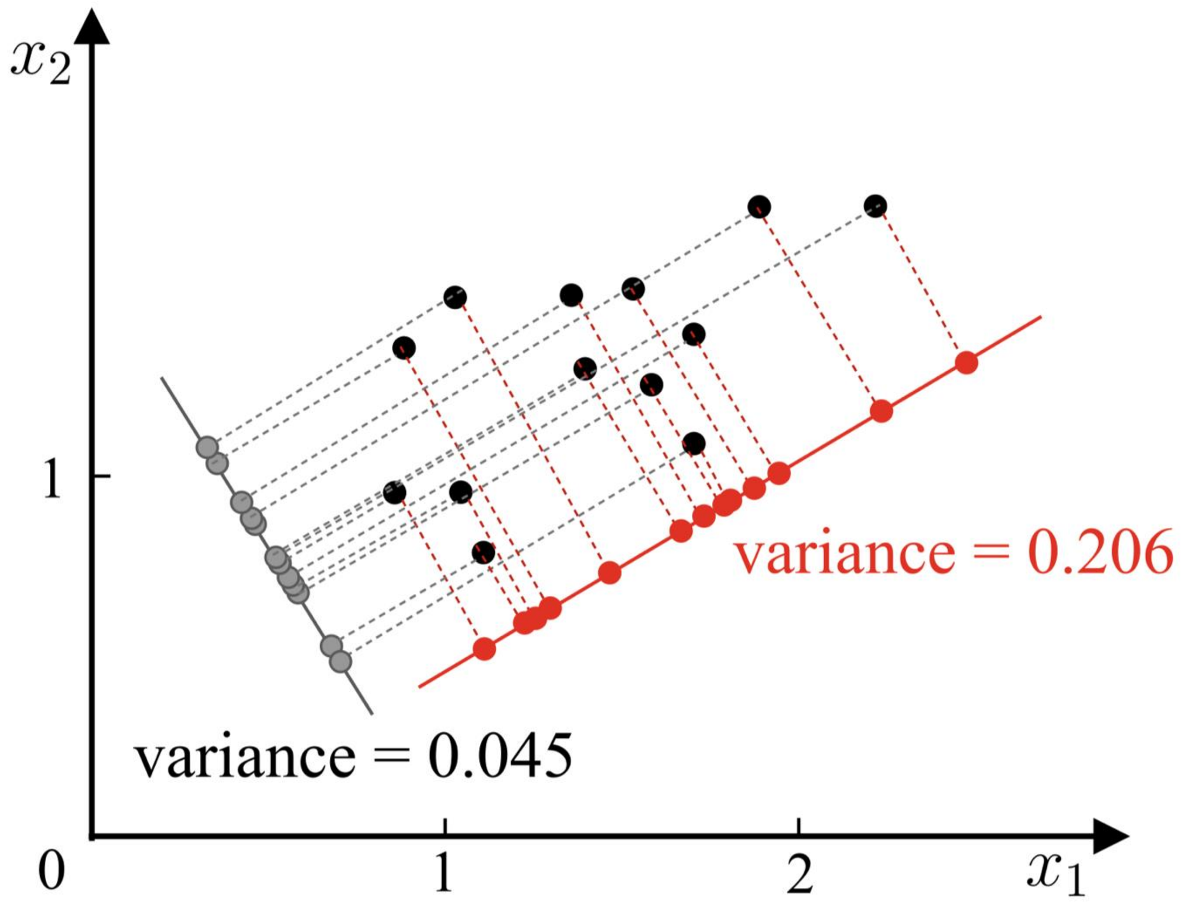


Key idea of Principal Component Analysis (PCA)

- These two are equivalent:

  - Maximum variance: the projections of samples onto the hyperplane should stay away from each other.

  - Minimum reconstruction error: the samples should have short distances to this hyperplane.



Principal Component Analysis (PCA)

Input: Data set $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$;
Dimension d' of the lower dimensional space.

Process:

1: Center all samples: $\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$; 0-mean samples

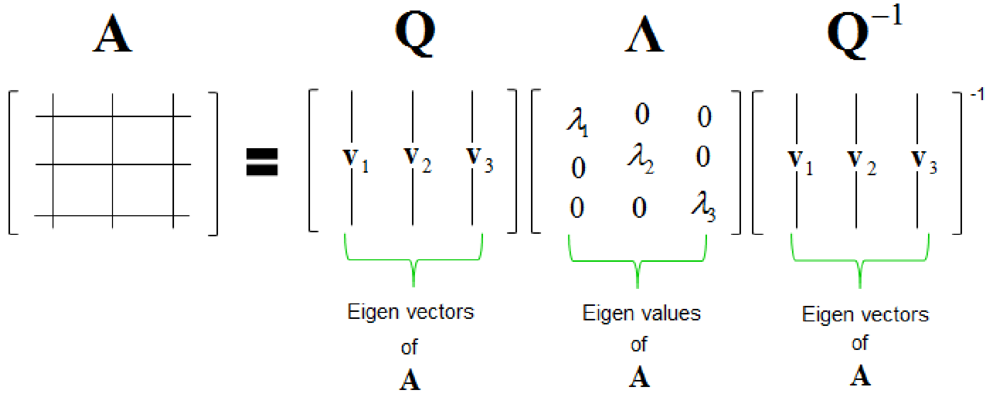
2: Compute the covariance matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$ of samples;

3: Perform eigenvalue decomposition on the covariance matrix $\mathbf{X}\mathbf{X}^{\mathrm{T}}$;

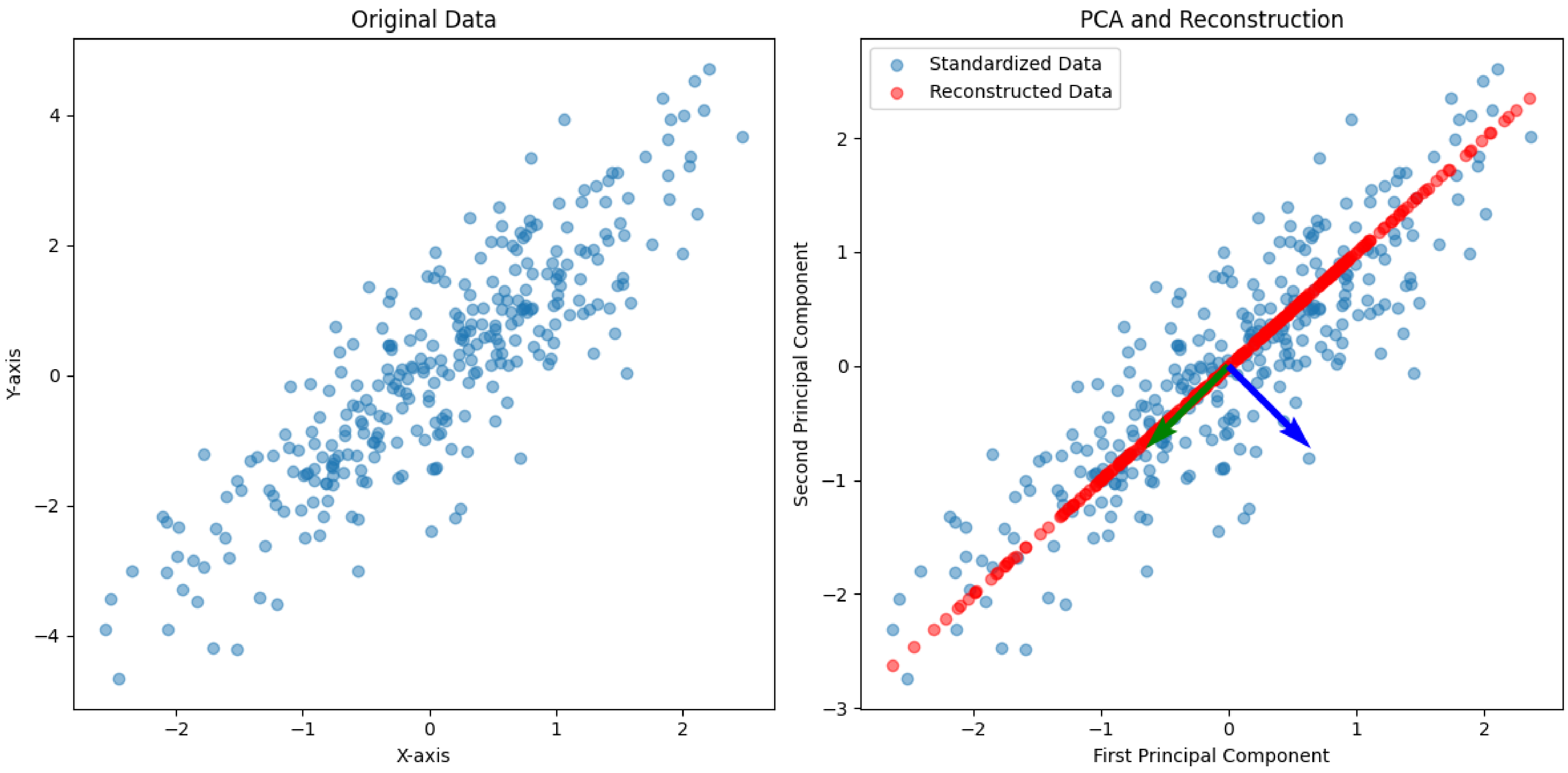
4: Take the eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{d'}$ corresponding to the d' largest eigenvalues.

Output: The projection matrix $\mathbf{W}^* = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{d'})$.

Eigenvalue decomposition



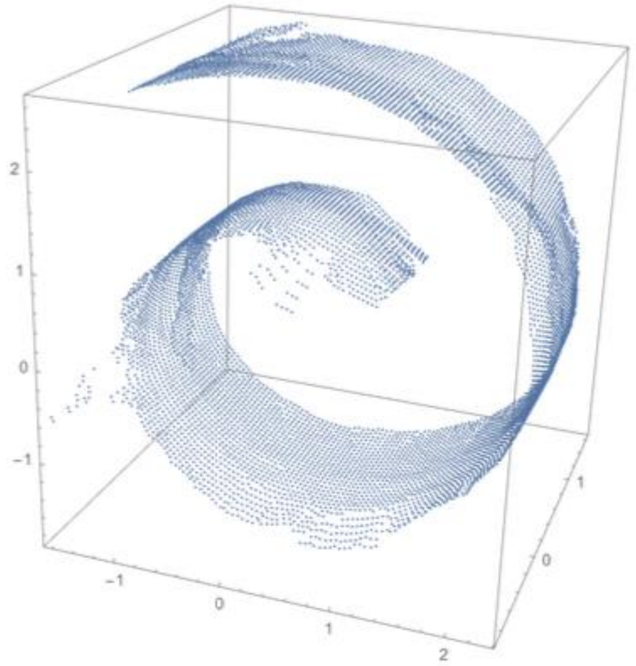
Example of PCA on Gaussian data



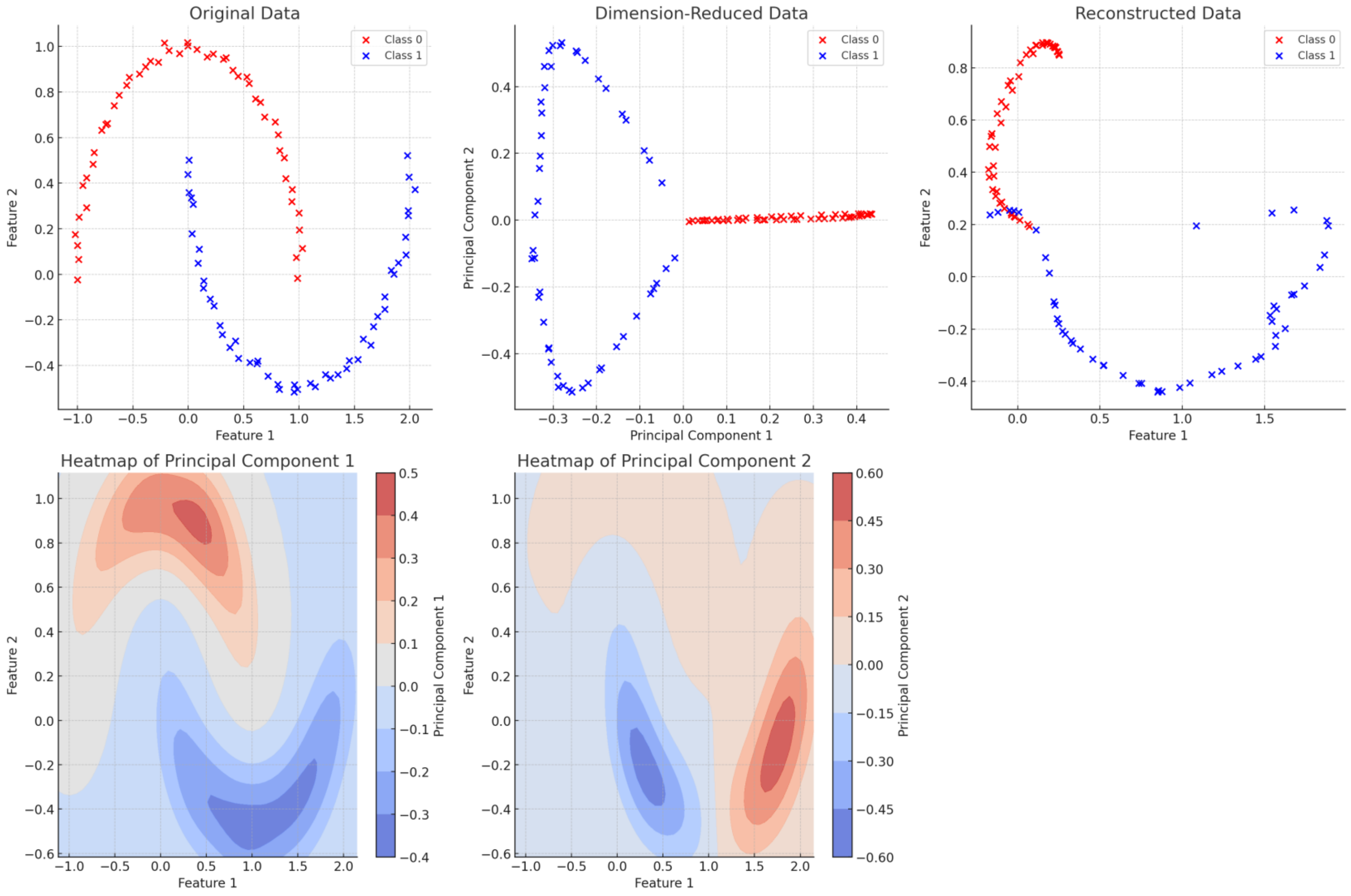
Different choices of dimension of PCA in image compression

Non-linear dimension reduction

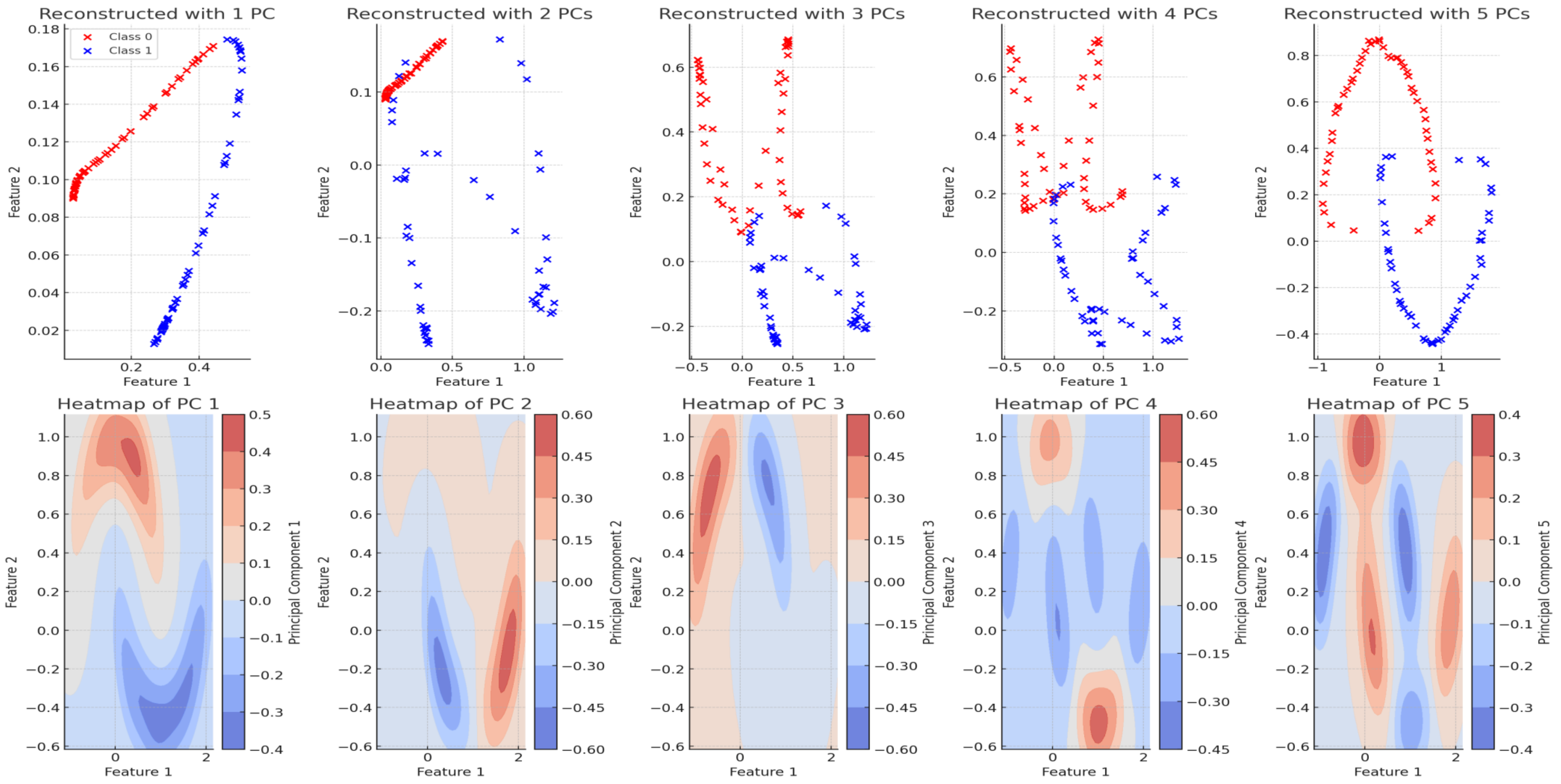
- Mixture of linear subspaces:
  - Subspace clustering
  - Mixture of probabilistic PCAs
  - A combination clustering and dim-reduction
- Kernel PCA
  - Run PCA on the kernel matrix instead of the covariance matrix
- Laplacian Eigenmaps (also the related Isomap)
  - First construct a nearest neighbor graph
  - Then run SVD on the Laplacian matrix of the graph
- Neural approaches:
  - Autoencoders / variational autoencoders
  - Transformers (for data-reconstruction)



Kernel PCA on the two-moon example



Increasing the number of principal components in kernel PCA improves the reconstruction



What's next?

- Thu Nov 21: Advanced Topic: Decision Making
  - Not part of the final exam
  - Most recent hot topics in machine learning!
  - It would be fun!

- Tue Nov 26: Course Review
- Tue Dec 3: Final project presentation
- Thu Dec 5: HW3 and HW4 Review
- Mon Dec 16, 8:30-10am: Final exam