Lecture 24: Non-linear dimensionality reduction techniques

Reading: ESL 14.5.4, 14.8, 14.9

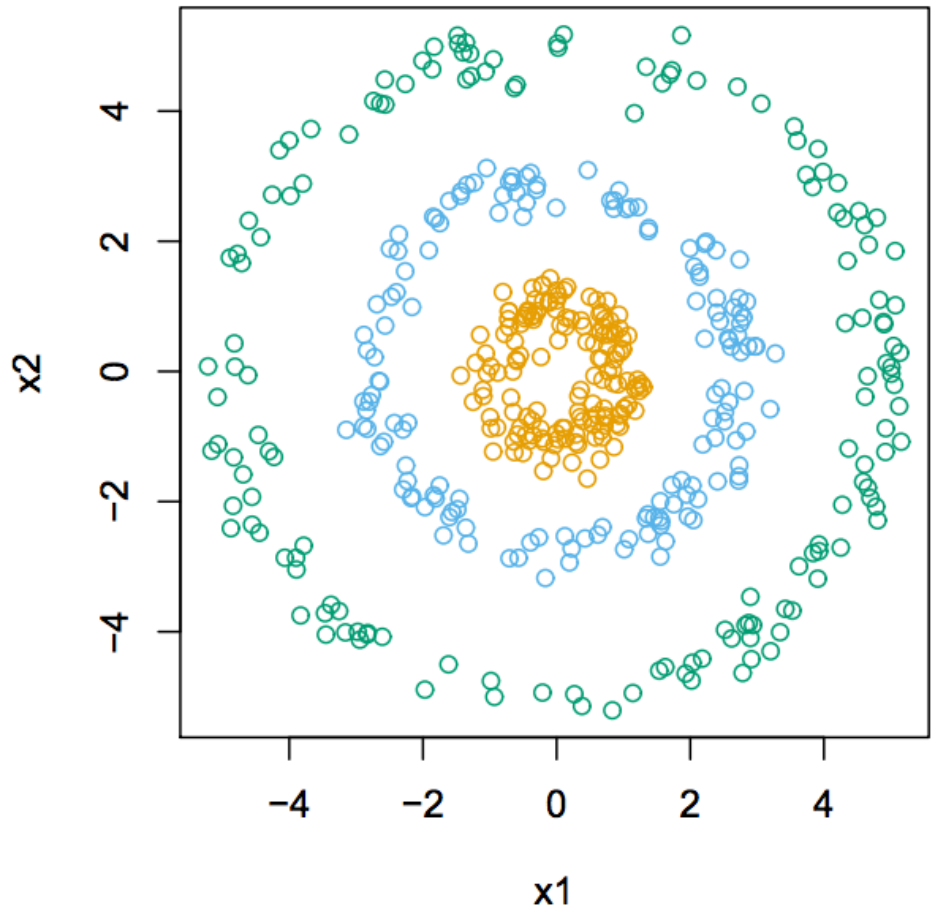
STATS 202: Data mining and analysis

November 18, 2019

Overview

- Methods for unsupervised learning or exploratory data analysis.
- PCA is a linear dimensionality reduction method: it finds linear combinations of input variables that best summarize input data.
- If the data show very non-linear patterns, these will be difficult to discover by PCA.
- Non-linear dimensionality reduction methods are useful to analyze data with a high signal to noise ratio, for example, images of physical objects.

Example. Shells



All directions have equal variance:
PCA wouldn't capture the obvious circular patterns.

Kernel PCA

- To make PCA non-linear, we transform the features through Φ.
- The feature map Φ gives rise to the kernel $\langle \Phi(x_i), \Phi(x_k) \rangle$.
- **Kernel PCA**:
  1. Find the vector g_1, in the expanded feature space, which maximizes the variance of the projections:

     $$\langle \Phi(x_1), g_1 \rangle, \langle \Phi(x_2), g_1 \rangle, \ldots, \langle \Phi(x_n), g_1 \rangle$$

  2. Find the vector g_2, orthogonal to g_1, which maximizes the variance of the projections:

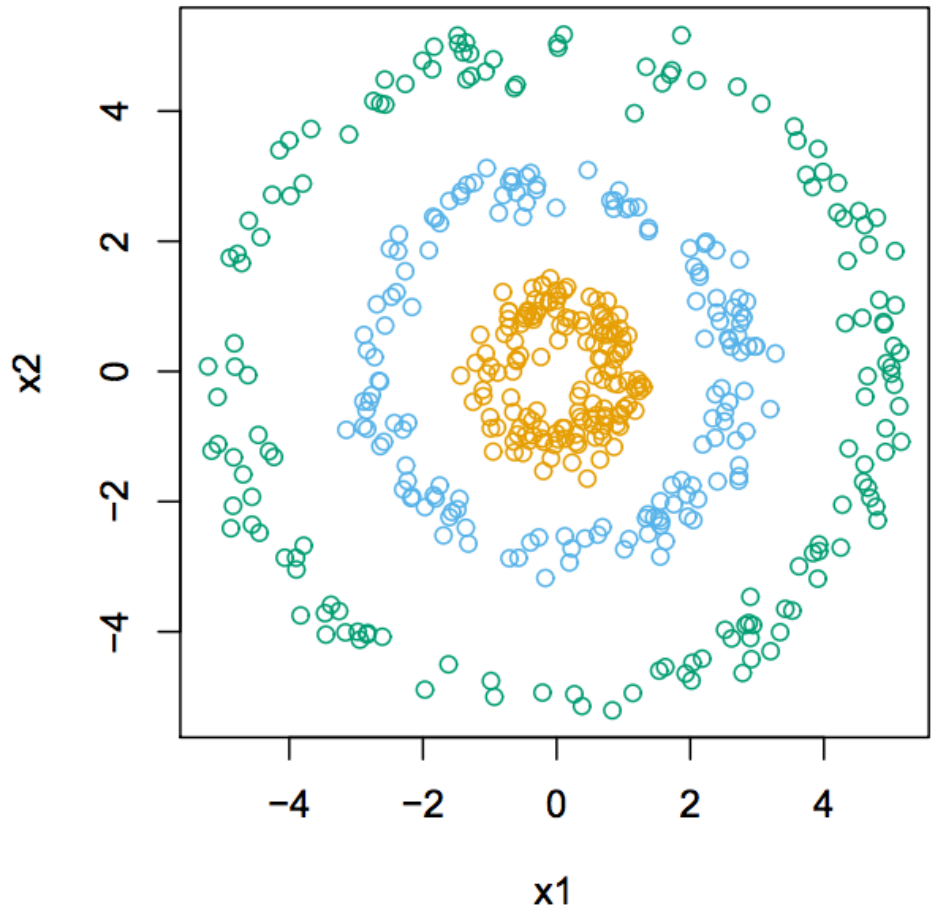
     $$\langle \Phi(x_1), g_2 \rangle, \langle \Phi(x_2), g_2 \rangle, \ldots, \langle \Phi(x_n), g_2 \rangle$$

  3. ...

Kernel PCA

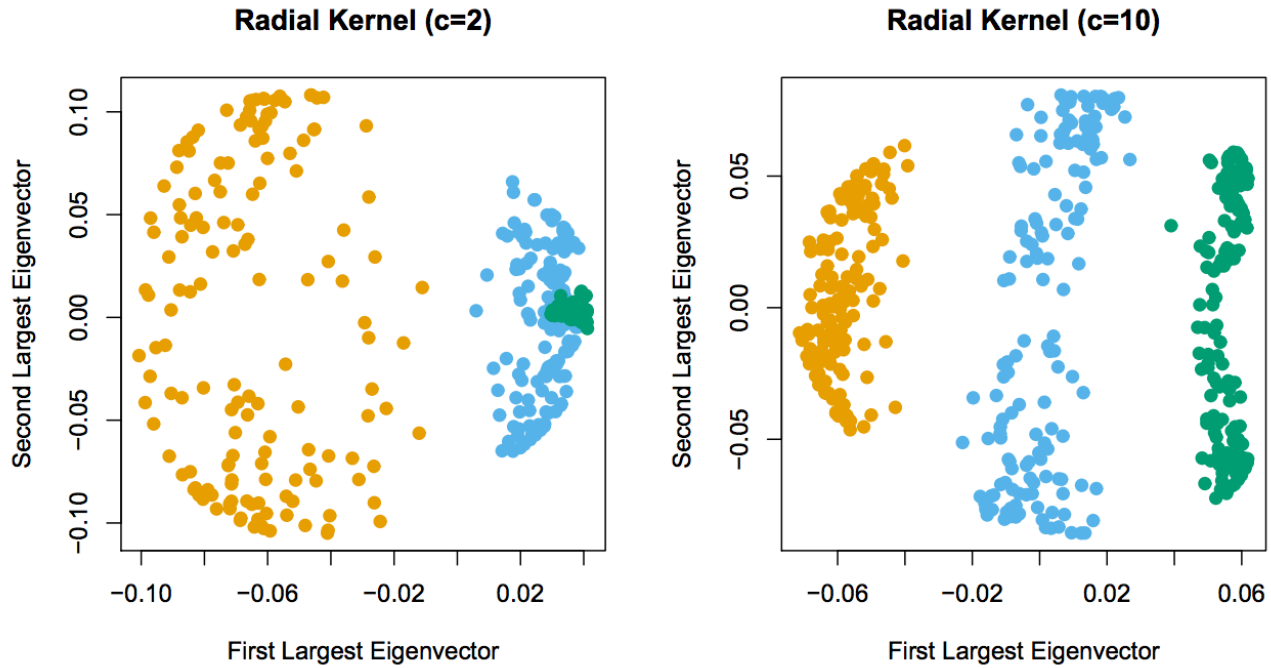
- Recall **regular PCA**: Find SVD of $X = U\Sigma V^T$.
  Then the PC loadings are given by V and the PC scores are $U\Sigma$.
- To get the PC scores $U\Sigma$ find the eigendecomposition of $XX^T = U\Sigma^2 U^T$.
- This eigendecomposition only depends on inner products: $(XX^T)_{ik} = \langle x_i, x_k \rangle$
- We can replace this with a kernel matrix $K(i,k) = \langle \Phi(x_i), \Phi(x_k) \rangle$.
- This frees us to use feature maps that are too expensive to instantiate explicitly or are unknown.

Example. Shells



All directions have equal variance:
PCA wouldn't capture the obvious circular patterns.

Example. Shells



The 1st principal component using the RBF kernel with $c = 1/\gamma = 10$ captures the distance from the center and clearly separates the three clusters.

How to choose the right kernel?

- In Kernel PCA, we have to choose the right kernel to obtain a meaningful visualization or summary.
- This choice is not always easy.
- There are methods which use the data to learn the right kernel.
- Several non-linear dimensionality reduction methods can be viewed as kernel PCA with kernels learned from the data; see "A kernel view of the dimensionality reduction of manifolds" for an in-depth discussion.
- These methods exploit the local structure of the data (similarity is only meaningful among nearest neighbors).
- We will talk about two examples:
  1. Locally linear embeddings
  2. Isomap

Locally linear embeddings (LLE)

Idea:

1. Represent each sample as a linear combination of neighbors:

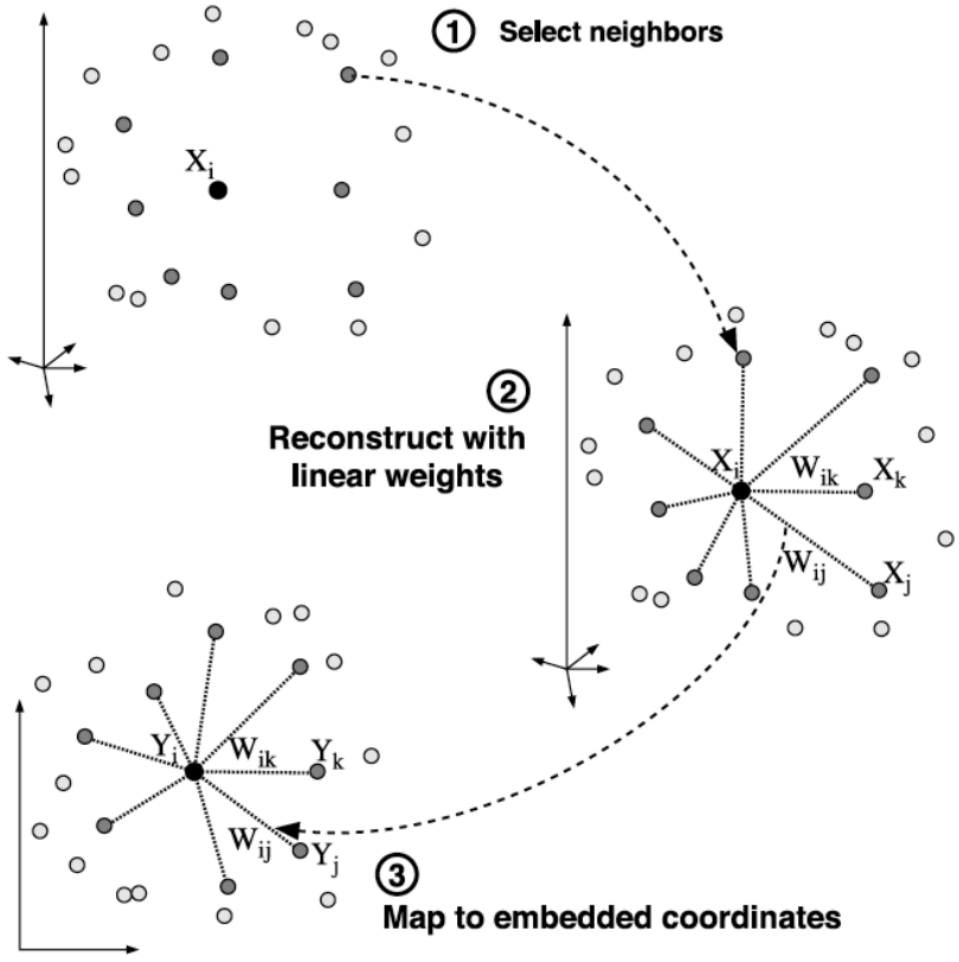
$$x_i \approx \sum_{k=1}^{n} x_k W_{ik}, \qquad W_{ik} \neq 0 \iff x_i, x_k \text{ are neighbors}$$

2. Map each sample x_i to a point z_i in low-dimensional space (usual 2 or 3 dimensional), such that the local linear representation holds approximately:

$$z_i \approx \sum_{k=1}^{n} z_k W_{ik}.$$

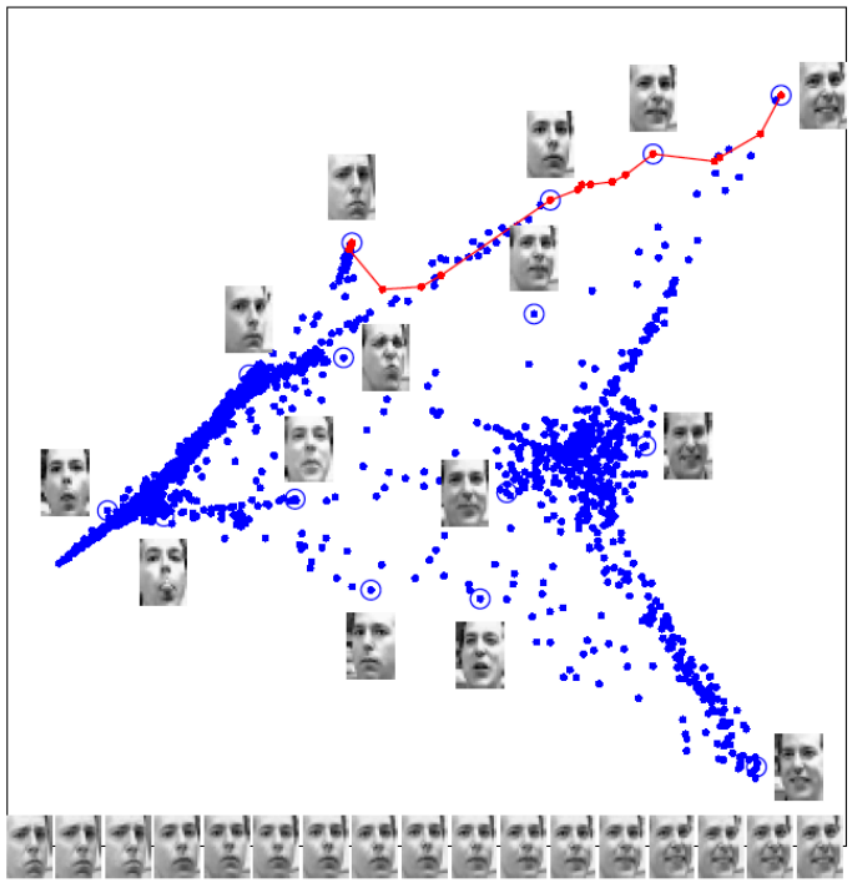
- In step 1, we find the weights W.
- In step 2, we fix W and find the optimal low-dimensional embedding.
- The second problem is solved by an eigendecomposition.

Locally linear embeddings (LLE)



From Roweis and Saul (2000).

Example. Faces dataset



- 2000 images, 20×28 pixels.
- Number of features: $p = 560$.
- Applied LLE with 16 nearest neighbors to find a 2D projection.
- Components correlated with nonlinear features of image, such as pose and expression.

Multidimensional scaling

Multidimensional scaling is a technique for projecting data onto a low-dimensional space, while approximately preserving the distance between every pair of samples in the original dataset.

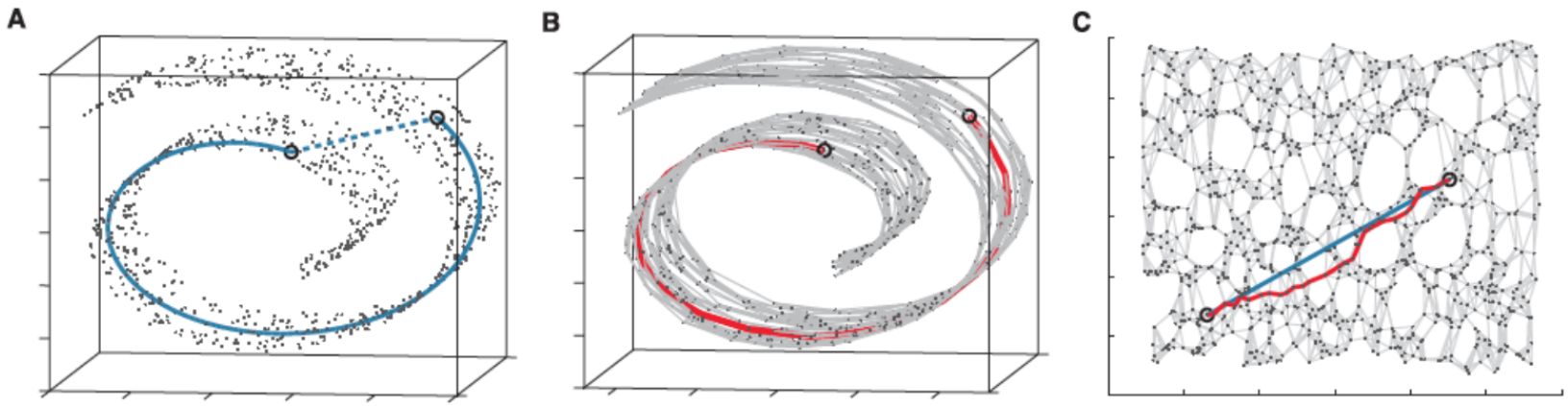
If $d(i, j)$ is the distance between x_i and x_j, we try to find a low-dimensional representation z_i of every sample, which minimizes:

$$\sum_{i,j} (d(i,j) - \parallel z_i - z_j \parallel)^2$$

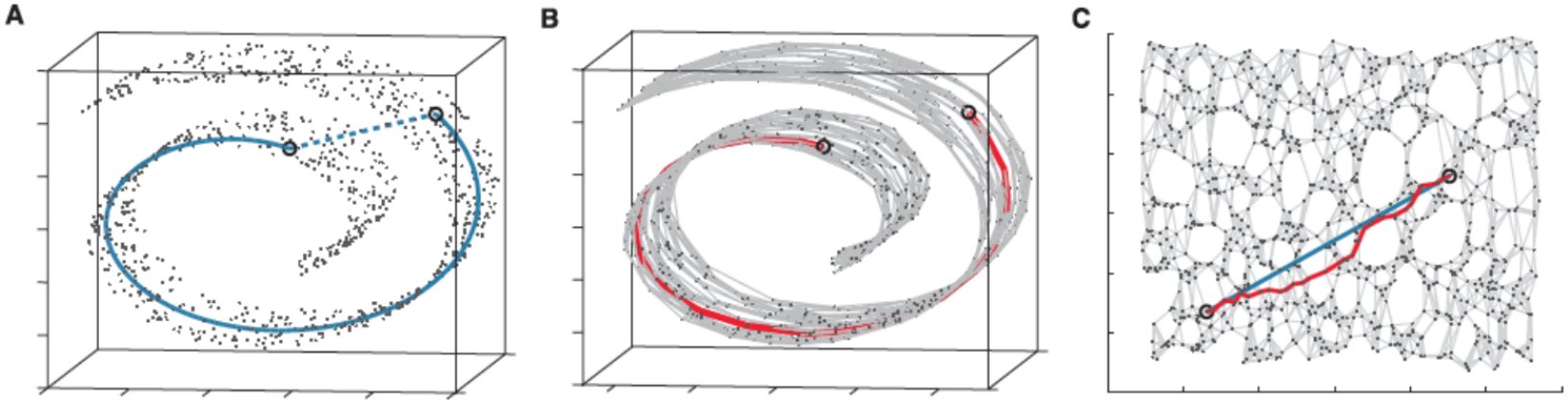
The function d can be any distance, not just the Euclidean distance between two samples.

Isomap

- Suppose that the data are clustered on a low dimensional **manifold** embedded in a high dimensional space.
- The relevant distance between two samples may not be the Euclidean distance on the space of predictors, but the shortest distance on the manifold.
- This distance is called the **geodesic**.

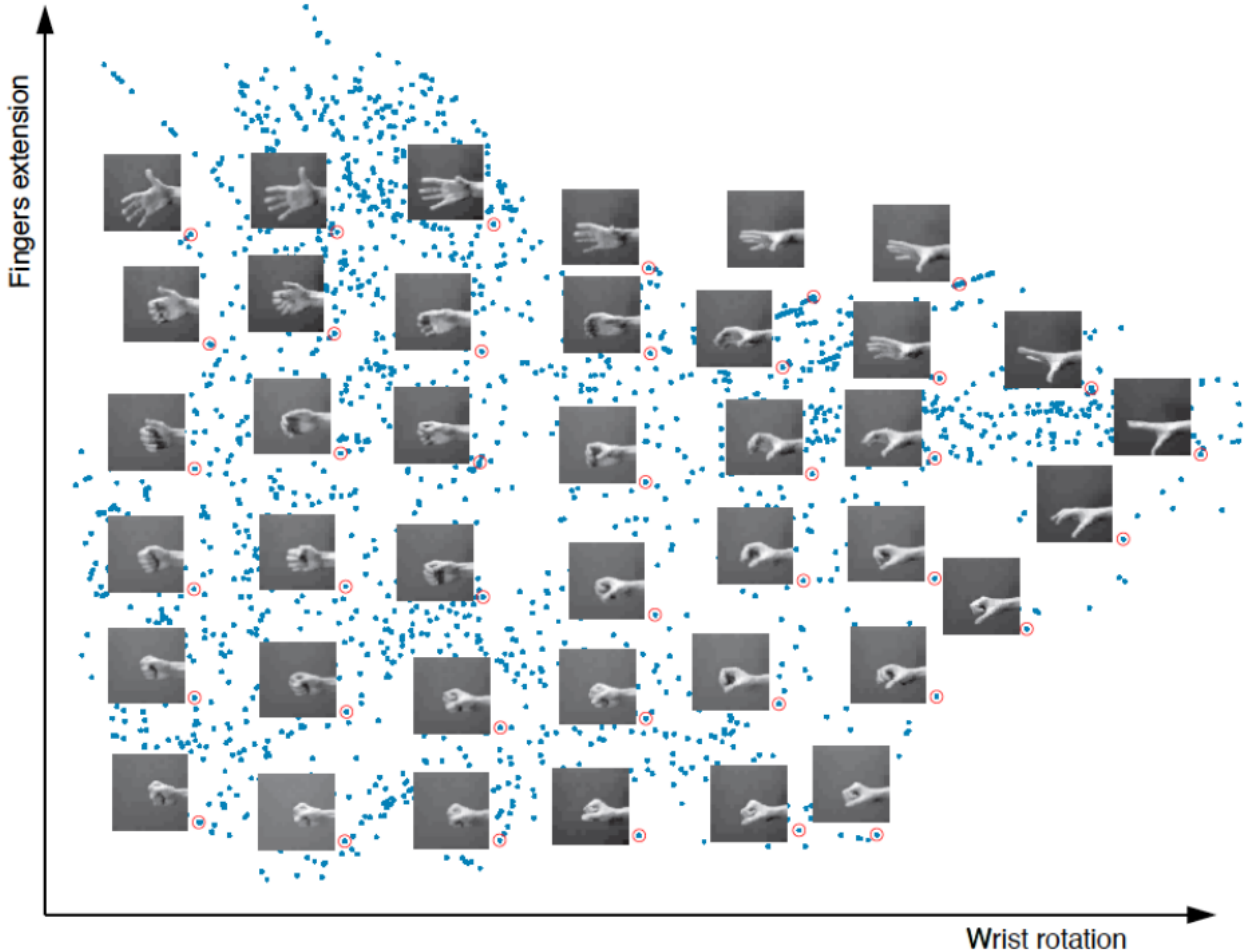


Isomap



- We don't know the manifold a priori.
- However, a nearest neighbor graph gives an approximation.
- **Idea:**
  1. Use the length of the shortest path on the graph as a proxy for the geodesic distance.
  2. Apply multidimensional scaling to visualize the manifold.

Example. Hands dataset



Summary

- Non-linear dimensionality reduction allows us to visualize complex data in low dimensions.
- This is useful when the samples concentrate on a non-linear manifold in high-dimensional space.
- Most methods exploit the nearest neighbor graph in some form or another.
- The data must have a good signal to noise ratio and high density so that the manifold properties are identifiable from local information. This is common in computer vision and artificial intelligence tasks more generally, e.g.:
  1. Digit and letter recognition.
  2. Facial expression analysis.
  3. 3D physical models.