Lecture 5: Dimensionality Reduction and Data Visualization

Bingqing Cheng

Trinity College, the University of Cambridge

bc509@cam.ac.uk

Recap

- The basic concept in machine learning
- Linear regression, nonlinear regression, kernel regression
- Neural networks, deep learning
- Representations of molecules

Outline

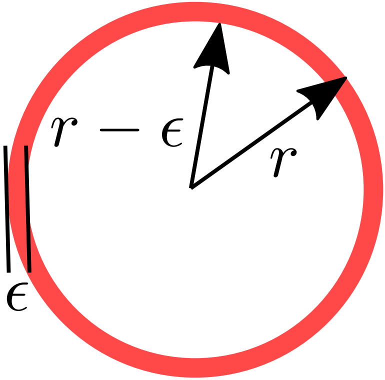
- Curse of the dimensionality
- blessing of non-uniformity
- Prerequisite in Statistics and Mathematics
- Principal component analysis (PCA)
- Non-linear dimensionality reduction methods

Curse of the dimensionality

Volume of a d dimensional sphere of radius r:

$$V(r) = \frac{2\pi^{d/2} r^d}{d\Gamma(d/2)}$$

where $\Gamma()$ is the gamma function.



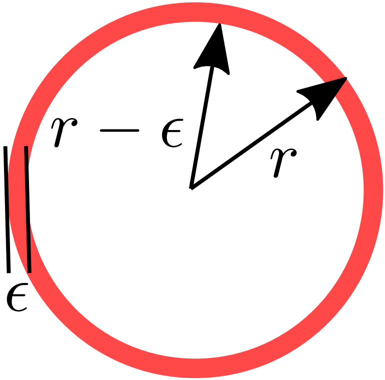
Question:
What is the fraction of volume contained in the outermost shell of thickness ϵ?

Curse of the dimensionality

Volume of a d dimensional sphere of radius r:

$$V(r) = \frac{2\pi^{d/2} r^d}{d\Gamma(d/2)}$$

where $\Gamma()$ is the gamma function.



Answer:

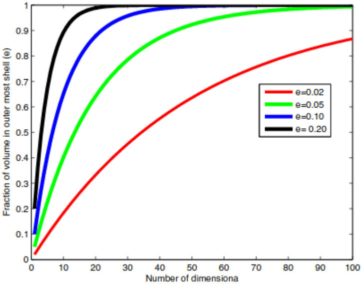
$$f_d = \frac{V_d(r) - V_d(r-\epsilon)}{V_d(r)} = 1 - (1 - \frac{\epsilon}{r})^d$$

Curse of the dimensionality

Volume of a d dimensional sphere of radius r:

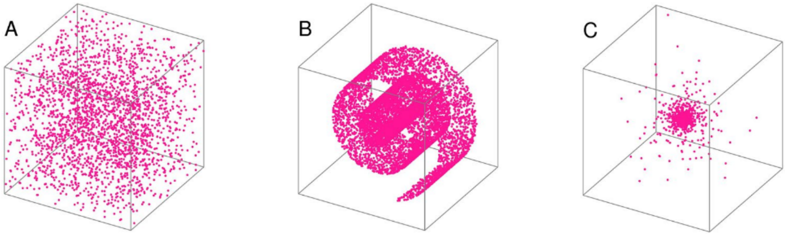
$$V(r) = \frac{2\pi^{d/2} r^d}{d\Gamma(d/2)}$$

where $\Gamma()$ is the gamma function.



This means that in high dimensional spaces most of the data are on the surface.

Blessing of non-uniformity



Question:
How to reduce the dimensionality of the three data sets here?

Motivation

Why dimensionality reduction?

- Visualization, interpretation
- Cheap to train a model
- Missing data, extrapolation
- Remove redundency in the data set

Prerequisite in Statistics and Mathematics

- Variance σ^2 of 1D data:

$$\sigma^2 = \frac{\sum_i (x_i - \bar{x})^2}{N-1}$$

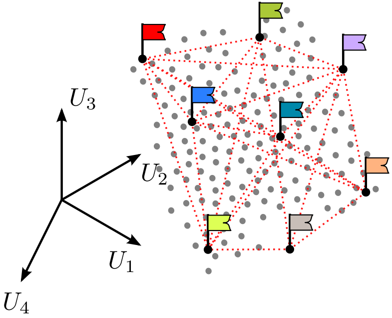
- Covariance

$$C_{jj'} = \frac{\sum_i (X_i^j - \bar{X^j})(X_i^{j'} - \bar{X^{j'}})}{N-1}$$

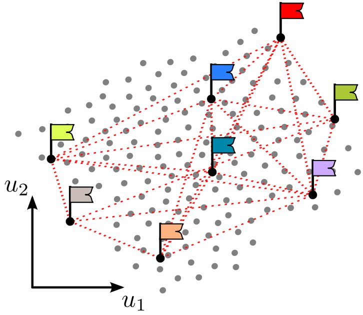
- Eigenvalues and eigenvectors

$$\mathbf{C}\mathbf{v}^j = \lambda^j \mathbf{v}^j$$

Dimension reduction



$\{\mathbf{X}_i\}^D$

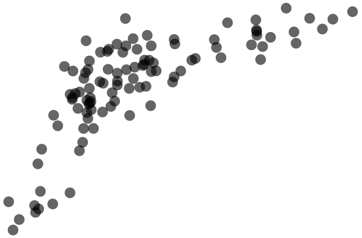


$\{\mathbf{x}_i\}^d$

Question:

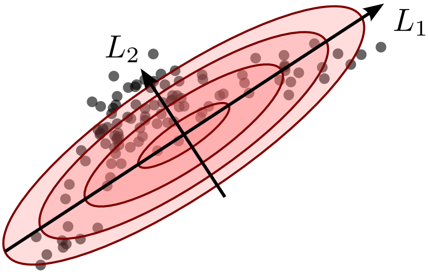
What do you want to preserve?

Principal component analysis



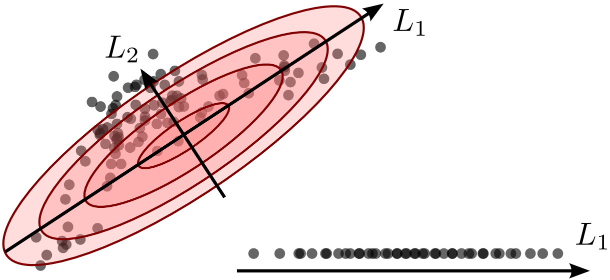
Question:
What is preserved during PCA?

Principal component analysis



Question:
What is preserved during PCA?

Principal component analysis



Question:
What is preserved during PCA?

Principal component analysis

PCA identifies the axis that accounts for the largest amount of variance in the data set.

- $\{\mathbf{X}_i\}^D$: data in the D-dimensional space
- $\{\mathbf{x}_i\}^d$: linear projection in the low d dimensional space
- $\mathbf{c}$: normalized projection matrix

$$\mathbf{x}_i = \mathbf{X}_i\mathbf{c}$$

- Covariance of the data: $\mathbf{C} = \mathbf{X}^T\mathbf{X}$
- Covariance of the projected data: $\mathbf{x}^T\mathbf{x}$

Principal component analysis

PCA identifies the axis that accounts for the largest amount of variance in the data set.

- $\{\mathbf{X}_i\}^D$: data in the D-dimensional space
- $\{\mathbf{x}_i\}^d$: linear projection in the low d dimensional space
- $\mathbf{c}$: normalized projection matrix

$$\mathbf{x}_i = \mathbf{X}_i \mathbf{c}$$

- Covariance of the data: $\mathbf{C} = \mathbf{X}^T \mathbf{X}$
- Covariance of the projected data: $\mathbf{x}^T \mathbf{x}$

Given d , how to reserve the largest amount of variance?

Principal component analysis

PCA identifies the axis that accounts for the largest amount of variance in the data set.

- $\{\mathbf{X}_i\}^D$: data in the D-dimensional space
- $\{\mathbf{x}_i\}^d$: linear projection in the low d dimensional space
- $\mathbf{c}$: normalized projection matrix

$$\mathbf{x}_i = \mathbf{X}_i \mathbf{c}$$

- Covariance of the data: $\mathbf{C} = \mathbf{X}^T\mathbf{X}$
- Covariance of the projected data: $\mathbf{x}^T\mathbf{x}$

Keep the first d eigenvectors of the covariance matrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}$

Principal component analysis

- The covariance matrix $\mathbf{C} = \mathbf{X}^T\mathbf{X}$: $D \times D$ form.
- Eigenvalues $\{\lambda^j\}$
- Corresponding eigenvectors $\{\mathbf{v}^j\}$ of the matrix

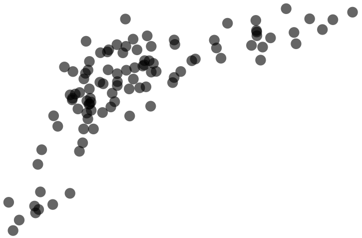
Eigenvalues and eigenvectors fulfills

$$\mathbf{C}\mathbf{v}^j = \lambda^j \mathbf{v}^j$$

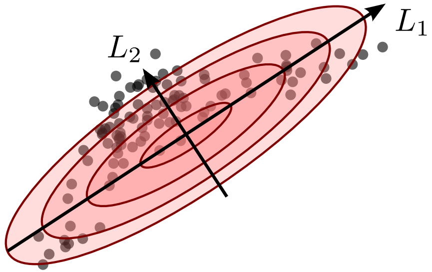
for $j = 1 \dots D$.
One can find the eigenvalues $\{\lambda^j\}$ by solving

$$det(\mathbf{C} - \lambda \mathbf{I}) = 0$$

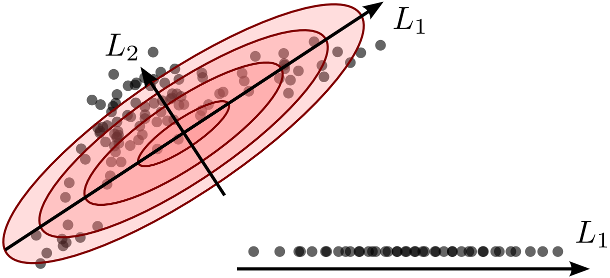
Principal component analysis



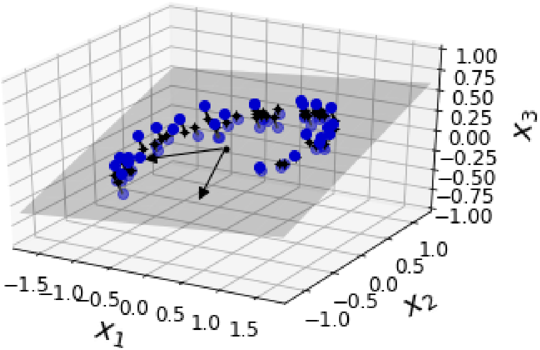
Principal component analysis



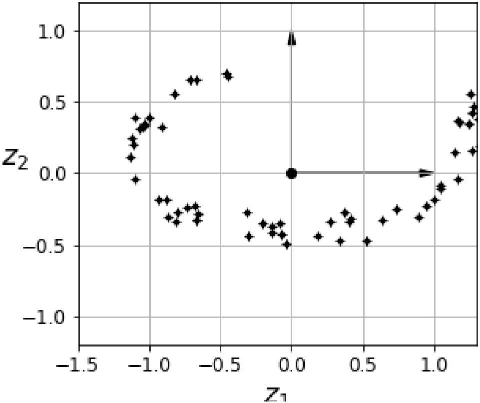
Principal component analysis



Principal component analysis

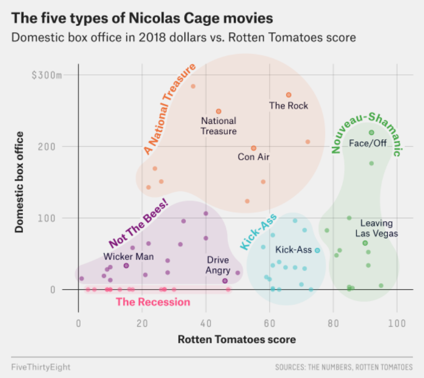


Principal component analysis



PCA, step by step

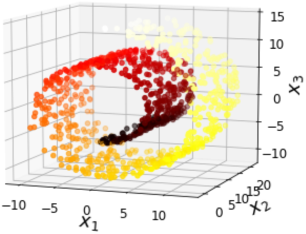
1. Scale and center the data. (How/whether to scale?)
2. Calculate the covariance matrix.
3. Choosing principal components
4. Visualize and validate.



what if we change the units of the axes?

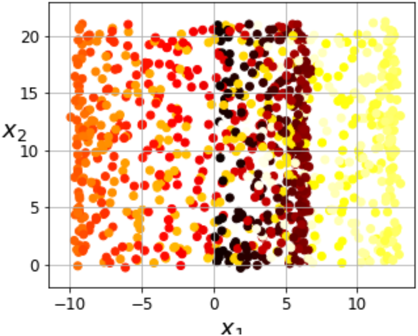
Criticism of PCA

- No universal way of scaling.
- Interesing features of the data may be hiding in "small" dimensions.
- Do not handle complex topology well.

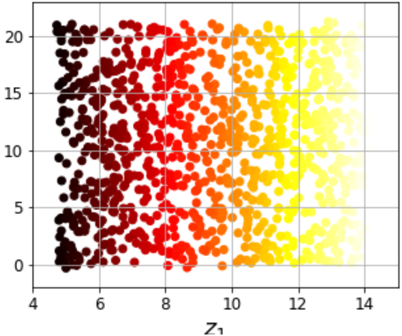


Criticism of PCA

- No universal way of scaling.
- Interesing features of the data may be hiding in “small” dimensions.
- Do not handle complex topology well.



What you get with PCA



What you want

Multidimensional scaling (MDS)

A literal implementation of the general idea of dimensionality reduction

- Define a $N \times N$ matrix $\mathbf{D}$ of pairwise distances between N points
- Find a low d projection $\{\mathbf{x}_i\}$ that maximally preserves $\mathbf{D}$

$$\|\mathbf{x}_i - \mathbf{x}_j\| \approx D_{ij}.$$

which means to minimize the "stress" (distortion)

$$\text{Stress} \sim \left\| \mathbf{D} - \|\mathbf{x}_i - \mathbf{x}_j\| \right\|^2.$$

- Usually the optimization is done iteratively.

Multidimensional scaling (MDS)

A literal implementation of the general idea of dimensionality reduction

- Define a $N \times N$ matrix $\mathbf{D}$ of pairwise distances between N points
- Find a low d projection $\{\mathbf{x}_i\}$ that maximally preserves $\mathbf{D}$

$$\|\mathbf{x}_i - \mathbf{x}_j\| \approx D_{ij}.$$

which means to minimize the "stress" (distortion)

$$\text{Stress} \sim \left\| \mathbf{D} - \|\mathbf{x}_i - \mathbf{x}_j\| \right\|^2.$$

- Usually the optimization is done iteratively.

If $\mathbf{D}$ is the Euclidean norm, classical MDS (use eigenvalues) is the best linear projection preserving the squared distances. It corresponds to PCA, but it is more easily generalized to different dissimilarities.

Multidimensional scaling (MDS)

A literal implementation of the general idea of dimensionality reduction

- Define a $N \times N$ matrix $\mathbf{D}$ of pairwise distances between N points
- Find a low d projection $\{\mathbf{x}_i\}$ that maximally preserves $\mathbf{D}$

$$\|\mathbf{x}_i - \mathbf{x}_j\| \approx D_{ij}.$$

which means to minimize the “stress” (distortion)

$$\text{Stress} \sim \left\| \mathbf{D} - \|\mathbf{x}_i - \mathbf{x}_j\| \right\|^2.$$

- Usually the optimization is done iteratively.

What are the other ways of defining $\mathbf{D}$?

Transforming the feature Space

- Design matrix $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(d)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(d)} \\ \vdots & \vdots & \vdots & \vdots \\ x_N^{(1)} & x_N^{(2)} & \dots & x_N^{(d)} \end{bmatrix}$$

- Take a feature vector and apply non-linear transformation ϕ:

$$\phi : \mathcal{R}^d \rightarrow \mathcal{R}^k$$

- Polynomial basis:

$$\phi(x) = \begin{bmatrix} 1 & x^1 & x^2 & x^3 & x^4 \end{bmatrix}$$

- Others: splines, radial basis functions, ...
- Take the distance $\mathbf{D}$ in the feature space.

Kernel to distance

- Derivation of the kernel trick in the lecture notes.
- $\mathbf{K}$ is the kernel matrix. k_{ij} is the similarities between each pair of data i and j.
- Obtain the distance matrix $\mathbf{D}$ from the kernel matrix $\mathbf{K}$, e.g.

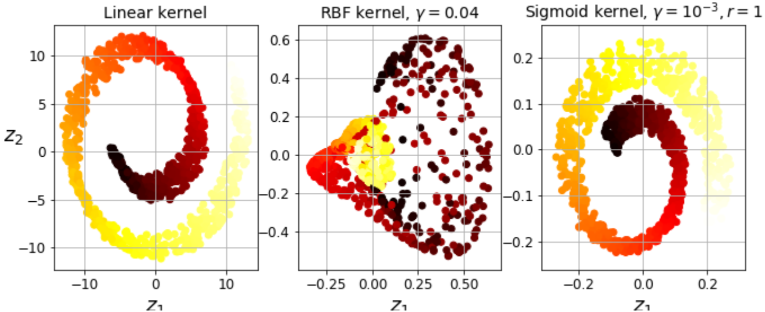
$$d_{ij} = 1 - \frac{k_{ij}}{\sqrt{k_{ii}k_{jj}}}$$

- Cauchy–Schwarz inequality needs to hold for $\mathbf{D}$:

$$d_{ij} + d_{jk} \geq d_{ik}$$

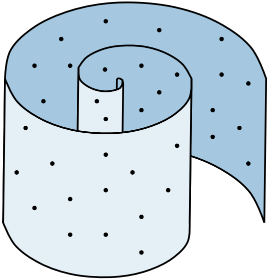
Kernel PCA

- PCA on the kernel matrix



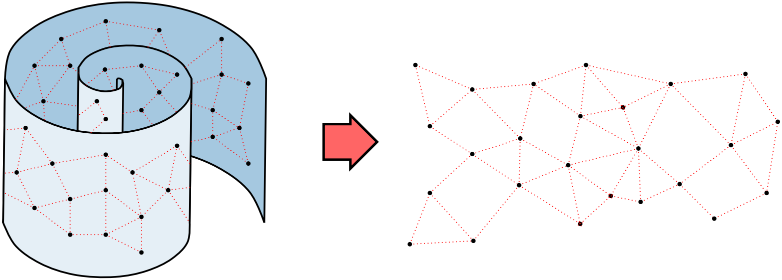
Isometric Mapping (ISOMAP)

- Approximate pairwise geodesic distances.
- A geodesic is the shortest path in manifold between two points x and y. (Approximate geodesics by hopping between neighbours, sensitive to noise).
- Given these pairwise geodesic distances, use MDS to find a d-dimensional embedding that preserves geodesic distances.



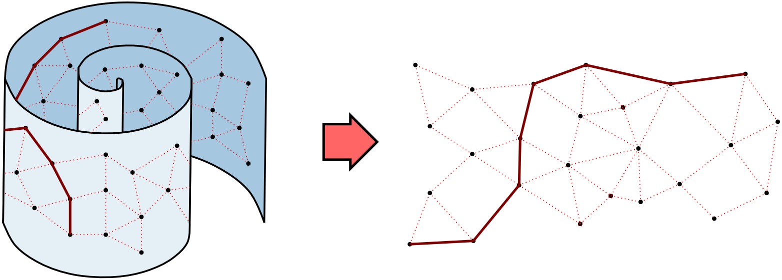
Isometric Mapping (ISOMAP)

- Approximate pairwise geodesic distances.
- A geodesic is the shortest path in manifold between two points x and y. (Approximate geodesics by hopping between neighbours, sensitive to noise).
- Given these pairwise geodesic distances, use MDS to find a d-dimensional embedding that preserves geodesic distances.



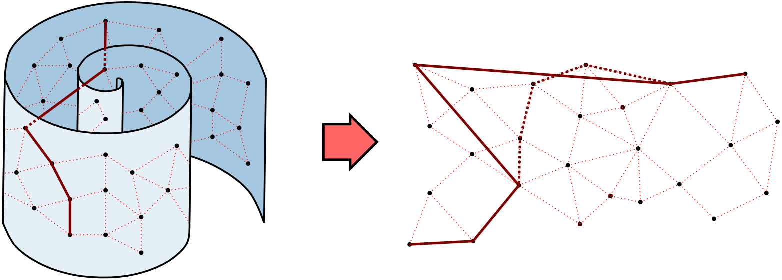
Isometric Mapping (ISOMAP)

- Approximate pairwise geodesic distances.
- A geodesic is the shortest path in manifold between two points x and y. (Approximate geodesics by hopping between neighbours, sensitive to noise).
- Given these pairwise geodesic distances, use MDS to find a d-dimensional embedding that preserves geodesic distances.



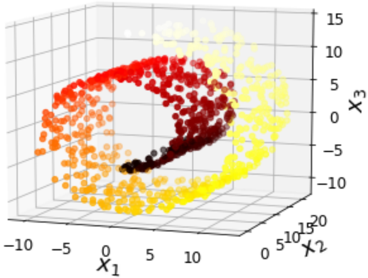
Isometric Mapping (ISOMAP)

- Approximate pairwise geodesic distances.
- A geodesic is the shortest path in manifold between two points x and y. (Approximate geodesics by hopping between neighbours, sensitive to noise).
- Given these pairwise geodesic distances, use MDS to find a d-dimensional embedding that preserves geodesic distances.



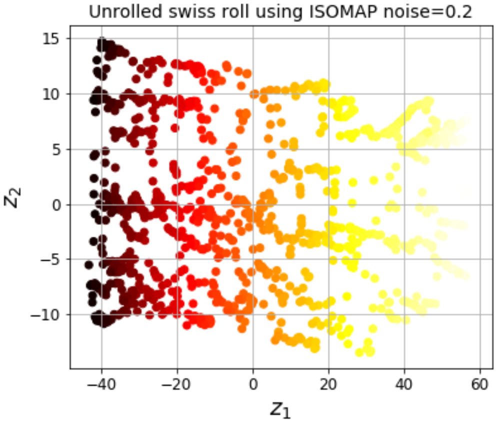
Isometric Mapping (ISOMAP)

Unroll the Swiss roll, with different noise level.



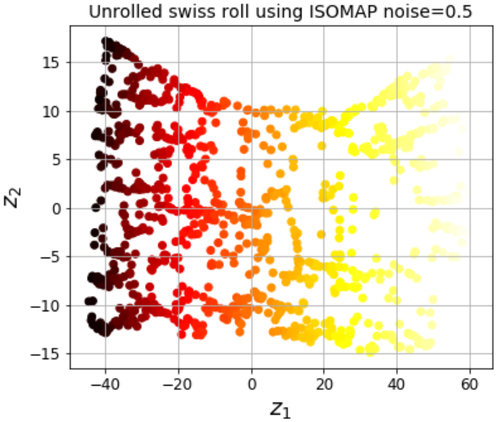
Isometric Mapping (ISOMAP)

Unroll the Swiss roll, with different noise level.



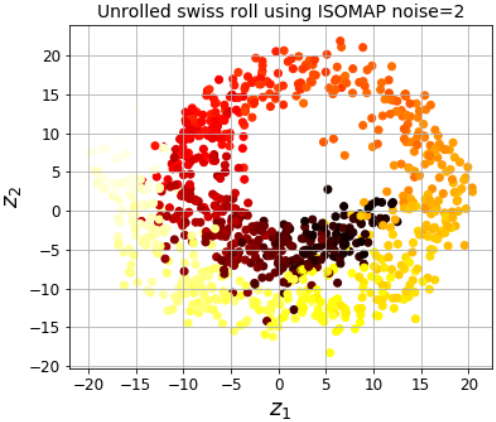
Isometric Mapping (ISOMAP)

Unroll the Swiss roll, with different noise level.



Isometric Mapping (ISOMAP)

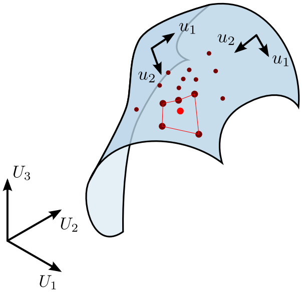
Unroll the Swiss roll, with different noise level.



Locally Linear Embedding (LLE)

If the manifold is locally flat each point can be expressed as a combination of its neighbors.

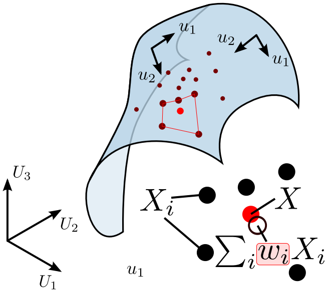
- Find (k-nearest) neighbors for each data point.
- Compute a weight vector $\mathbf{w}_i$ that best reconstructs each $\mathbf{x}_i$ by a linear combination of its k-NN.
- Compute the embedding in $\mathcal{R}^d$ that minimizes reconstruction error using $\mathbf{w}_i$ and its corresponding k-NN.



Locally Linear Embedding (LLE)

If the manifold is locally flat each point can be expressed as a combination of its neighbors.

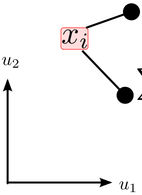
- Find (k-nearest) neighbors for each data point.
- Compute a weight vector $\mathbf{w}_i$ that best reconstructs each $\mathbf{x}_i$ by a linear combination of its k-NN.
- Compute the embedding in $\mathcal{R}^d$ that minimizes reconstruction error using $\mathbf{w}_i$ and its corresponding k-NN.



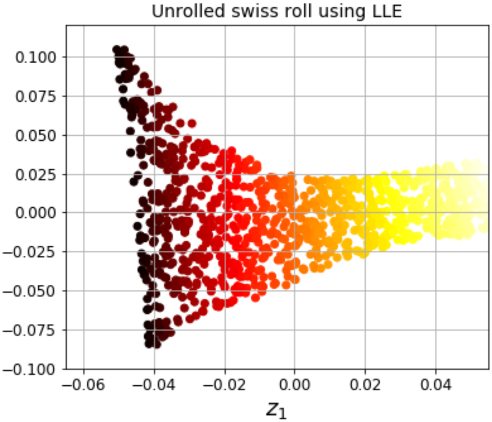
Locally Linear Embedding (LLE)

If the manifold is locally flat each point can be expressed as a combination of its neighbors.

- Find (k-nearest) neighbors for each data point.
- Compute a weight vector $\mathbf{w}_i$ that best reconstructs each $\mathbf{x}_i$ by a linear combination of its k-NN.
- Compute the embedding in $\mathcal{R}^d$ that minimizes reconstruction error using $\mathbf{w}_i$ and its corresponding k-NN.

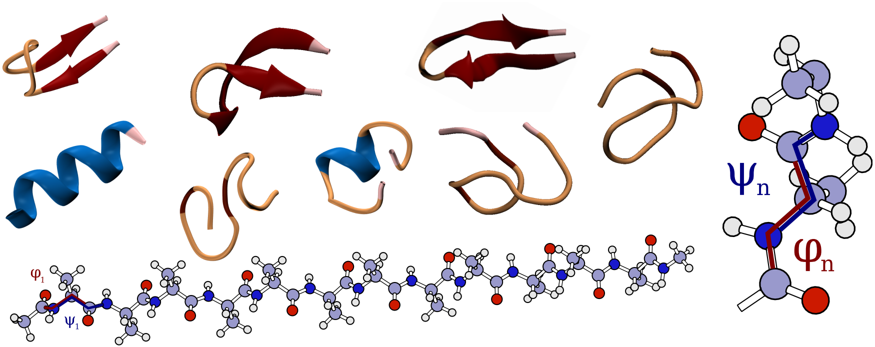


Unroll the Swiss roll, with noise level = 0.2.

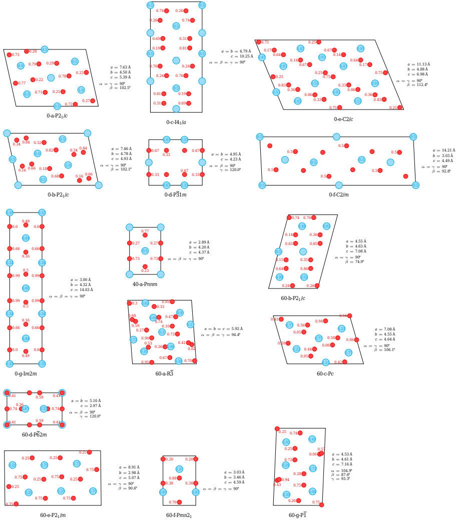


Other dimensionality reduction schemes

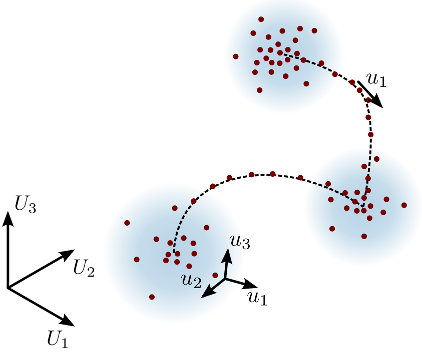
Dimensionality reduction for chemical data

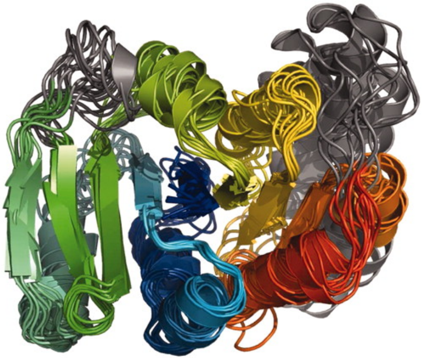


Dimensionality reduction for chemical data



Dimensionality reduction for chemical data





Outline of next lecture

- Representing structual data
- More tricks of the trade
    - Sparsification
    - Clustering

Questions?

Thank you for your attention!