Multidimensional Data Visualization

Nonlinear Projection Methods

Dimensionality Reduction

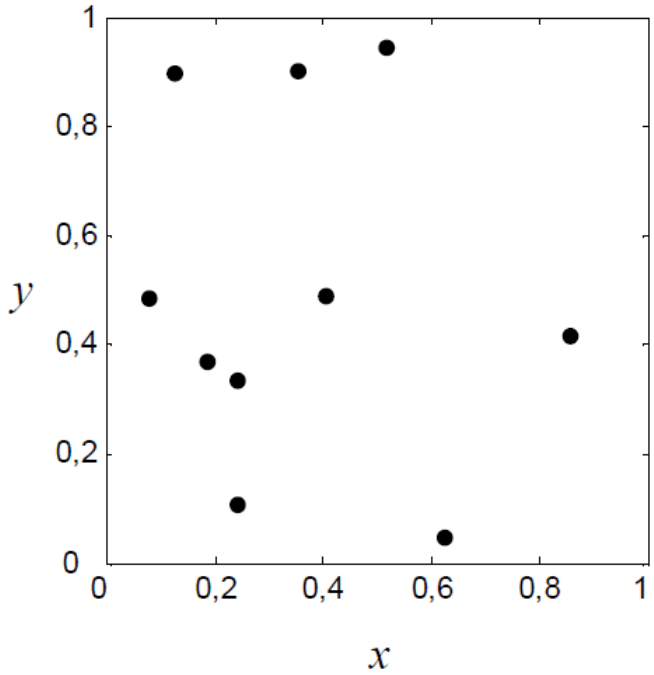
- It is difficult to perceive the data structure using the direct visualization methods, particularly when we deal with large data sets or data of high dimensionality.
- Projection methods are based on reduction of the dimensionality of data.
- Their advantage is that each n-dimensional object is represented as a point in the space of low-dimensionality d, $d < n$, usually $d = 2$.
- There exists a lot of methods that can be used for reducing the dimensionality.
- The aim of these methods is to represent the multidimensional data in a low-dimensional space so that certain properties (such as distances, topology or other proximities) of the data set were preserved as faithfully as possible.
- These methods can be used to visualize the multidimensional data, if a small enough resulting dimensionality is chosen.

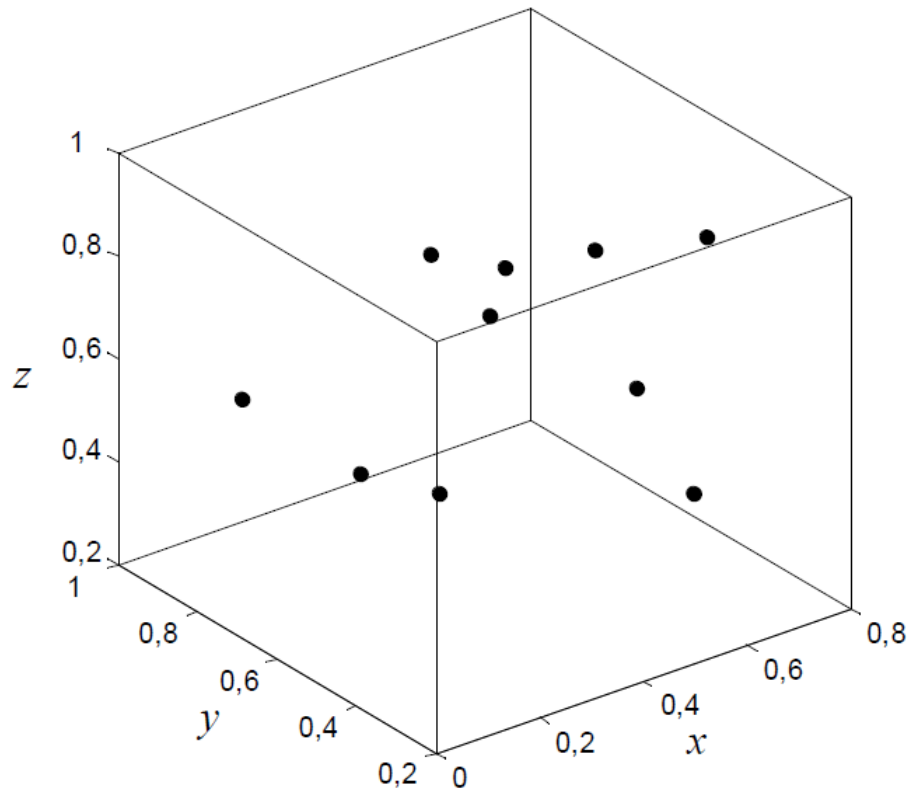
Projection Methods

- The projection methods are used for *transformation* of multidimensional data to a low-dimensional space.
- The aim of these methods is to represent the multidimensional data in a low-dimensional space so that certain properties of the data set were preserved as faithfully as possible.
- These methods can be used to visualize the multidimensional data, if a sufficiently small dimensionality of the projection space $\mathbb{R}^d$ is chosen ($d = 2$ or $d = 3$).
- We call the space $\mathbb{R}^d$ as a display or image space, since its points can be observed visually.
- These methods usually invoke formal mathematical criteria by which the projection distortion is minimized.

Scatter Plots

- *Scatter plots* are one of the most commonly used techniques for data representation on a plane $\mathbb{R}^2$ or space $\mathbb{R}^3$. Points are displayed in the classic (x, y) or (x, y, z) format.





Projection Methods

- Suppose that the multidimensional data set is defined by a matrix

$$X = \{X_1, X_2, \ldots, X_m\} = \left\{x_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, n\right\}.$$

  Here m is the number of objects (n-dimensional points $X_i \in \mathbb{R}^n$, where $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $i \in \{1, \ldots, m\}$). x_{ij} is the jth coordinate, corresponding to the jth feature.
- One needs to find a transformation of the points $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1, \ldots, m$, into points $Y_i = (y_{i1}, y_{i2}, \ldots, y_{id})$, $i = 1, \ldots, m$, that are on a low-dimensional space $\mathbb{R}^d$, $d < n$. One-dimensional space ($d = 1$) can also be used, however more information can be preserved when observing points on a plane ($d = 2$) or a 3D space ($d = 3$).

Criteria of the Projection Quality

- There are some formal mathematical criteria of the projection quality. These criteria are optimized in order to get the optimal projection of multidimensional data onto a low-dimensional space.
- The main goal is to preserve the proportions of distances or estimations of other proximities between the multidimensional points in the image space as well as to preserve, or even to highlight other characteristics of the multidimensional data (for example, clusters).

Nonlinear Transformation

- There are linear and nonlinear projection methods.
- Linear projection methods pursue a linear transformation of data. A *linear transformation* may be described by linear equations

$$Y_i = X_i A.$$

- A nonlinear transformation may be described as follows:

$$Y = f(X),$$

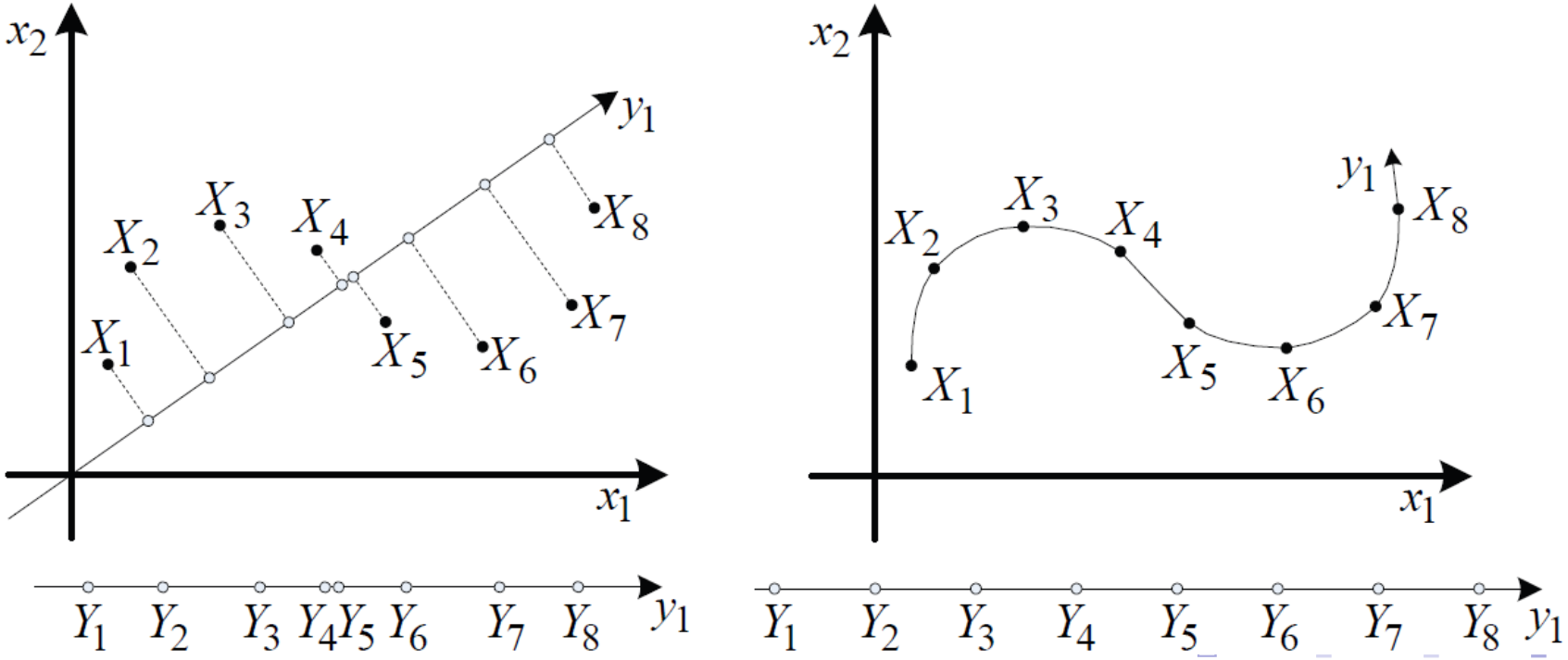
where f is a nonlinear function and

$$Y = \{Y_1, Y_2, \ldots, Y_m\} = \{y_{ij},\ i = 1, \ldots, m,\ j = 1, \ldots, n\}.$$

- The nonlinear transformation is more complicated than the linear one and requires more time-consuming computations. However, such a transformation allows us to preserve the characteristics of multidimensional data better as compared with the linear transformation if $d < n$, i.e. the data are projected to a lower-dimensional space.

Linear and Nonlinear Projection

- Let the two-dimensional points $X_1, X_2, \ldots, X_8$ be spread so that the distances between the nearest points are equal.
- If we project to the one-dimensional space using the linear projection (to the line y_1), equal distances between the nearest points are not preserved. However, in the case of the nonlinear projection, when the proper transformation is found, the distances between the nearest points remain equal.



Nonlinear Projection Methods

- multidimensional scaling,
- locally linear embedding,
- isometric feature mapping,
- principal curves.

Proximity Measures

- The aim of projection methods is to transform multidimensional data to a low-dimensional space so that the proximity of the data was possibly preserved. Therefore, *proximity measures* should be defined.
- Often the proximity is measured using the Euclidean distance, which belongs to the group of Minkowski distances. The Minkowski distance between two objects $X_k = (x_{k1}, x_{k2}, \ldots, x_{kn})$ and $X_l = (x_{l1}, x_{l2}, \ldots, x_{ln})$ is defined by the formula:

$$d_q(X_k, X_l) = \left\{ \sum_{j=1}^{n} |x_{kj} - x_{lj}|^q \right\}^{\frac{1}{q}}.$$

- Some other proximity measures are also possible: Canberra distance, Bray-Curtis dissimilarity, correlation, etc.

Minkowski Distances

- City-block or Manhattan distance, $q = 1$:

$$d_1(X_k, X_l) = \sum_{j=1}^{n} |x_{kj} - x_{lj}|.$$

- Euclidean distance, $q = 2$:

$$d_2(X_k, X_l) = \sqrt{\sum_{j=1}^{n} |x_{kj} - x_{lj}|^2}.$$

- Chebyshev distance, $q = \infty$:

$$d_\infty(X_k, X_l) = \max_j \left| x_{kl} - x_{lj} \right|.$$

Distances between two objects X_k and X_l satisfy

- $d(X_k, X_l)$ is a nonnegative real number;
- $d(X_k, X_k) = 0$;
- $d(X_k, X_l) = d(X_l, X_k)$, i.e. the distance from object X_k to object X_l is equal to the distance from object X_l to object X_k;
- $d(X_k, X_l) \leq d(X_k, X_j) + d(X_j, X_l)$, i.e. the distance between any two objects X_k and X_l cannot be larger than a sum of distances between objects X_k, X_j and X_l, X_j (triangle inequality).

Multidimensional scaling

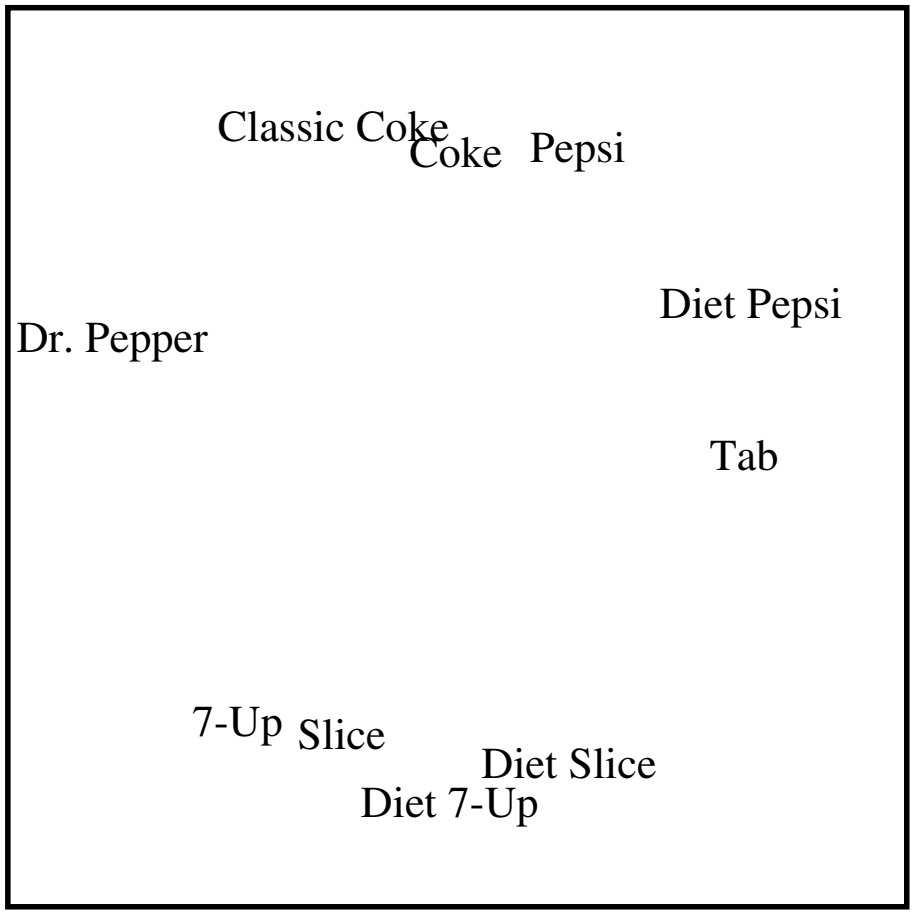
- *Multidimensional scaling* (MDS) refers to a group of methods that are widely used for dimensionality reduction and visualization of multidimensional data.
- The data for MDS is a matrix consisting of pairwise proximities of the objects.
- Let us denote the pairwise proximity of the ith and jth objects by δ_{ij}. If the objects are defined by the multidimensional points $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1, \ldots, m$, the proximity can be measured by the distance between points: $\delta_{ij} = d(X_i, X_j)$, where various distances can be used. The distance between points in a low-dimensional space Y_i and Y_j corresponding to the ith and jth objects is denoted by $d(Y_i, Y_j)$.

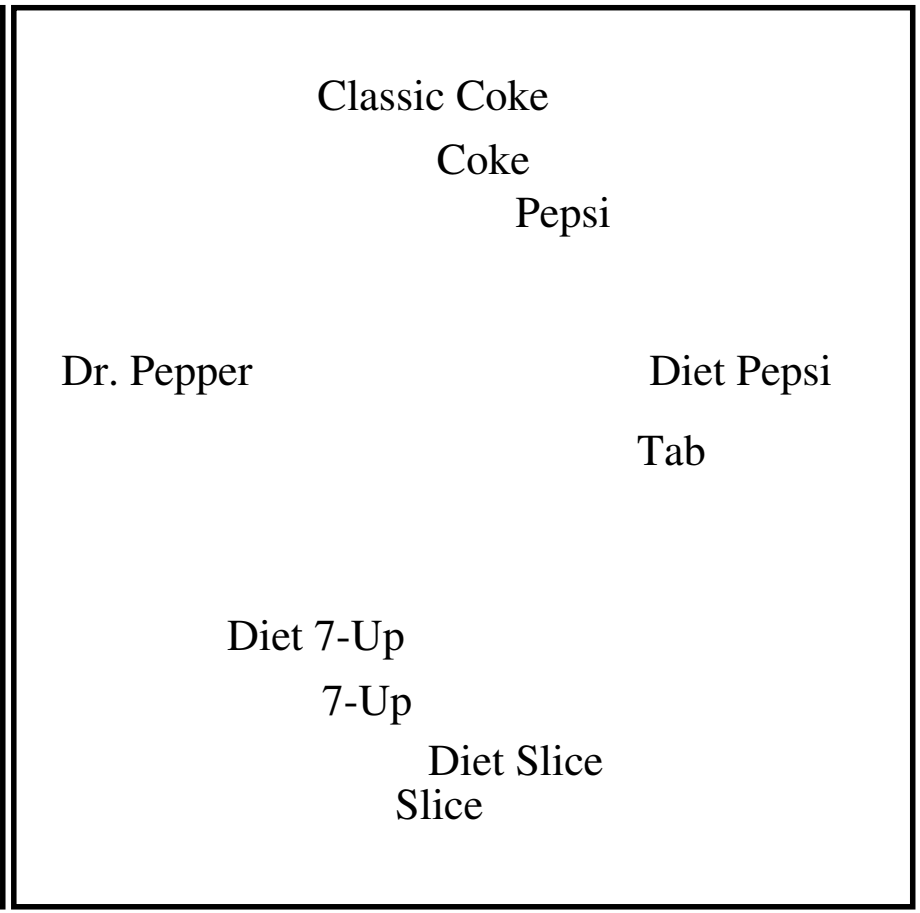
Cola data set

- Cola data set is based on experimental testing of several soft drinks.
- 38 students have tested ten ($m = 10$) different brands of soft drinks.
- Each pair was judged on its dissimilarity in a nine-point scale (1 – very similar, 9 – completely different).
- Accumulated dissimilarities form the data set.

	1	2	3	4	5	6	7	8	9	10
1. Pepsi	0	127	169	204	309	320	286	317	321	238
2. Coke	127	0	143	235	318	322	256	318	318	231
3. Classic Coke	169	143	0	243	326	327	258	318	318	242
4. Diet Pepsi	204	235	243	0	285	288	259	312	317	194
5. Diet Slice	309	318	326	285	0	155	312	131	170	285
6. Diet 7-Up	320	322	327	288	155	0	306	164	136	281
7. Dr Pepper	286	256	258	259	312	306	0	300	295	256
8. Slice	317	318	318	312	131	164	300	0	132	291
9. 7-Up	321	318	318	317	170	136	295	132	0	297
10. Tab	238	231	242	194	285	281	256	291	297	0

Cola data set visualized using multidimensional scaling





Stress Function

- The goal of multidimensional scaling is to find low-dimensional points $Y_i = (y_{i1}, y_{i2}, \ldots, y_{id})$, such that the distances between the points in the low-dimensional space were as close to the proximities as possible. The least-squares objective function (*raw Stress*) to be minimized can be written as

$$\sigma_r(Y) = \sum_{i<j} w_{ij}(d(Y_i, Y_j) - \delta_{ij})^2,$$

  where $Y = \{Y_1, Y_2, \ldots Y_m\}$, w_{ij} are non-negative weights.
- The *normalized Stress* is defined as follows:

$$\sigma_n(Y) = \frac{\sum_{i<j} w_{ij}(d(Y_i, Y_j) - \delta_{ij})^2}{\sum_{i<j} w_{ij}\delta_{ij}^2}.$$

  The normalization using the parameter $\sum_{i<j} w_{ij}\delta_{ij}^2$ gives a clear interpretation of the visualization quality that depends less on the number of objects m and the scale of proximities.

Relative Error

- The *relative error* is defined as follows:

$$E(Y) = \sqrt{\sigma_n(Y)} = \sqrt{\frac{\sum_{i<j} w_{ij}(d(Y_i, Y_j) - \delta_{ij})^2}{\sum_{i<j} w_{ij}\delta_{ij}^2}}.$$

  The reason for using $E(Y)$ rather than the normalized error $\sigma_n(Y)$ is that $\sigma_n(Y)$ is almost always very small in practice, so $E(Y)$ values are easier to discriminate.

- Often $w_{ij} = 1$, then previous formula becomes as follows:

$$E(Y) = \sqrt{\frac{\sum_{i<j} (d(Y_i, Y_j) - \delta_{ij})^2}{\sum_{i<j} \delta_{ij}^2}}.$$

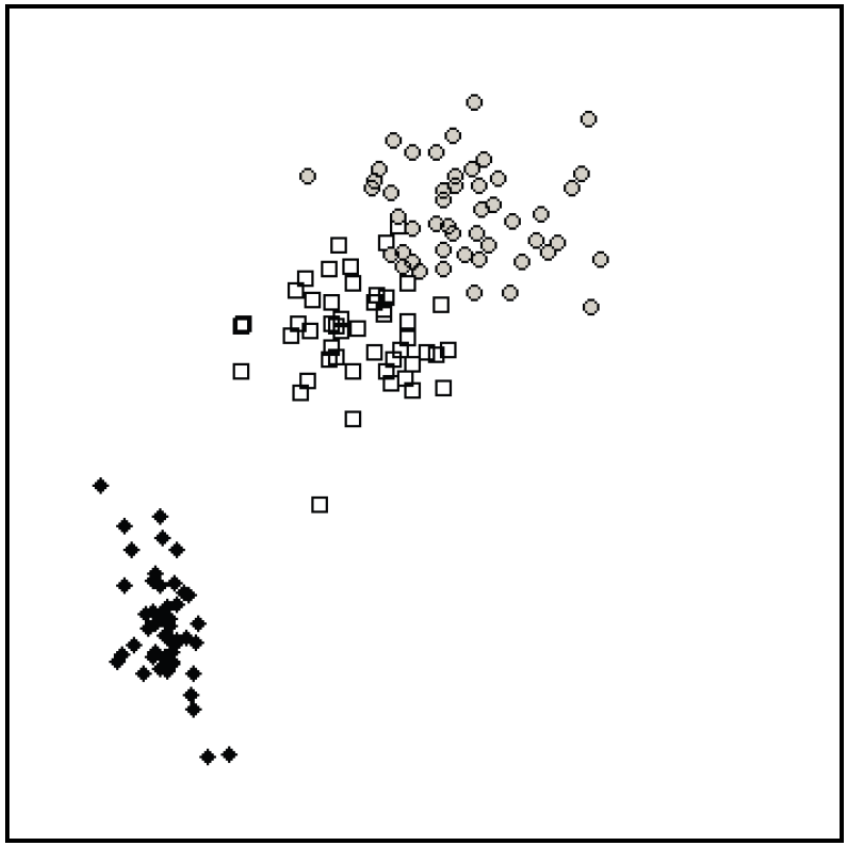
Stress-1 and Other Variants

- *Stress-1* is defined by the formula:

$$\sigma_1(Y) = \sqrt{\frac{\sum_{i<j} (d(Y_i, Y_j) - \delta_{ij})^2}{\sum_{i<j} (d(Y_i, Y_j))^2}}.$$

- There exists a multitude of variants of MDS with different weights and optimization algorithms. Various local optimization strategies will be discussed in this lecture.

Iris Data Set Visualized Using Multidimensional Scaling



SMACOF Algorithm

- The multidimensional scaling Stress function can be minimized in the majorization way. The idea of majorization is to replace iteratively the original complicated function $f(x)$ by an auxiliary function $g(x, z)$, where z is some fixed value. The function $g(x, z)$ has to meet some requirements to be called a majorizing function of $f(x)$. The auxiliary function $g(x, z)$ should be
  - simpler to minimize than $f(x)$,
  - not smaller than the original function, i.e. $f(x) \leq g(x, z)$,
  - touch $f(x)$ at the so-called supporting point z, i.e. $f(z) = g(z, z)$.

Relative Mapping

- *Relative mapping* can be used for mapping new objects, when some objects had been mapped before. It may be of interest to see where new objects are visualized among the already mapped objects.
- Such an optimization problem has a smaller number of variables and takes much shorter computing time. This is achieved by modifying the Stress function:

$$E_R(Y_{\hat{m}+1}, \ldots, Y_m) = \sum_{i,j=\hat{m}+1, i<j}^{m} w_{ij}(\delta_{ij} - d(Y_i, Y_j))^2 + \sum_{i=\hat{m}+1}^{m} \sum_{j=1}^{\hat{m}} w_{ij}(\delta_{ij} - d(Y_i, \hat{Y}_j))^2,$$

where $\hat{m}$ is the number of previously mapped points, $\hat{Y}_j$ are previously mapped points. The number of variables is $(m - \hat{m})d$ instead of md.

Relative Multidimensional Scaling

- The relative mapping may be used in the *relative multidimensional scaling method* to visualize large data sets. The visualization process is divided into three steps:
  1. The basic objects are chosen.
  2. The basic objects are visualized by the MDS algorithm.
  3. The remaining objects are visualized using the relative mapping.
- If objects are defined by multidimensional points, then the basic objects may be chosen according to k-means clustering.
- The visualization results are very dependent on the selected set of the basic objects. The basic objects should be selected so that they were distributed as uniformly as possible all over the data set, which yields better results of the obtained visualization.

Sammon's Mapping

- *Sammon's mapping* is one of the MDS methods.
- The Stress function of Sammon's mapping is as follows:

$$E_S(Y) = \frac{1}{\sum_{k<l} \delta_{kl}} \sum_{i<j} \frac{(\delta_{ij} - d(Y_i, Y_j))^2}{\delta_{ij}}.$$

- Sammon's Stress $E_S(Y)$ is coincident with the function $\sigma_r(Y)$, if

$$w_{ij} = \frac{1}{\sum_{k<l} \delta_{kl}\delta_{ij}}.$$

- Due to the normalization (division by δ_{ij}), the preservation of small values of proximities is emphasized.

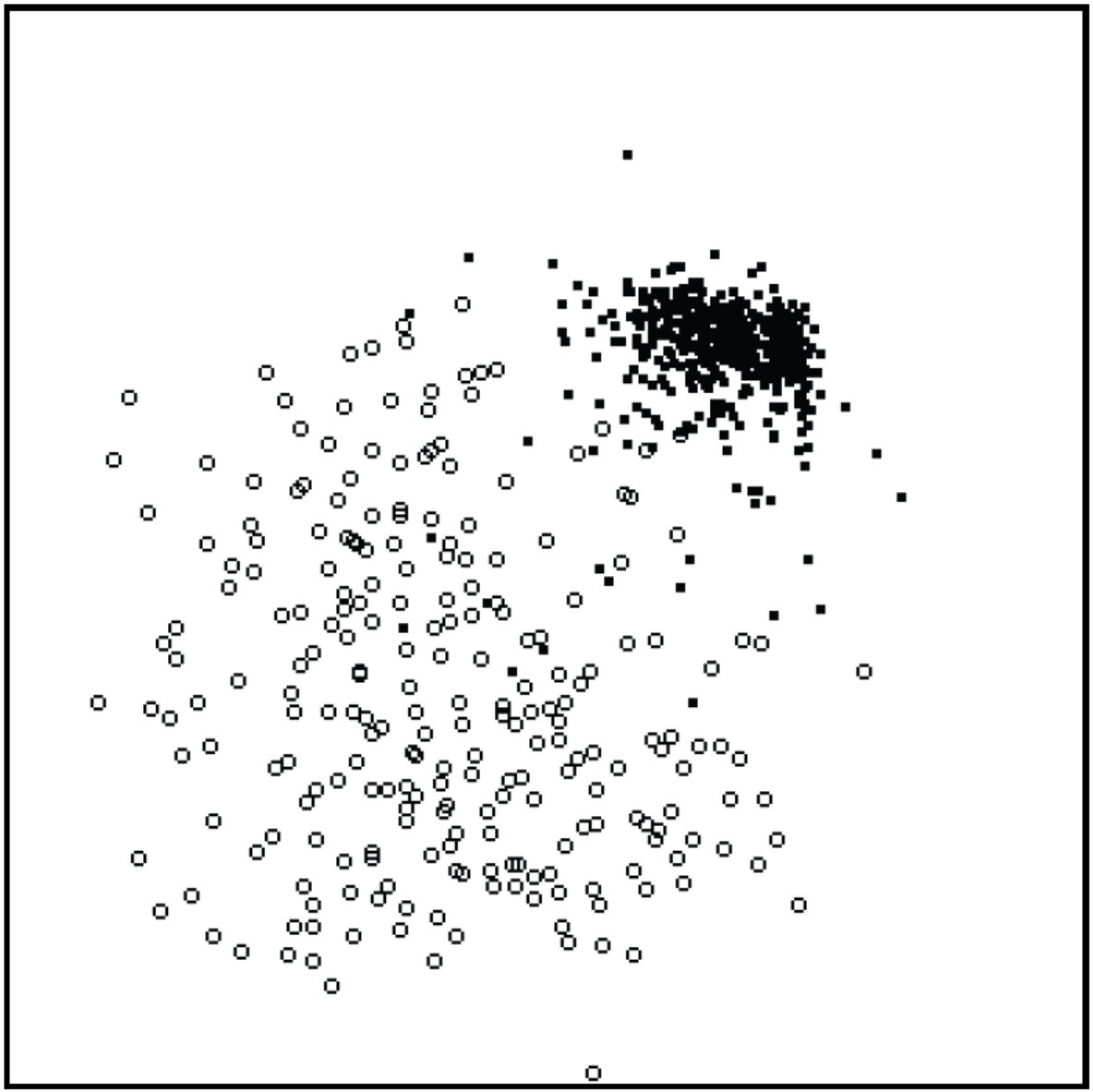
Sammon's Method

- Various optimization methods could be used to minimize the function $E_S(Y)$ when projecting multidimensional objects on the plane ($d = 2$).
- The coordinates y_{ik}, $i = 1, \ldots, m$, $k = 1, 2$, of the two-dimensional points $Y_i \in \mathbb{R}^2$ are computed by the iteration formula:

$$\begin{aligned} y_{ik}(t+1) &= y_{ik}(t) - \eta \triangle(t), \\ \triangle &= \frac{\frac{\partial E_S}{\partial y_{ik}}}{\left| \frac{\partial^2 E_S}{\partial y_{ik}^2} \right|}, \end{aligned}$$

  where t denotes the order number of iteration, η is an optimization step parameter. The coordinates of all m points $Y_i \in \mathbb{R}^2$, $i = 1, \ldots, m$, are recomputed during every iteration.
- The results of $E_S(Y)$ minimization depend on η and on the initial coordinates of points $Y_1, Y_2, \ldots, Y_m$. $\eta \in [0.3, 0.4]$ is recommended.

Breast Cancer Data Set Visualized Using Sammon's Mapping



Manifold-Based Visualization

- Most of real-life data are multidimensional, but they are not truly high-dimensional.
- Multidimensional points just lie on a low-dimensional manifold embedded into a high-dimensional space.
- A *manifold* is an abstract topological space, in which the neighborhood of each point is a subset of the Euclidean space, however the global structure of a manifold may be more complicated.
- A line and a curve are one-dimensional manifolds. The neighborhood of each point on the one-dimensional manifold is a line segment.
- A plane, the surface of a ball, and a toroid are two-dimensional manifolds, etc. The neighborhood of each point on the two-dimensional manifold is a flat region.
- The surface of the Earth is also a two-dimensional manifold.
- A manifold is a smooth low-dimensional surface embedded in a higher dimensional space.

Manifold-Based Visualization

- Multidimensional data can have meaningful hidden low-dimensional structures in the sense of lying on or near to a smooth low-dimensional manifold.
- The intrinsic dimensionality $d \ll n$ of multidimensional data is defined as the minimal number of parameters or latent variables necessary to describe the data.
- An important property of a manifold is its topology, i.e. neighborhood relationships between the subregions of the manifold.
- Nonlinear manifold learning methods are topology preserving methods. The key purpose of such methods is to preserve neighborhood relationships between points. If multidimensional points are close to each other, the points representing them in the low-dimensional space should also be close. In some cases, it is like unfolding a nonlinear manifold.

Manifold-Based Visualization Methods

- A large number of nonlinear manifold learning methods have been proposed over the last decade: locally linear embedding (LLE), isometric feature mapping (ISOMAP), Laplacian eigenmaps (LE), Hessian LLE (HLLE), etc.
- These methods are supposed to overcome the difficulties experienced with other classical nonlinear approaches. They are able to recover the intrinsic geometric structure of nonlinear multidimensional data.
- Local approaches (e.g., LLE, Laplacian eigenmaps) attempt to preserve the topology (the local geometry) of the data.
- In addition, global approaches (e.g. ISOMAP) attempt to preserve geometry at all scales: nearby multidimensional points are projected to nearby points in a low-dimensional space, and faraway multidimensional points to faraway low-dimensional points.

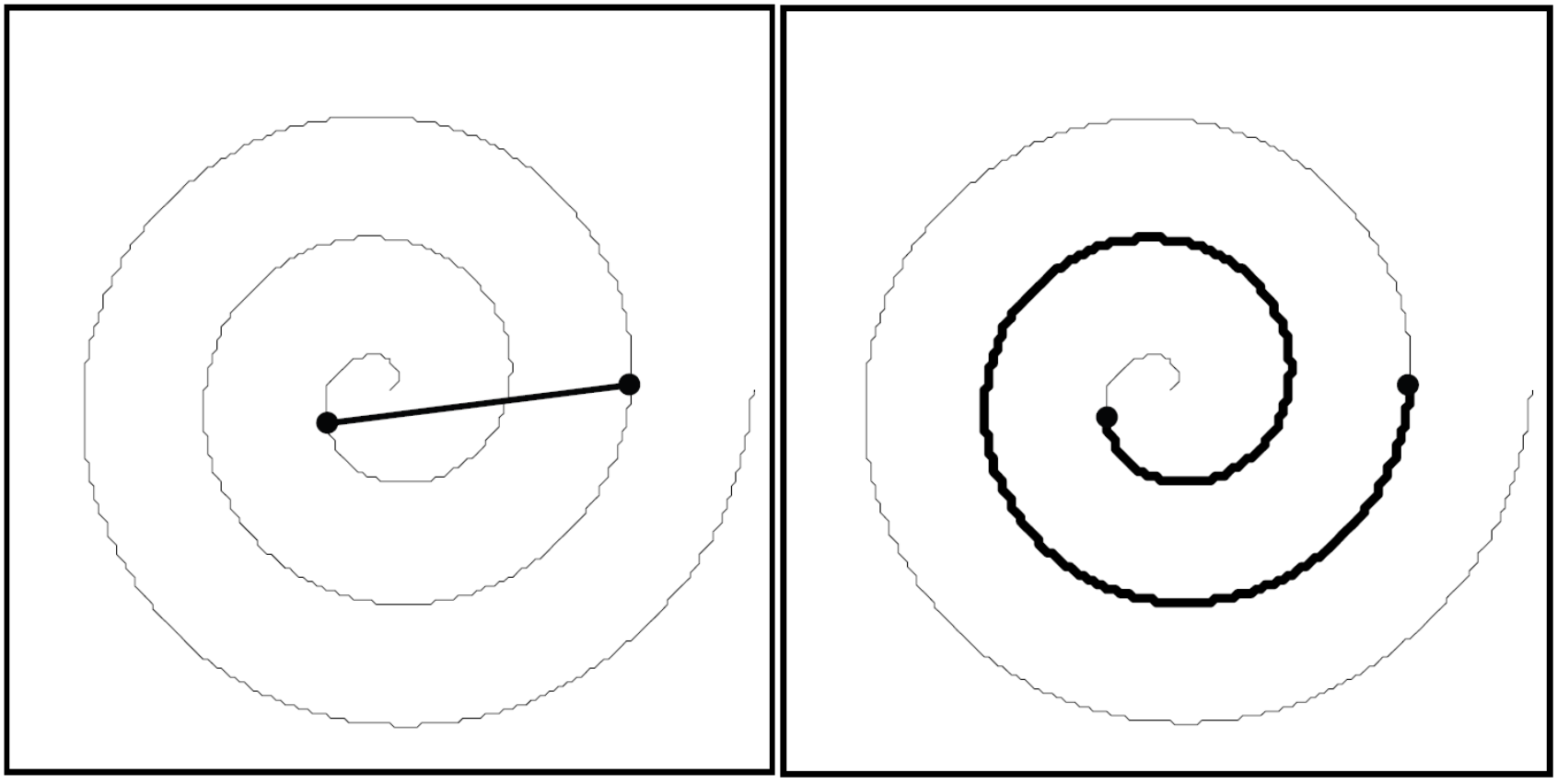
Isometric feature mapping (ISOMAP)

- *Isometric feature mapping* (ISOMAP) can be assigned to the group of multidimensional scaling. ISOMAP is designed for dimensionality reduction as well as for visualization of multidimensional data. An assumption is made that the multidimensional points are located on a lower-dimensional manifold. Therefore geodesic distances are used as a measure of proximity between the multidimensional points.
- Usually Euclidean distances between the points (as a proximity measure) are used in multidimensional scaling. In this case, the existence of a manifold is not taken into consideration.
- In ISOMAP, the geodesic distance is a proximity measure between the multidimensional points. A *geodesic distance* is the length of the shortest path between two points along the surface of a manifold.

Geodesic Distances

- In order to compute the geodesic distances between n-dimensional points from $\{X_1, X_2, \ldots, X_m\}$, it is necessary to build a weighted graph over the points that are vertices of the graph. The vertices, corresponding to the neighboring points, are connected using edges.
- The neighborhood of the point X_i can be defined:
  1. by a fixed number of the nearest points,
  2. by all the points within some fixed distance from X_i.
- The weights of edges are Euclidean distances between the corresponding points.
- Using one of the algorithms for the shortest path in the graph, for example Dijkstra's algorithm, the shortest path length between the pair of points is computed. This length is an estimate of the geodesic distance between the points.
- The matrix of geodesic distances between all multidimensional points is formed. This matrix defines dissimilarities between the objects. It can be used as data for multidimensional scaling.

Euclidean and Geodesic Distances



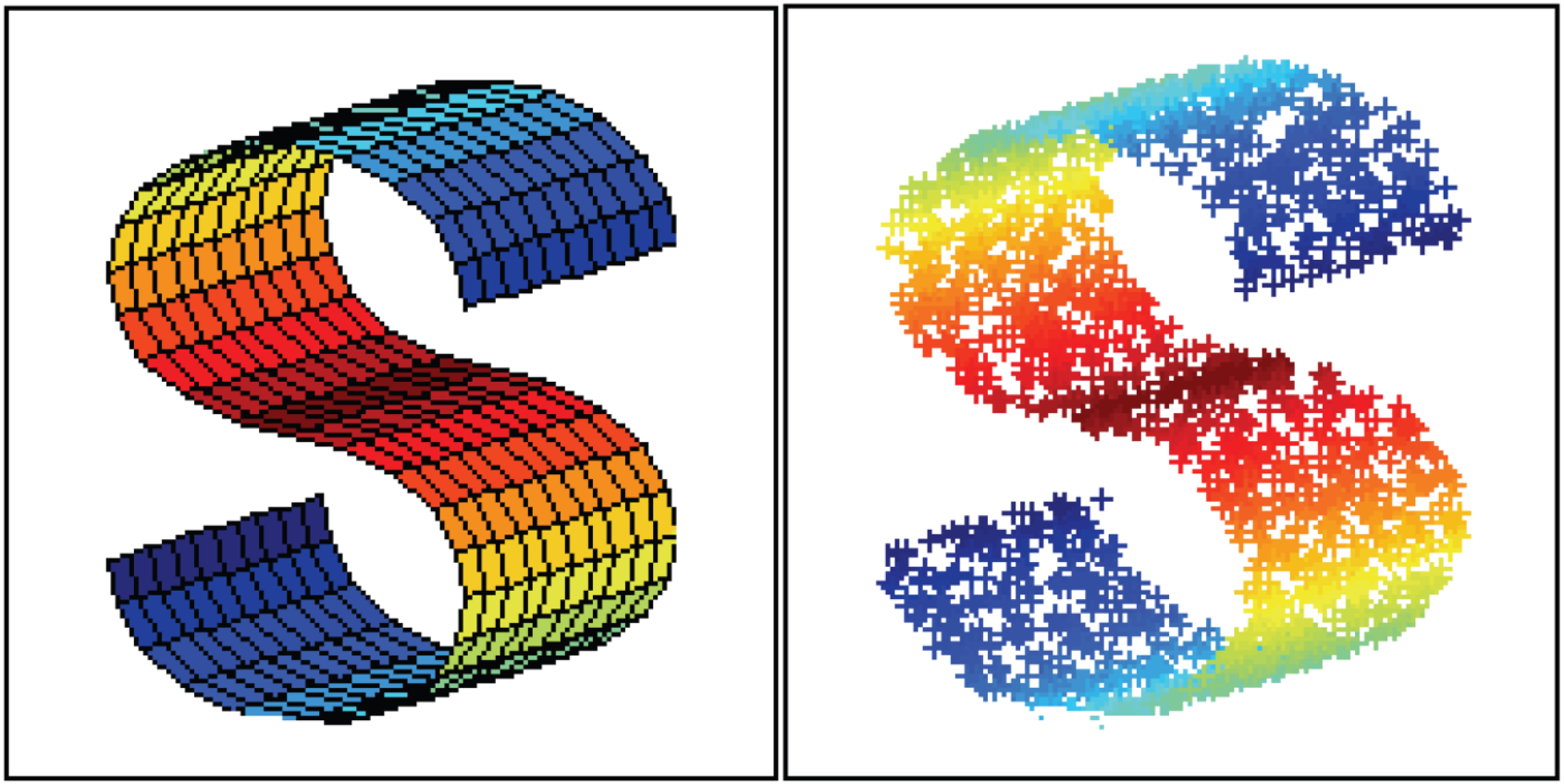
ISOMAP Algorithm

- The ISOMAP algorithm can be summarized as follows:

1. The neighbors of each multidimensional point are chosen from $\{X_1, X_2, \ldots, X_m\}$.
2. A weighted graph is constructed.
3. The geodesic distances between the pairs of all points are computed; a dissimilarity matrix is formed.
4. The projection of multidimensional points to a low-dimensional space (projection space) is obtained by multidimensional scaling.

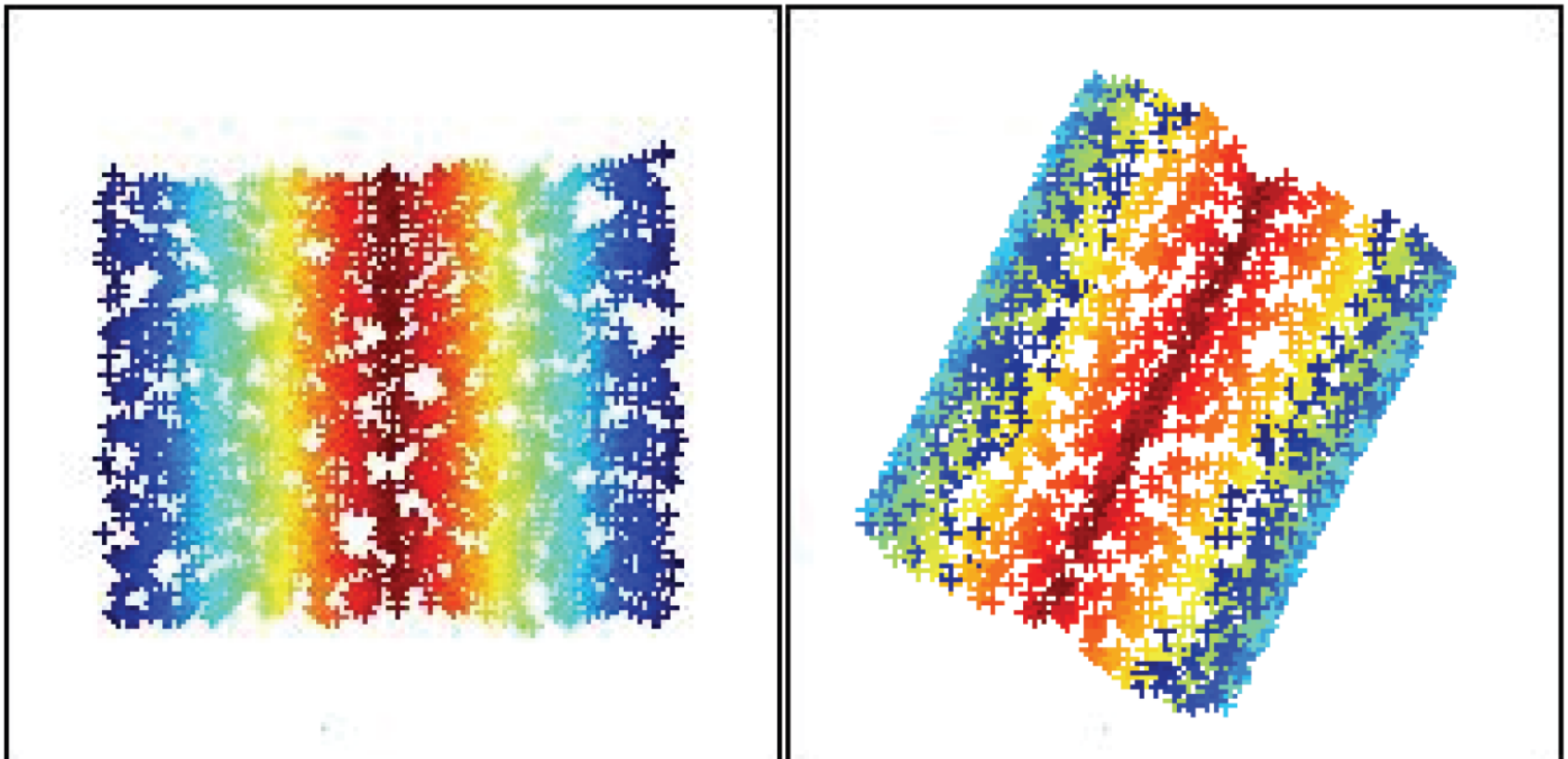
S-manifold

- The S-manifold is presented, $n = 3$. The points on the manifold are also shown, $m = 1000$.



ISOMAP and MDS Projections of S-manifold

- The structure of the manifold is well preserved by ISOMAP, because the S-manifold is unfolded: the farthest points on the manifold remain the farthest ones on the projection.
- The farthest points obtained by MDS are pale blue. These points are the farthest in multidimensional space in the sense of Euclidean distances, but they are not farthest in the sense of geodesic distances on the S-manifold.

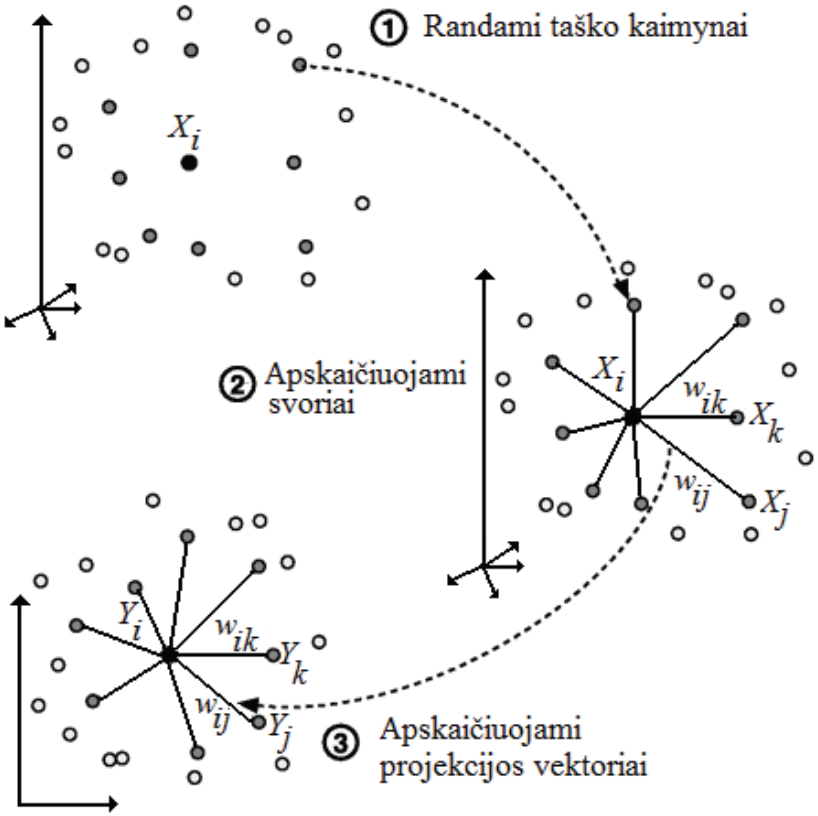


Locally Linear Embedding

- *Locally linear embedding* (LLE) is a nonlinear method for dimensionality reduction and manifold learning.
- Given a set of data points distributed on a manifold in a multidimensional space, LLE is able to project the data to a low-dimensional space by unfolding the manifold.
- LLE works by assuming that the manifold is well sampled, i.e. there are enough data, each data point and its neighbors lie on or close to a locally linear patch.
- Therefore, a data point can be approximated as a weighted linear combination of its neighbors. The basic idea of LLE is that such a linear combination is invariant under linear transformations (translation, rotation, and scaling) and, therefore, it should remain unchanged after the manifold has been unfolded to a low-dimensional space.
- The low-dimensional configuration of data points is obtained by solving two least squares optimization problems.

Scheme of Locally Linear Embedding

- The LLE algorithm transforms the set X of n-dimensional points X_i, $i = 1, \ldots, m$ ($X_i \in \mathbb{R}^n$) to a set Y of d-dimensional points Y_i, $i = 1, \ldots, m$ ($Y_i \in \mathbb{R}^d$).



LLE Algorithm

- The LLE algorithm consists of three steps.
- In the *first step*, we identify k neighbors of each data point X_i.
- As the neighbors, k-nearest points from the set X may be chosen according to the Euclidean distance.

LLE Algorithm: Second Step

- In the *second step*, we compute the weights w_{ij} that reconstruct each n-dimensional point X_i best from its neighbors minimizing the following error function:

$$E_{LLE}(W) = \sum_{i=1}^{m} \left\| X_i - \sum_{j=1}^{m} w_{ij} X_j \right\|^2 .$$

where $W = \{w_{ij},\ i, j = 1, \ldots, m\}$; $w_{ij} = 0$, if X_i and X_j are not neighbors; $\sum_{j=1}^{m} w_{ij} = 1$; $\|\cdot\|$ is the Euclidean distance. This is a typical least squares optimization problem, the minimum of which can be found by solving a linear system of equations.

LLE Algorithm: Third Step

- In the *third step*, we map each data point X_i to a low-dimensional point Y_i, which best preserves a multidimensional neighborhood geometry, represented by the weights w_{ij}. So, the weights are fixed and coordinates of low-dimensional points are sought by minimizing the following function:

$$E_{LLE}(Y) = \sum_{i=1}^{m} \left\| Y_i - \sum_{j=1}^{m} w_{ij} Y_j \right\|^2$$

subject to:

$$\sum_{i=1}^{m} Y_i = 0 \text{ and } \frac{1}{m} \sum_{i=1}^{m} Y_i^T Y_i = I,$$

where I is the identity matrix consisting of d rows and d columns.

LLE Algorithm: Third Step

- The most straightforward method for computing the d-dimensional coordinates ($d < n$) is to find the bottom $d + 1$ eigenvectors of the sparse matrix

$$\bar{M} = (I - W)^T (I - W)^T ,$$

  where I is the identity matrix, consisting of m rows and m columns.
- These eigenvectors are associated with the $d + 1$ smallest eigenvalues of $\bar{M}$. The bottom eigenvector, the eigenvalue of which is closest to zero, is the unit vector with all equal components and it is discarded. The remaining d eigenvectors form the d embedding coordinates of points Y_i.

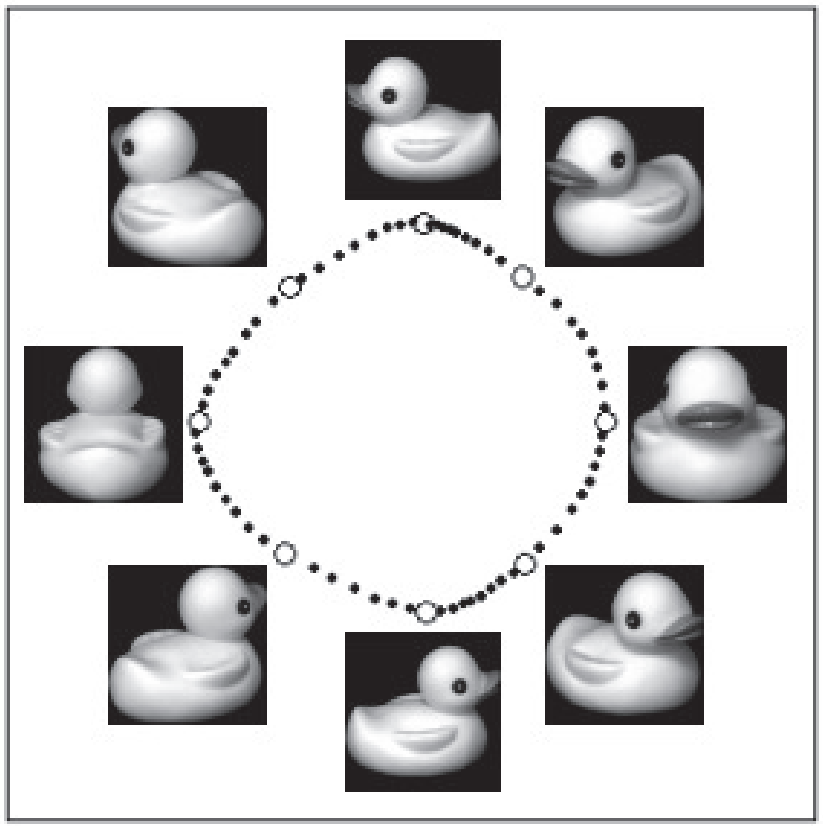
The Number of Nearest Neighbors

- The most important step to success of LLE is to choose the proper number k of the nearest neighbors for each data point. The mapping quality is rather sensitive to this parameter.
- If k is too small, a continuous manifold can falsely be divided into disjoint sub-manifolds. In this way, the mapping does not reflect any global properties.
- The large number of the nearest neighbors k causes smoothing or elimination of small-scale structures in the manifold, the mapping looses its specific character, and behaves like PCA and MDS.

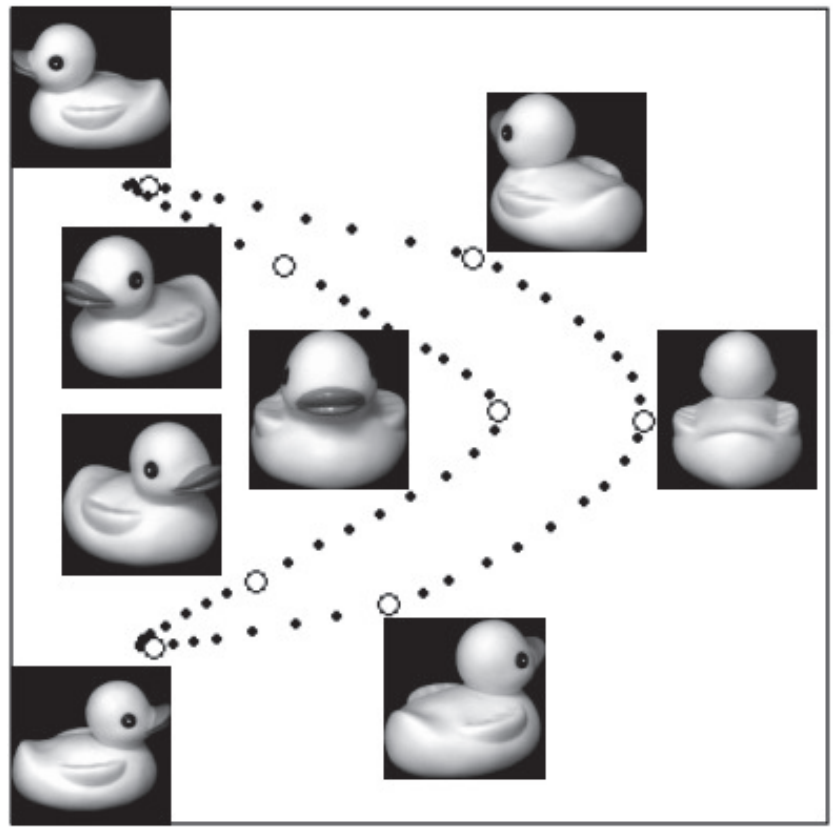
Application of LLE

- One of the applications of the LLE method is the analysis of images. For example, the data set consists of pictures of the same moving object. Features of pictures are color parameters of pixels. Since the number of pixels in the picture is usually large, the dimensionality of the analyzed data is very large.
- For example, a set of uncolored pictures, obtained by gradually (by 5°) rotating a duckling around, was analyzed. The number of pictures is $m = 72$. The pictures consist of 128×128 gray-scale pixels, therefore the dimensionality of data is $n = 16384$. Intuitively, one would expect multidimensional data that represent these pictures, to lie on a manifold parameterized by a rotation angle.
- The results of LLE are presented. Since the duckling was gradually turned around, a better representation is obtained as $k = 4$, because, in this case, we see the points on a circle.

Visualization of Pictures of a Rotating Duckling by LLE



$k = 4$



$k = 9$

Principal Curves

- Principal curves are a nonlinear generalization of principal components.
- The idea of principal curves is to find a nonlinear fit to the localized centers of some given data in n-dimensional space.
- Principal curve is a smooth one-dimensional curve that passes through the middle of the n-dimensional data: every point in the curve is the mean of the points that project onto this point.
- Principal curve minimizes the distance from the points, and provides a non-linear summary of the data.
- The principal curves approximate the data points better than PCA or linear regression.
- The idea of the principal curves can be extended to principal surfaces.

Example of Principal Curve