Manifold Learning

4.1-4.3 Introduction

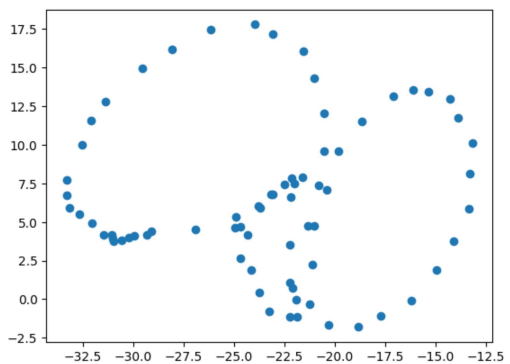
20230112

Jiahao Lai

Abstract Manifold learning methods are one of the most exciting developments in machine learning in recent years. The central idea underlying these methods is that although natural data is typically represented in very high-dimensional spaces, the process generating the data is often thought to have relatively few degrees of freedom. A natural mathematical characterization of this intuition is to model the data as lying on or near a low-dimensional manifold embedded in a higher dimensional space.

In this chapter, we discuss the problem of nonlinear dimensionality reduction (NLDR): the task of recovering meaningful low-dimensional structures hidden in high-dimensional data. In many cases of interest, the observed data are found to lie on an embedded submanifold of the high-dimensional space, e.g., images generated by different views of the same three-dimensional (3D) object. Intuitively,

Figure 2: Tomato pictures token from 0, 50, and 100 degree angles. In total, there are 72($= \frac{360}{5}$) different angles.



Manifold:

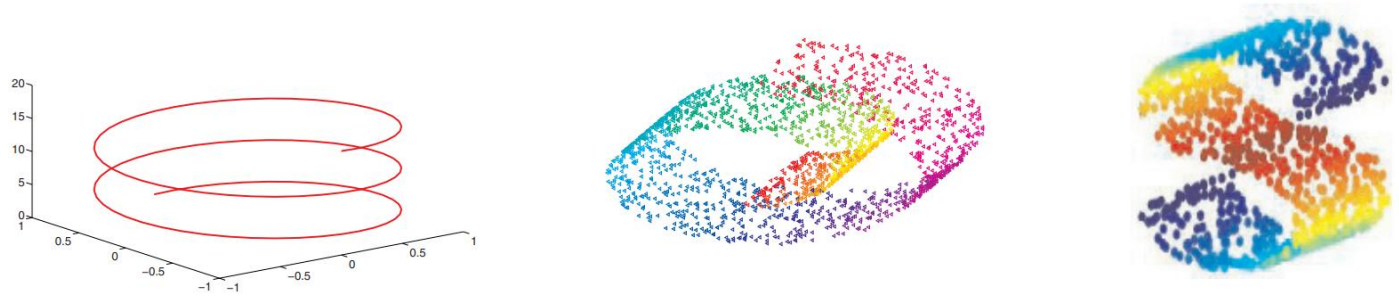


Fig. 4.1 Manifold examples. Left: A curve in $\mathbb{R}^3$ can be considered as a 1D manifold embedded in the 3D Euclidean space. Middle: the set of points sampled from Swiss roll manifold. The set of points sampled from S-shaped manifold

Manifold: A d-dimensional manifold $\mathbb{M}$ has the property that it is locally homeomorphic with respect to $\mathbb{R}^d$. That is, for each $x \in \mathbb{R}^d$, there is an open neighborhood around x, N_x, and a homeomorphism $f: N_x \to \mathbb{R}^d$. These neighborhoods are referred to as coordinate patches, and the map is referred to as a coordinate chart. The image of the coordinate chart is referred to as the parameter space.

Embedding: an embedding of a manifold $\mathbb{M}$ into $\mathbb{R}^D$ is a smooth homeomorphism from $\mathbb{M}$ to a subset of $\mathbb{R}^D$. The algorithms discussed in this chapter find embeddings of discrete point sets, by which we simply mean a mapping of the point set into another space (typically a lower dimensional Euclidean space). An embedding of a dissimilarity matrix into $\mathbb{R}^D$ is a configuration of points whose interpoint distances match those given in the matrix.

Graph embedding: the embedding of a graph G on a surface $\mathbb{M}$ is a representation of G on $\mathbb{M}$ in which the points of $\mathbb{M}$ are associated with vertices and simple arcs (homeomorphic images of [0,1]) and are associated with edges in such a way that:

- the end points of the arc associated with an edge e are the points associated with the end vertices of e;
- an arc includes no points associated with other vertices;
- two arcs never intersect at a point which is interior to either of the arcs.

Manifold learning in dimensionality reduction：

- In the classical approaches to dimensionality reduction, various methods generate linear/nonlinear maps like kernel PCA, kernel Fisher discriminant. They don’t explicitly consider the structure of the manifold on which the data may possibly reside.
- Manifold learning methods address the dimensionality reduction problem by **uncovering the intrinsic low dimensional geometric structure(local or global geometry)** hidden in their high-dimensional observations and constructing a representation of the data in low-dimensional space.

How to construct a map using certain properties of the manifold?

- Global Methods: preserve metircs at all scale
  MDS(preserve inner product or Euclidean distance),
  Isomap(preserve geodesic distance)
- Local Methods: preseve local geometry of the data
  LLE(find optimal linear reconstruction in a neighborhood),
  Laplacian eigenmaps(embed vertices into a gragh and use graph Laplacian to derive a smooth mapping)
- Hybrid Methods: make use of both local and global properties of the manifold
  LTSA

denote $\mathbb{R}^{d\times n} \ni \boldsymbol{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ and $\mathbb{R}^{p\times n} \ni \boldsymbol{Y} := [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n]$.

Classical MDS : preserves the similarity of the data.
Similarity measure of points: **inner product**
Cost function: **STRAIN**

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_1 := \sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{x}_i^\top \boldsymbol{x}_j - \boldsymbol{y}_i^\top \boldsymbol{y}_j)^2, \tag{1}$$

whose matrix form is:

$$\underset{\boldsymbol{Y}}{\text{minimize}} \quad c_1 = ||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2, \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\boldsymbol{X}^\top \boldsymbol{X}$ and $\boldsymbol{Y}^\top \boldsymbol{Y}$ are the Gram matrices of the original data $\boldsymbol{X}$ and the embedded data $\boldsymbol{Y}$, respectively.

The objective function, in Eq. (2), is simplified as:

$$\begin{aligned}
&||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2 \\
&\quad = \mathbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})^\top (\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})\big] \\
&\quad = \mathbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})\big] \\
&\quad = \mathbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})^2\big],
\end{aligned}$$

$$\boldsymbol{X}^\top \boldsymbol{X} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top, \tag{3}$$

$$\boldsymbol{Y}^\top \boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top, \tag{4}$$

$$\begin{aligned}
\therefore \quad &||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2 \\
&= \mathbf{tr}\big[(\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y})^2\big] \\
&= \mathbf{tr}\big[(\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top - \boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top)^2\big] \\
&\overset{(a)}{=} \mathbf{tr}\big[(\boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top - \boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top\boldsymbol{V}\boldsymbol{V}^\top)^2\big] \\
&= \mathbf{tr}\Big[\big(\boldsymbol{V}(\boldsymbol{\Delta} - \boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top\boldsymbol{V})\boldsymbol{V}^\top\big)^2\Big] \\
&= \mathbf{tr}\Big[\boldsymbol{V}^2(\boldsymbol{\Delta} - \boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top\boldsymbol{V})^2(\boldsymbol{V}^\top)^2\Big] \\
&\overset{(b)}{=} \mathbf{tr}\Big[(\boldsymbol{V}^\top)^2\boldsymbol{V}^2(\boldsymbol{\Delta} - \boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top\boldsymbol{V})^2\Big] \\
&= \mathbf{tr}\Big[(\underbrace{\boldsymbol{V}^\top\boldsymbol{V}}_{\boldsymbol{I}})^2(\boldsymbol{\Delta} - \boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top\boldsymbol{V})^2\Big] \\
&\overset{(c)}{=} \mathbf{tr}\Big[(\boldsymbol{\Delta} - \boldsymbol{V}^\top\boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top\boldsymbol{V})^2\Big],
\end{aligned}$$

Let $\mathbb{R}^{n\times n} \ni \boldsymbol{M} := \boldsymbol{V}^\top\boldsymbol{Q}$, so:

$$||\boldsymbol{X}^\top \boldsymbol{X} - \boldsymbol{Y}^\top \boldsymbol{Y}||_F^2 = \mathbf{tr}\Big[(\boldsymbol{\Delta} - \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2\Big].$$

$$\begin{aligned}
c_1 &= \mathbf{tr}\Big[(\boldsymbol{\Delta} - \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2\Big] \\
&= \mathbf{tr}(\boldsymbol{\Delta}^2 + (\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2 - 2\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top) \\
&= \mathbf{tr}(\boldsymbol{\Delta}^2) + \mathbf{tr}((\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2) - 2\,\mathbf{tr}(\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top).
\end{aligned}$$

$$\mathbb{R}^{n\times n} \ni \frac{\partial c_1}{\partial \boldsymbol{M}} = 2(\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)\boldsymbol{M}\boldsymbol{\Psi} - 2\boldsymbol{\Delta}\boldsymbol{M}\boldsymbol{\Psi} \overset{\text{set}}{=} \boldsymbol{0}$$
$$\implies (\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)(\boldsymbol{M}\boldsymbol{\Psi}) = (\boldsymbol{\Delta})(\boldsymbol{M}\boldsymbol{\Psi})$$
$$\overset{(a)}{\implies} \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top = \boldsymbol{\Delta}, \tag{5}$$

where (a) is because $\boldsymbol{M}\boldsymbol{\Psi} \neq \boldsymbol{0}$.

$$\mathbb{R}^{n\times n} \ni \frac{\partial c_1}{\partial \boldsymbol{\Psi}} = 2\boldsymbol{M}^\top(\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)\boldsymbol{M} - 2\boldsymbol{M}^\top\boldsymbol{\Delta}\boldsymbol{M} \overset{\text{set}}{=} \boldsymbol{0}$$
$$\implies \boldsymbol{M}^\top(\boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)\boldsymbol{M} = \boldsymbol{M}^\top(\boldsymbol{\Delta})\boldsymbol{M}$$
$$\overset{(a)}{\implies} \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top = \boldsymbol{\Delta}, \tag{6}$$

whose one possible solution is:

$$\boldsymbol{M} = \boldsymbol{I}, \tag{7}$$
$$\boldsymbol{\Psi} = \boldsymbol{\Delta}. \tag{8}$$

which means that the minimum value of the non-negative objective function $\mathbf{tr}((\boldsymbol{\Delta} - \boldsymbol{M}\boldsymbol{\Psi}\boldsymbol{M}^\top)^2)$ is zero.
We had $\boldsymbol{M} = \boldsymbol{V}^\top\boldsymbol{Q}$. Therefore, according to Eq. (7), we have:

$$\therefore \quad \boldsymbol{V}^\top\boldsymbol{Q} = \boldsymbol{I} \implies \boldsymbol{Q} = \boldsymbol{V}. \tag{9}$$

According to Eq. (4), we have:

$$\boldsymbol{Y}^\top\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{\Psi}\boldsymbol{Q}^\top \overset{(a)}{=} \boldsymbol{Q}\boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{Q}^\top \implies \boldsymbol{Y} = \boldsymbol{\Psi}^{\frac{1}{2}}\boldsymbol{Q}^\top$$
$$\overset{(8),(9)}{\implies} \boldsymbol{Y} = \boldsymbol{\Delta}^{\frac{1}{2}}\boldsymbol{V}^\top, \tag{10}$$

In summary, for embedding $\boldsymbol{X}$ using classical MDS, the eigenvalue decomposition of $\boldsymbol{X}^\top\boldsymbol{X}$ is obtained as in Eq. (3). Then, using Eq. (10), $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ is obtained. Truncating this $\boldsymbol{Y}$ to have $\boldsymbol{Y} \in \mathbb{R}^{p\times n}$, with the first (top) p rows, gives us the p-dimensional embedding of the n points. Note that the leading p columns are used because singular values are sorted from largest to smallest in SVD which can be used for Eq. (3).

Kernel MDS / generalized classical MDS:

$$\begin{aligned} d_{ij}^2 &= ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_j) \\ &= \boldsymbol{x}_i^\top\boldsymbol{x}_i - \boldsymbol{x}_i^\top\boldsymbol{x}_j - \boldsymbol{x}_j^\top\boldsymbol{x}_i + \boldsymbol{x}_j^\top\boldsymbol{x}_j \\ &= \boldsymbol{x}_i^\top\boldsymbol{x}_i - 2\boldsymbol{x}_i^\top\boldsymbol{x}_j + \boldsymbol{x}_j^\top\boldsymbol{x}_j = \boldsymbol{G}_{ii} - 2\boldsymbol{G}_{ij} + \boldsymbol{G}_{jj}, \end{aligned}$$

where $\mathbb{R}^{n\times n} \ni \boldsymbol{G} := \boldsymbol{X}^\top\boldsymbol{X}$ is the Gram matrix. If $\mathbb{R}^n \ni \boldsymbol{g} := [\boldsymbol{g}_1, \dots, \boldsymbol{g}_n] = [\boldsymbol{G}_{11}, \dots, \boldsymbol{G}_{nn}] = \mathbf{diag}(\boldsymbol{G})$, we have:

$$d_{ij}^2 = \boldsymbol{g}_i - 2\boldsymbol{G}_{ij} + \boldsymbol{g}_j,$$
$$\boldsymbol{D} = \boldsymbol{g}\boldsymbol{1}^\top - 2\boldsymbol{G} + \boldsymbol{1}\boldsymbol{g}^\top = \boldsymbol{1}\boldsymbol{g}^\top - 2\boldsymbol{G} + \boldsymbol{g}\boldsymbol{1}^\top,$$

where $\boldsymbol{1}$ is the vector of ones and $\boldsymbol{D}$ is the distance matrix with squared Euclidean distance (d_{ij}^2 as its elements). Let $\mathbb{R}^{n\times n} \ni \boldsymbol{H} := \boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top$ denote the centering matrix. We

double-center the matrix $\boldsymbol{D}$ as follows (Oldford, 2018):

$$
\begin{aligned}
\boldsymbol{HDH} &= (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{D}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) \\
&= (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)(\mathbf{1}\boldsymbol{g}^\top - 2\boldsymbol{G} + \boldsymbol{g}\mathbf{1}^\top)(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) \\
&= \big[\underbrace{(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\mathbf{1}}_{=\mathbf{0}}\boldsymbol{g}^\top - 2(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{G} \\
&\quad + (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{g}\mathbf{1}^\top\big](\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) \\
&= -2(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{G}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) \\
&\quad + (\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{g}\underbrace{\mathbf{1}^\top(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)}_{=\mathbf{0}} \\
&= -2(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\boldsymbol{G}(\boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top) = -2\,\boldsymbol{HGH}
\end{aligned}
$$

$$
\therefore \qquad \boldsymbol{HGH} = \boldsymbol{HX}^\top\boldsymbol{XH} = -\frac{1}{2}\boldsymbol{HDH}. \tag{11}
$$

If data $\boldsymbol{X}$ are already centered, i.e., the mean has been removed ($\boldsymbol{X} \leftarrow \boldsymbol{XH}$), Eq. (11) becomes:

$$
\boldsymbol{X}^\top\boldsymbol{X} = -\frac{1}{2}\boldsymbol{HDH}. \tag{12}
$$

use kernel matrix rather than gram matrix:

$$
\mathbb{R}^{n\times n} \ni \boldsymbol{K} = -\frac{1}{2}\boldsymbol{HDH}. \tag{13}
$$

$$
\boldsymbol{K} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top. \tag{14}
$$

Then, using Eq. (10), $\boldsymbol{Y} \in \mathbb{R}^{n\times n}$ is obtained. It is noteworthy that in this case, we are replacing $\boldsymbol{X}^\top\boldsymbol{X}$ with the kernel $\boldsymbol{K} = \boldsymbol{\Phi}(\boldsymbol{X})^\top\boldsymbol{\Phi}(\boldsymbol{X})$ and then, according to Eqs. (10) and (14), we have:

$$
\boldsymbol{K} = \boldsymbol{Y}^\top\boldsymbol{Y}. \tag{15}
$$

Truncating the $\boldsymbol{Y}$, obtained from Eq. (10), to have $\boldsymbol{Y} \in \mathbb{R}^{p\times n}$, with the first (top) p rows, gives us the p-dimensional embedding of the n points. It is noteworthy that, because of using kernel in the generalized classical MDS, one can name it the *kernel classical MDS*.

Metric MDS : preserves the distance of points rather than the similarity.
Cost function: **stress/sstress function**

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_2 := \left(\frac{\sum_{i=1}^n \sum_{j=1,j<i}^n \left(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)\right)^2}{\sum_{i=1}^n \sum_{j=1,j<i}^n d_x^2(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right)^{\frac{1}{2}}, \tag{16}$$

or, without the normalization factor:

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_2 := \left(\sum_{i=1}^n \sum_{j=1,j<i}^n \left(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)\right)^2 \right)^{\frac{1}{2}}, \tag{17}$$

where $d_x(.,.)$ and $d_y(.,.)$ denote the distance metrics in the input and the embedded spaces, respectively.

the matrix form: $||\mathbf{D} - \mathbf{D}'||_F^2$

Other criteria for MDS have also been studied, for example, the STRESS and SSTRESS given by $||\mathbf{D} - \mathbf{D}'||_F^2$ and $||\mathbf{D}^2 - \mathbf{D}'^2||_F^2$, respectively. However, the

Use gradient descent or Newton's method to solve (16) since it doesn't have closed-form solution.

Non-Metric MDS :

In *non-metric MDS*, rather than using a distance metric, $d_y(\boldsymbol{x}_i, \boldsymbol{x}_j)$, for the distances between points in the embedding space, we use $f(d_y(\boldsymbol{x}_i, \boldsymbol{x}_j))$ where $f(.)$ is a non-parametric monotonic function. In other words, only the order of dissimilarities is important rather than the amount of dissimilarities (Agarwal et al., 2007; Jung, 2013):

$$\begin{aligned} d_y(\boldsymbol{y}_i, \boldsymbol{y}_j) \leq d_y(\boldsymbol{y}_k, \boldsymbol{y}_\ell) \iff \\ f(d_y(\boldsymbol{y}_i, \boldsymbol{y}_j)) \leq f(d_y(\boldsymbol{x}_k, \boldsymbol{y}_\ell)). \end{aligned} \tag{20}$$

The optimization in non-metric MDS is (Agarwal et al., 2007):

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_3 := \left(\frac{\sum_{i=1}^n \sum_{j=1,j<i}^n \left(d_x(\boldsymbol{x}_i, \boldsymbol{x}_j) - f(d_y(\boldsymbol{y}_i, \boldsymbol{y}_j))\right)^2}{\sum_{i=1}^n \sum_{j=1,j<i}^n d_x^2(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right)^{\frac{1}{2}}. \tag{21}$$

Isometric Feature Mapping(Isomap):
In classical MDS, we use Euclidean distance

$$\underset{\{\boldsymbol{y}_i\}_{i=1}^n}{\text{minimize}} \quad c_1 := \sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{x}_i^\top \boldsymbol{x}_j - \boldsymbol{y}_i^\top \boldsymbol{y}_j)^2,$$

$$\boldsymbol{X}^\top \boldsymbol{X} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}.$$

which is not suitable for a nonliearly embedded manifold. Thus in kernel MDS, we use kernel matrix to substitute the gram matrix.

$$\mathbb{R}^{n\times n} \ni \boldsymbol{K} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}\boldsymbol{H}.$$

Isomap is a special case of kernel MDS. Isomap uses an approximation of the geodesic distance in above matirx D.
Geodesic distance: the length of shortest path between two points on the possibly curvy manifold.

Calculation of the geodesic distance is very difficult because it requires the knowledge of differential geometry. Therefore, Isomap approximates the geodesic distance by piecewise Euclidean distances.
The shortest path is found using algorithms such as the Dijkstra algorithm or the Floyd-Warshal algorithm.

$$\boldsymbol{D}_{ij}^{(g)} := \min_{\boldsymbol{r}} \sum_{i=2}^{l} \|\boldsymbol{r}_i - \boldsymbol{r}_{i+1}\|_2,$$

$$\mathbb{R}^{n\times n} \ni \boldsymbol{K} = -\frac{1}{2}\boldsymbol{H}\boldsymbol{D}^{(g)}\boldsymbol{H}.$$

1. Find the neighborhood for each point $\mathbf{x}_i$.
2. Compute the shortest path distance between all pairs of points using Dijkstra's or Floyd's algorithm, and store the squares of these distances in a matrix.
3. Apply classical MDS to the distance matrix.

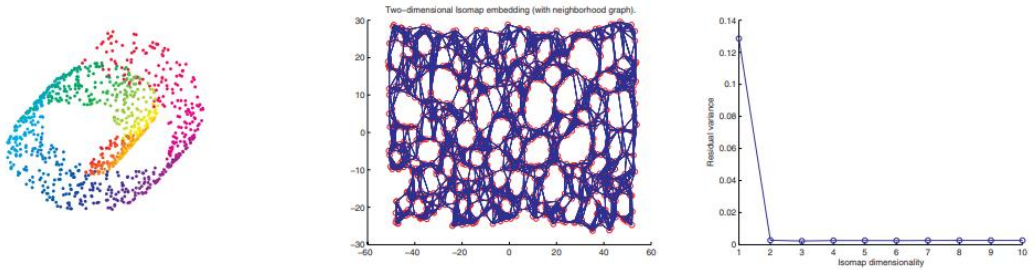


Fig. 4.4 The Swiss roll data set, illustrating the 2D embedding recovered by the Isomap in step three, and how the residual variance decreases as the dimensionality d is increased. Left: the true manifold; middle: the 2D embedding recovered by the Isomap with neighborhood graph overlaid; right: the residual variance decreases as dimensionality increases

Landmark Isomap:

One drawback of Isomap is its quadraic memory requirement: the geodesic distance matrix is dense,making the Isomap infeasible for large data set.

Nystrom Approximation:

Nystrom approximation is a technique used to approximate a positive semi-definite matrix using merely a subset of its columns (or rows) (Williams & Seeger, 2001). Consider a positive semi-definite matrix $\mathbb{R}^{n\times n} \ni \boldsymbol{K} \succeq 0$ whose parts are:

$$\mathbb{R}^{n\times n} \ni \boldsymbol{K} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{B}^\top & \boldsymbol{C} \end{array}\right], \tag{52}$$

where $\boldsymbol{A} \in \mathbb{R}^{m\times m}$, $\boldsymbol{B} \in \mathbb{R}^{m\times(n-m)}$, and $\boldsymbol{C} \in \mathbb{R}^{(n-m)\times(n-m)}$ in which $m \ll n$.

K is the similarity matrix or distance matrix. Instead of computing K, we designate m data points to be landmark points and compute A and B. Our goal is to approximate C.

As the matrix $\boldsymbol{K}$ is positive semi-definite, by definition, it can be written as $\boldsymbol{K} = \boldsymbol{O}^\top \boldsymbol{O}$. If we take $\boldsymbol{O} = [\boldsymbol{R}, \boldsymbol{S}]$ where $\boldsymbol{R}$ are the selected columns (landmarks) of $\boldsymbol{O}$ and $\boldsymbol{S}$ are the other columns of $\boldsymbol{O}$. We have:

$$\boldsymbol{K} = \boldsymbol{O}^\top \boldsymbol{O} = \begin{bmatrix} \boldsymbol{R}^\top \\ \boldsymbol{S}^\top \end{bmatrix} [\boldsymbol{R}, \boldsymbol{S}] \tag{53}$$

$$= \begin{bmatrix} \boldsymbol{R}^\top \boldsymbol{R} & \boldsymbol{R}^\top \boldsymbol{S} \\ \boldsymbol{S}^\top \boldsymbol{R} & \boldsymbol{S}^\top \boldsymbol{S} \end{bmatrix} \overset{(52)}{=} \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^\top & \boldsymbol{C} \end{bmatrix}. \tag{54}$$

Hence, we have $\boldsymbol{A} = \boldsymbol{R}^\top \boldsymbol{R}$. The eigenvalue decomposition (Ghojogh et al., 2019a) of $\boldsymbol{A}$ gives:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top \tag{55}$$

$$\implies \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top \implies \boldsymbol{R} = \boldsymbol{\Sigma}^{(1/2)}\boldsymbol{U}^\top. \tag{56}$$

Moreover, we have $\boldsymbol{B} = \boldsymbol{R}^\top \boldsymbol{S}$ so we have:

$$\boldsymbol{B} = (\boldsymbol{\Sigma}^{(1/2)}\boldsymbol{U}^\top)^\top \boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}^{(1/2)}\boldsymbol{S}$$

$$\overset{(a)}{\implies} \boldsymbol{U}^\top \boldsymbol{B} = \boldsymbol{\Sigma}^{(1/2)}\boldsymbol{S} \implies \boldsymbol{S} = \boldsymbol{\Sigma}^{(-1/2)}\boldsymbol{U}^\top \boldsymbol{B}, \tag{57}$$

$$\boldsymbol{C} = \boldsymbol{S}^\top \boldsymbol{S} = \boldsymbol{B}^\top \boldsymbol{U}\boldsymbol{\Sigma}^{(-1/2)}\boldsymbol{\Sigma}^{(-1/2)}\boldsymbol{U}^\top \boldsymbol{B}$$

$$= \boldsymbol{B}^\top \boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top \boldsymbol{B} \overset{(55)}{=} \boldsymbol{B}^\top \boldsymbol{A}^{-1}\boldsymbol{B}. \tag{58}$$

Therefore, Eq. (52) becomes:

$$\boldsymbol{K} \approx \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{B}^\top & \boldsymbol{B}^\top \boldsymbol{A}^{-1}\boldsymbol{B} \end{array}\right]. \tag{59}$$

After computing K, we need to do eigenvalue decomposition of the kernel matrix and compute Y.

$$\boldsymbol{K} = \boldsymbol{V}\boldsymbol{\Delta}\boldsymbol{V}^\top \qquad \boldsymbol{K} = \boldsymbol{Y}^\top \boldsymbol{Y}. \qquad \boldsymbol{Y} = \boldsymbol{\Delta}^{\frac{1}{2}}\boldsymbol{V}^\top$$

Recall that Eq. (14) decomposes the kernel matrix into eigenvectors and then Eq. (10) embeds data. However, for big data, the eigenvalue decomposition of kernel matrix is intractable. Therefore, using Eq. (55), we decompose an $m \times m$ submatrix of kernel. Comparing Eqs. (15) and (53) shows that:

$$\mathbb{R}^{n\times n} \ni \boldsymbol{Y} = [\boldsymbol{R}, \boldsymbol{S}] \overset{(a)}{=} [\boldsymbol{\Sigma}^{(1/2)}\boldsymbol{U}^\top, \boldsymbol{\Sigma}^{(-1/2)}\boldsymbol{U}^\top \boldsymbol{B}], \tag{60}$$