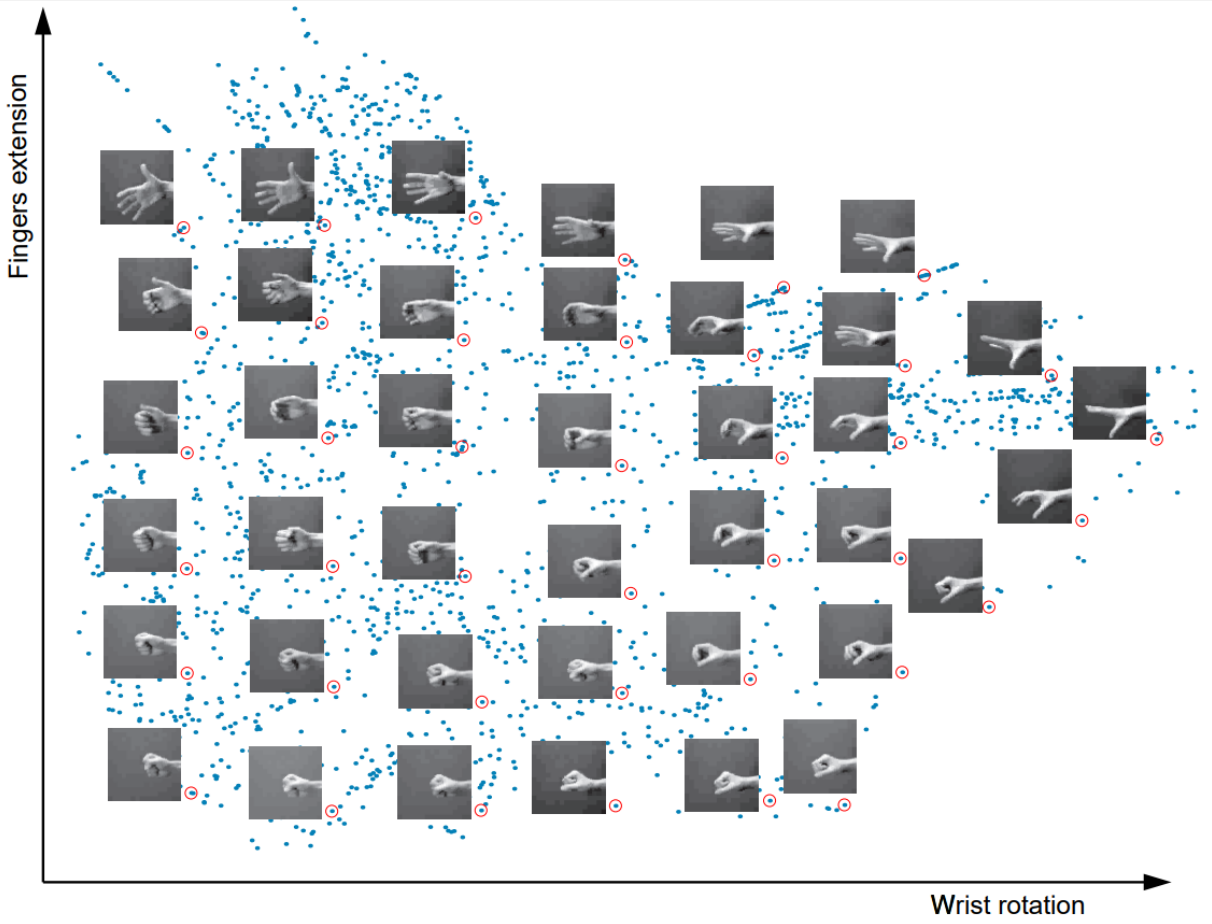
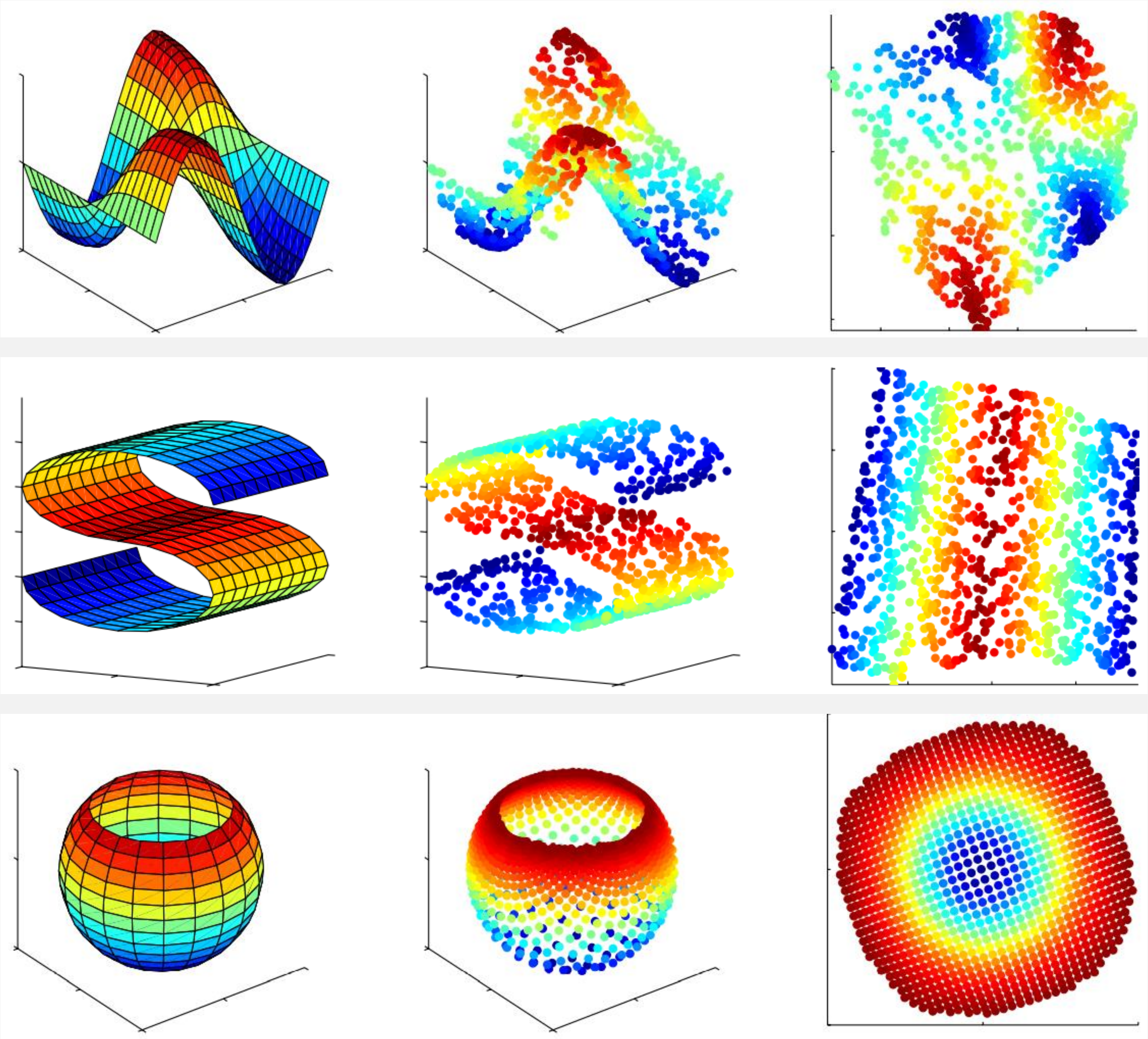
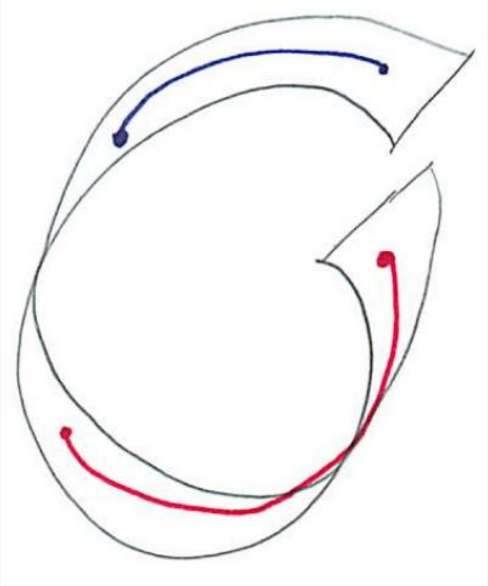
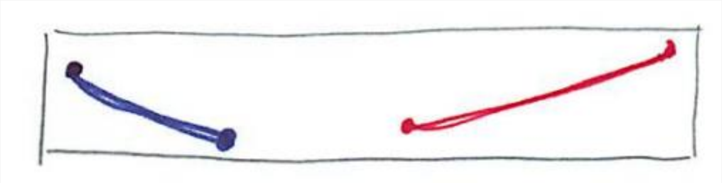
What does it mean to reduct the dimensionality?



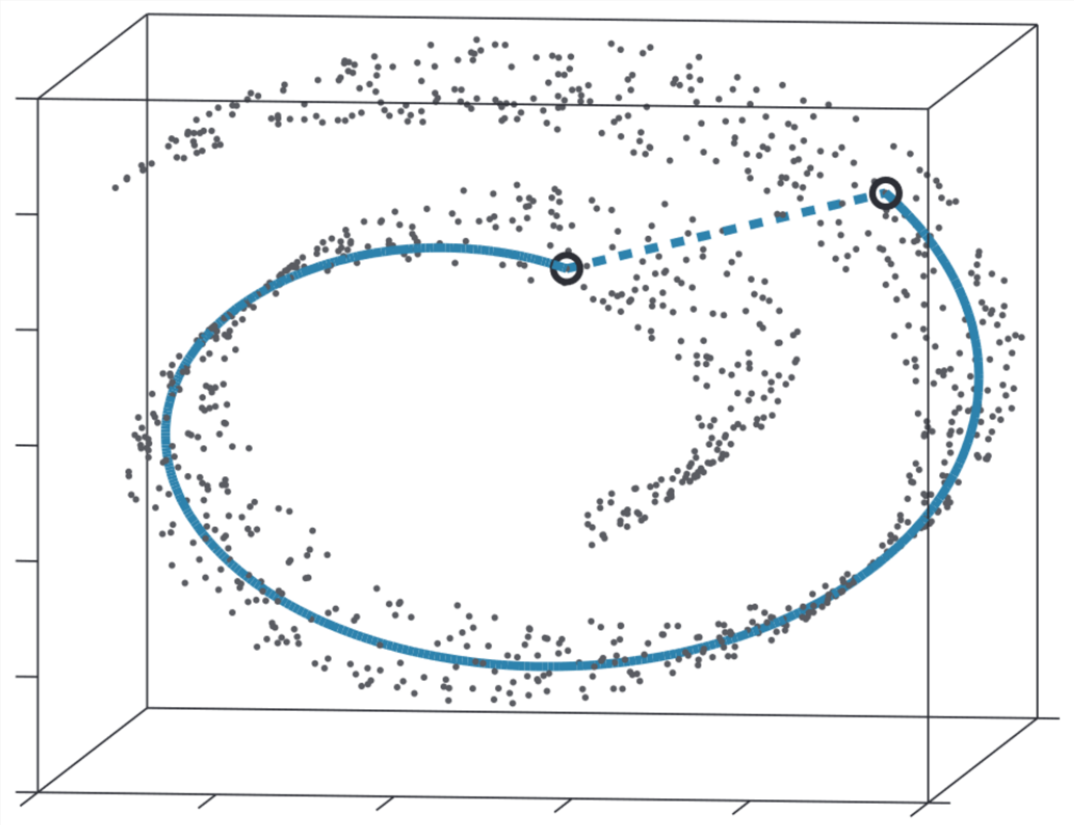


The intuition





ISOMAP Algorithm



Input:

$\boldsymbol{X}$ (high dimension) and a **distance function** $d(x_i, x_j)$, we choose the Euclidean distance

1) Construct weighted graph

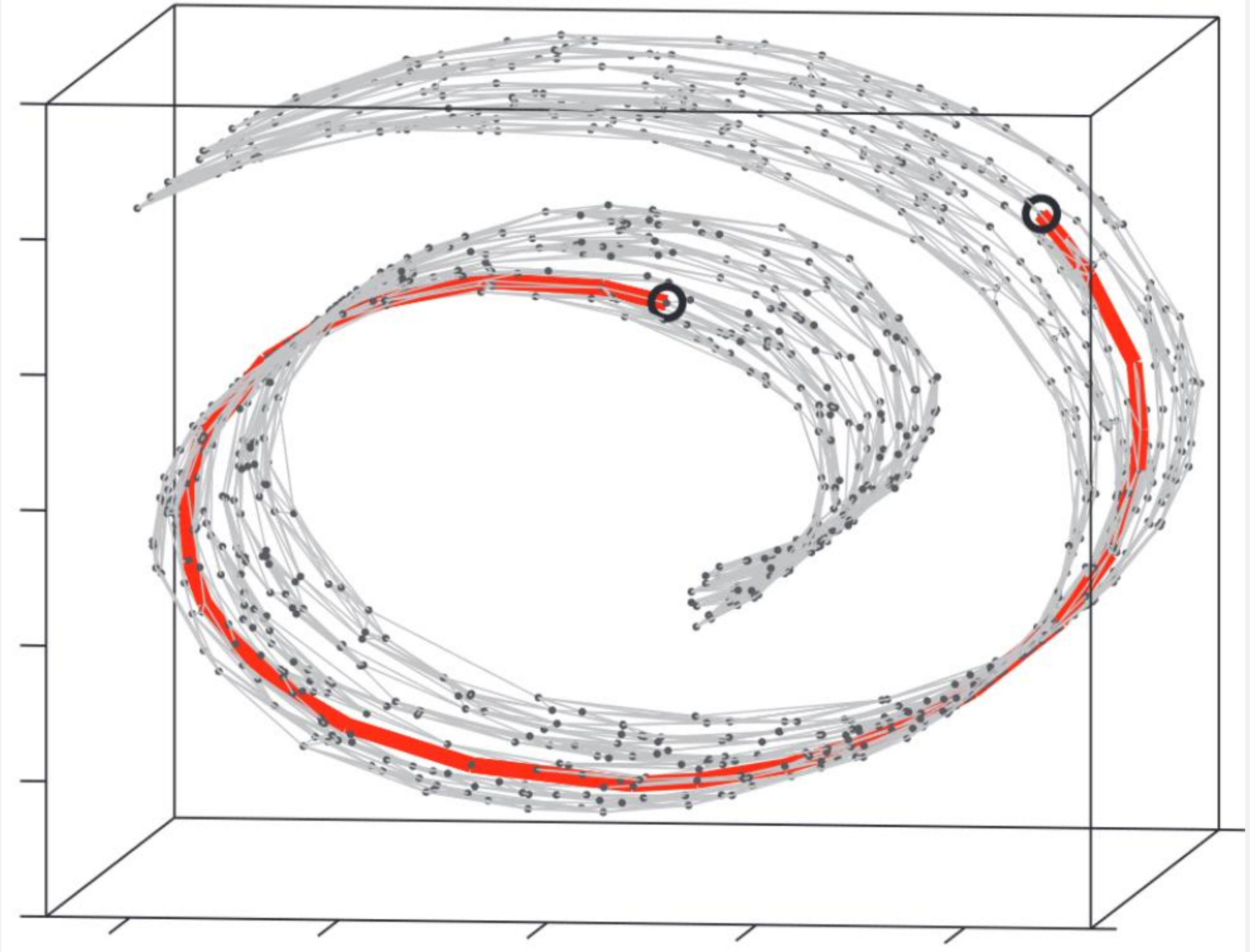
Find nearest neighbors $\mathcal{N}_i$ of each point x_i

K-nearest neighbors $\Rightarrow$ $|\mathcal{N}_i| = k$

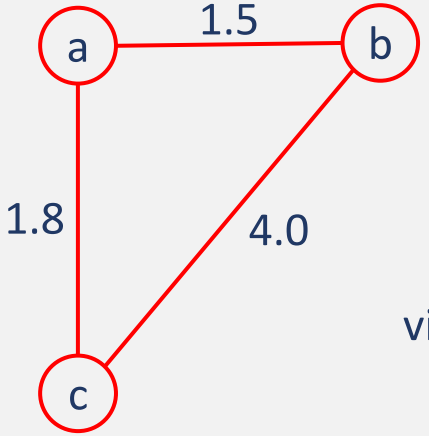
Fixed radius $\Rightarrow \mathcal{N}_i = \{ j \mid \|x_j - x_i\| \leq r \}$

Join nearest neighbors via edges weighted by the distances

$w_{ij} = \|x_j - x_i\|, \forall j \in \mathcal{N}_i$ (**local** distances)



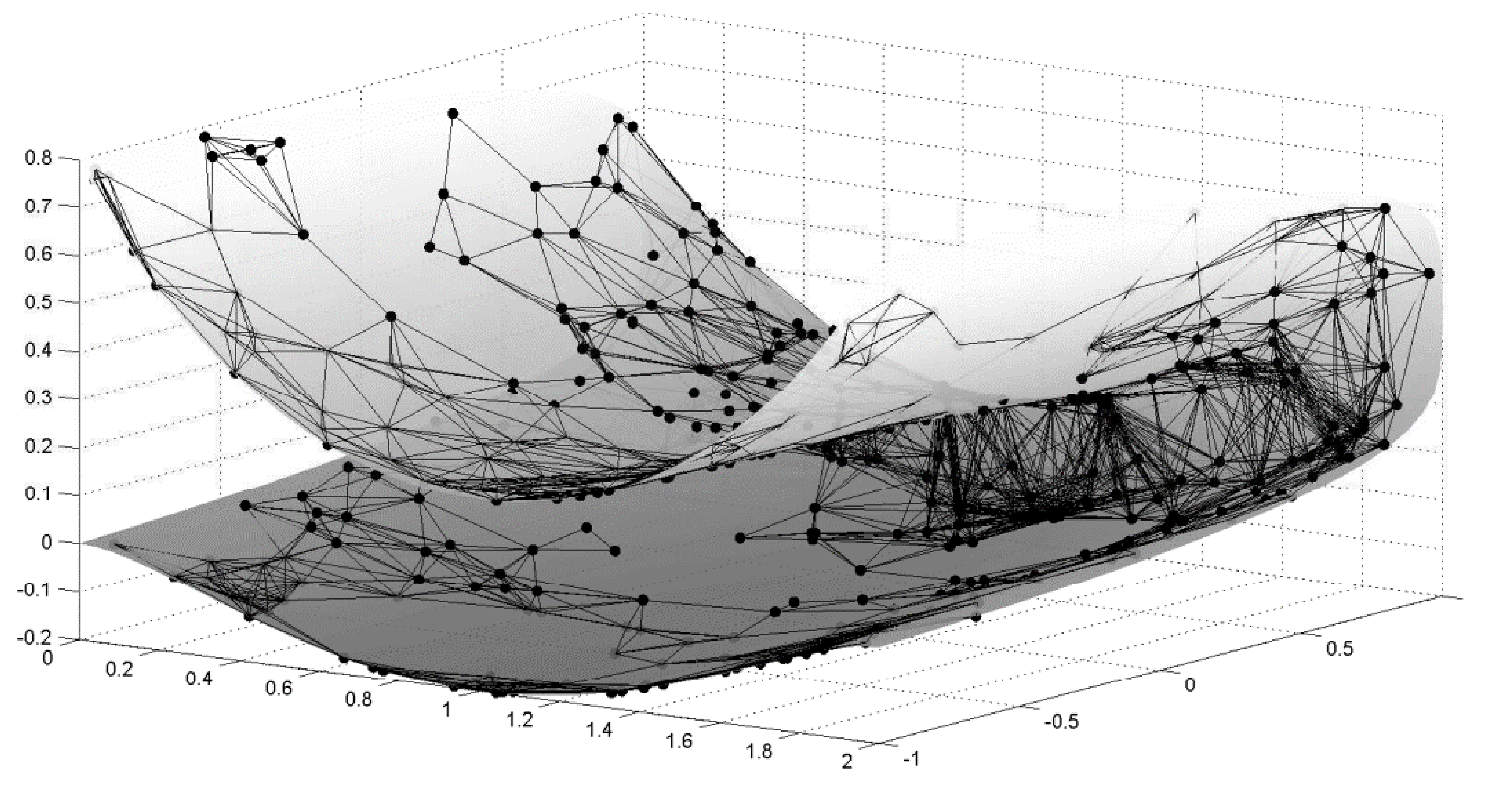
- **fully connected**
- **triangle inequality holds**



$w_{bc} > w_{ba} + w_{ac}$
$\Downarrow$
violates triangle inequality

2) Obtain (squared) distance matrix D
computing the shortest path distances d_{sp}
between all pairs of points
using Dijkstra's or Floyd's algorithm
(**geodesic** distances)

Why we should choose k carefully?



3) Apply metric MDS with D as input to get X (low dimension),
an embedding that preserves the geodesic distances

Embedding problem:

Given the distance matrix $\mathrm{D} \in \mathbb{R}^{n \times n}$, $d_{ij} = \left\| x_i - x_j \right\|$
Recover the points $(x_i)_{i=1 \dots n} \in \mathbb{R}^d$

This problem is called **(metric) multi-dimensional scaling**

Generally:

Given the objects $x_1, \dots, x_n \in X$
Find an embedding $\Phi : X \to \mathbb{R}^d$ such that $\left\| \Phi(x_i) - \Phi(x_j) \right\| = d_{ij}$

For general distance matrices D, we cannot achieve such an embedding without **distorting the data**

Classic MDS

Given a Euclidean distance matrix D we can express the entries s_{ij} of the **Gram matrix** $\boldsymbol{S} = \left(\langle x_i, x_j \rangle\right)_{ij=1\ldots n}$ in terms of entries of D :

$$d_{ij}{}^2 = \left\| x_i - x_j \right\|^2 = \langle x_i - x_j, x_i - x_j \rangle = \langle x_i, x_i \rangle + \langle x_j, x_j \rangle - 2\langle x_i, x_j \rangle$$

$$\boldsymbol{s_{ij}} = \langle x_i, x_j \rangle = \frac{1}{2}\left(\langle x_i, x_i \rangle + \langle x_j, x_j \rangle - d_{ij}{}^2\right) =$$
$$= \frac{1}{2}\left(d(0, x_i)^2 + d(0, x_j)^2 - d_{ij}{}^2\right) = \boldsymbol{\frac{1}{2}\left(d_{1i}{}^2 + d_{1j}{}^2 - d_{ij}{}^2\right)}$$

Because S it is positive definite, we can decompose S in the form $S = XX^t$ where $\mathrm{X} \in \mathbb{R}^{n \times d}$.
The rows of X are what we are looking for, **we set the embedding of point $\boldsymbol{x_i}$ as the $\boldsymbol{i}$-th row of $\boldsymbol{X}$**

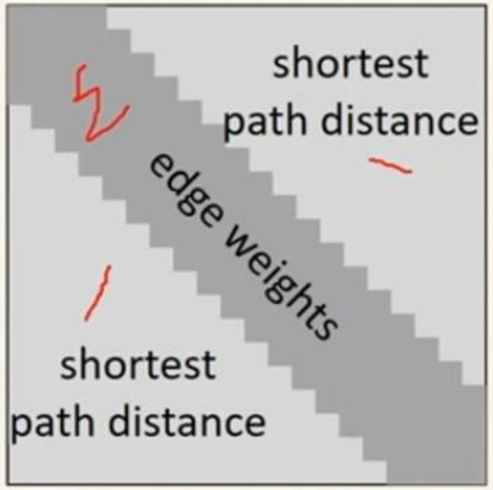
To find X we compute the eigenvalue decomposition $S = V\Lambda V^t$ and we define $\mathrm{X} = V\sqrt{\Lambda}$.
Tipically, we want to fix some dimension $d \leq n$, so we set V_d to be the first d columns of V and Λ_d the $d \times d$ diagonal matrix with the first d eigenvalues on the diagonal, then we set $\mathrm{X} = V_d\sqrt{\Lambda_d}$

Metric MDS

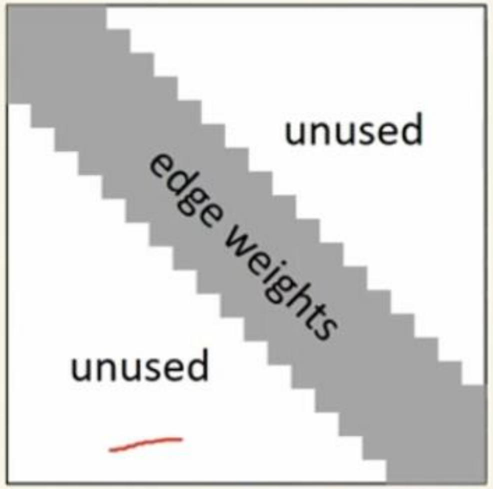
If the distance matrix D is not Euclidean we will not be able to recover an exact embedding.
Instead, we define a **stress function**, for example:

$$stress(embedding) = \frac{\sum_{ij}(\|x_i - x_j\| - d_{ij})^2}{\sum_{ij}\|x_i - x_j\|}$$

Then we try to find an embedding $x_1, \dots, x_n$ with small stress by a standard non-convex optimization algorithm, like gradient descent.



ISOMAP (global)



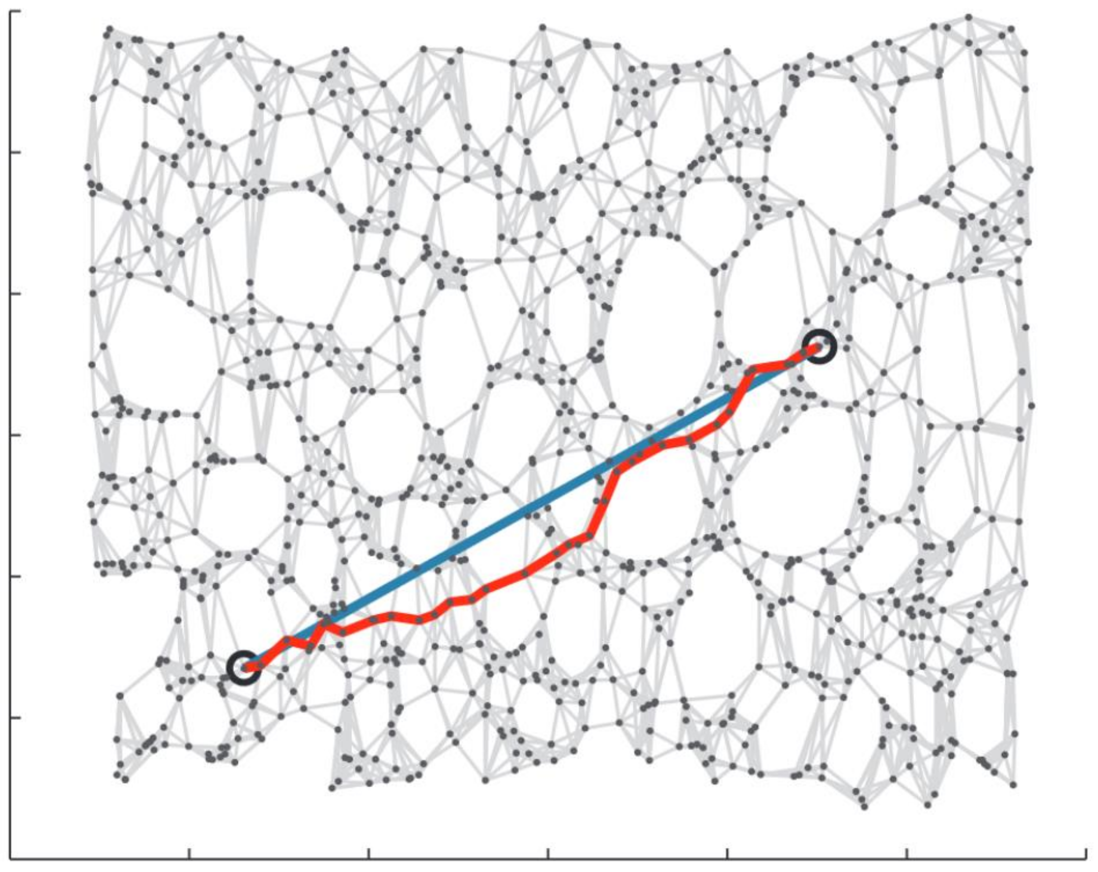
LLE (local)

ISOMAP

Proven asymptotic convergence

Distances between all pairs of points are required ⇒ $O(N^3)$ complexity
- Global method requires more computation
- Improvement: Landmark-ISOMAP (uses a subset of points (“landmarks”) to estimate geodesic distances

Not effective in the presence of non-convexities i.e. “holes” in manifold



ISOMAP on the MNIST digits dataset