Isometric Feature Mapping (ISOMAP)

Ke Chen

Department of Computer Science, The University of Manchester

Ke.Chen@manchester.ac.uk

Outline

Background

History, motivation and application

Principle

Problem setting, key ideas and main steps in ISOMAP

ISOMAP Algorithm

Algorithmic description for ISOMAP

Illustrative Example

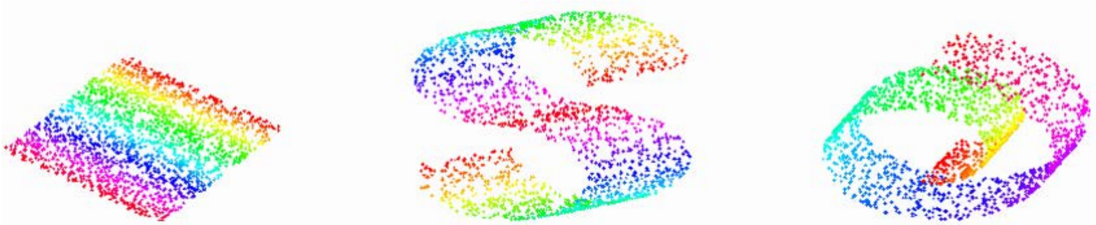
Synthetic, MNIST digit and visual perception datasets

Relevant Issue

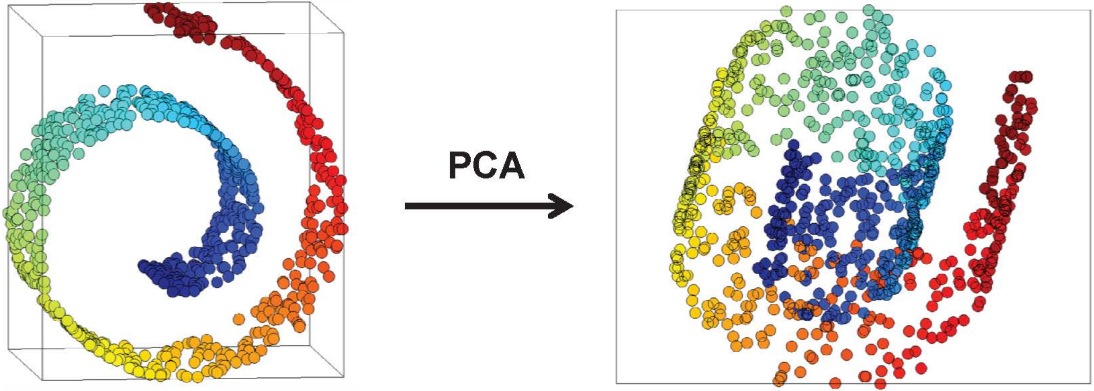
Limitation and extension

Background

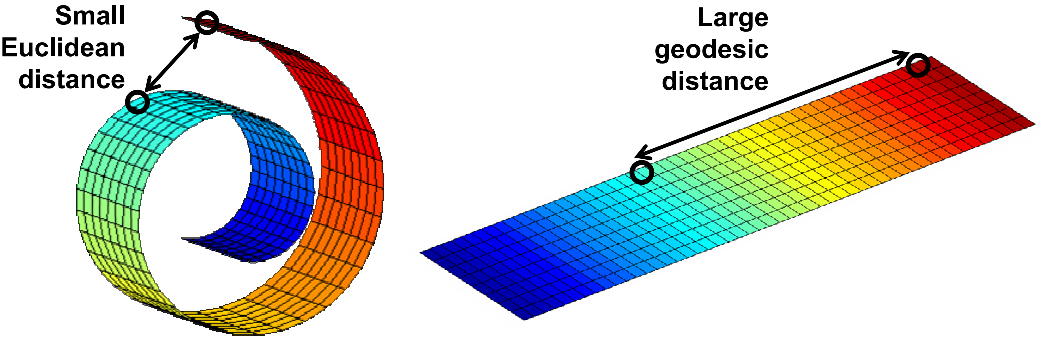
- ISOMAP algorithm proposed by Tenenbaum et al. (2000) published in *Science*, a seminal work in modern manifold learning research
- Motivation: tackle a fundamental problem in manifold learning; i.e a given high-dimensional data sampled from an unknown low-dimensional manifold, how can we automatically recover a good embedding?
- To model intrinsic nonlinear manifold for low-dimensional representation and visualisation



- Linear models, e.g. PCA, can only recover a linear manifold but do not work for nonlinear manifold. Why?

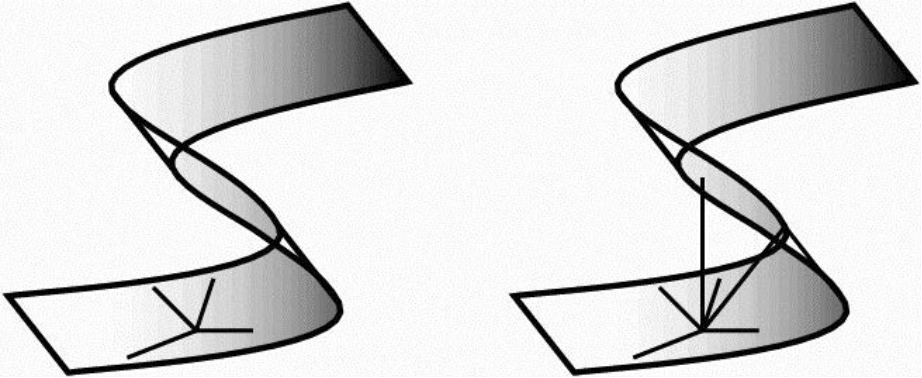


- Linear models, e.g. PCA, can only recover a linear manifold but do not work for nonlinear manifold. Why?
- Reason: Euclidean distance metric does not respect the geometry of nonlinear manifold!



Background

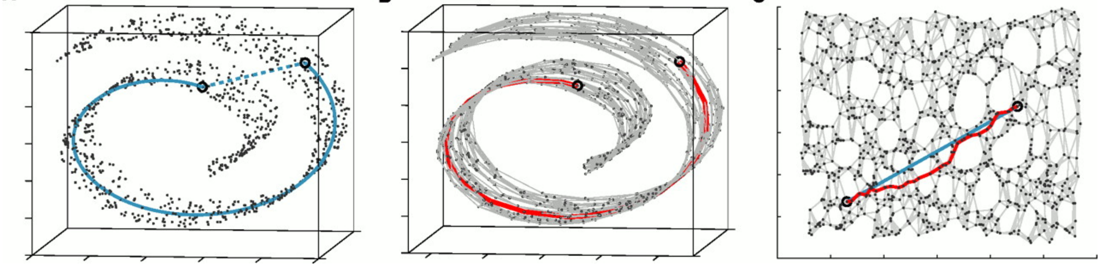
- Solution: find out proper distance metric that fits the geometry of nonlinear manifold.
- However, extremely difficult to find out a proper one for arbitrary nonlinear manifold.
- Fortunately, the homeomorphic property of manifold allows for an approximation of proper geodesic distance via the use of Euclidean distance locally.



Principle

- Problem: how to find out a transformation that preserves the geodesic distances between high-dimensional data points in a low-dimensional Euclidean space?
- Key idea: ISOMAP tackles this problem by
  - First, find out an approximation to geodesic distances between any points in high-dimensional space to establish a geodesic distance matrix
  - Then, apply cMDS to the geodesic distance matrix to achieve embedded coordinates in low-dimensional space where Euclidean distances between points are close to their corresponding geodesic distance.

A B C

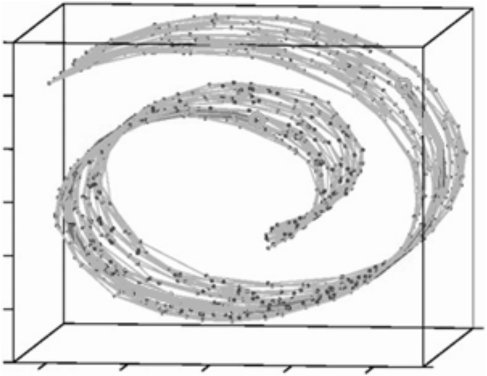


Principle

- Approximation to geodesic distance
  - For neighbouring data points, use Euclidean distances to approximate geodesic distances.
  - For distant data points, approximate geodesic distances with a sequence of steps on relevant groups of neighbouring points.
- To carry out this idea, a data set is represented as weighted graph where data points are treated vertices (nodes) and weights on edges are approximated distances between any connected data points.

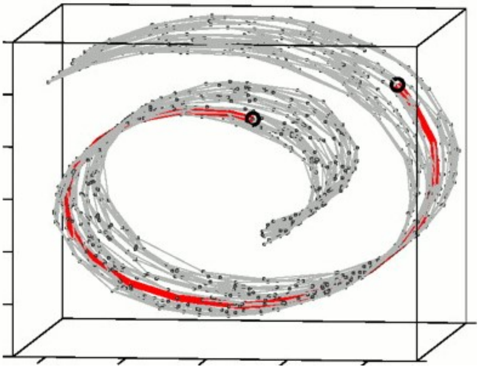
Approximation to geodesic distance for neighbouring data points

- Determine neighbours based on Euclidean distances between points in dataset.
  - ϵ-neighbourhood: connect each point to all points within a fixed radius ϵ.
  - K-NN: connect each point to all of its K nearest neighbours
- With the neighbourhood information, construct a weighted graph where there exist only edges with weights between a point and its neighbouring points.



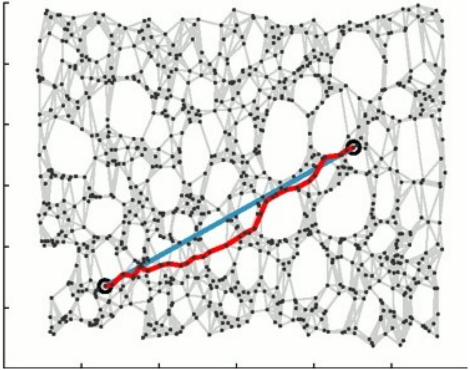
Approximation to geodesic distance for distant data points

- Approximate the geodesic distances between all pairs of non-connected points without edges by estimating their shortest-path distance in the weighted graph.
- Finding out shortest-path distance in the weighted graph is a classical optimisation problem in graph theory. There are many algorithms, e.g., Dijkstra algorithm.



Apply cMDS for low-dimensional embedding

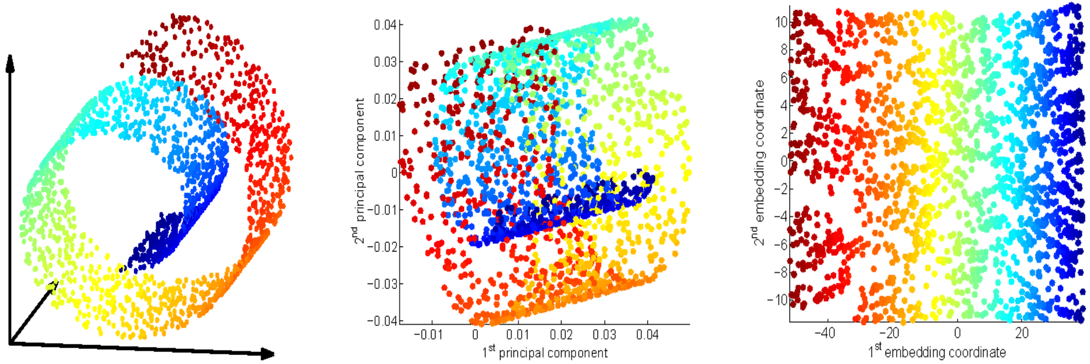
- Based on approximation to geodesic distances between data points in dataset, form a geodesic distance matrix in high-dimensional source space.
- Choose a proper dimension of low-dimensional target space, apply cMDS to the geodesic distance matrix for embedded coordinates of data points in low-dimensional Euclidean space (preserving a substantial amount of geodesic distances).



1. **Construct neighbourhood graph**
   - Based on Euclidean distance matrix in source space, $\Delta_X = (\delta_X(i,j))_{N \times N}$, set graph, $\mathcal{G}$, by connecting points i and j if $\delta_X(i,j) \leq \epsilon$ (ϵ-ISOMAP) or point i is one of the K-NN of point j (K-ISOMAP). Set edge lengths equal to $\delta_X(i,j)$.
2. **Compute shortest paths**
   - Initialise $\delta_{\mathcal{G}}(i,j) = \delta_X(i,j)$ if i and j linked by an edge; $\delta_{\mathcal{G}}(i,j) = \infty$ otherwise.
   - For $k = 1, \cdots, N$, replace all entries $\delta_{\mathcal{G}}(i,j)$ by $\min\{\delta_{\mathcal{G}}(i,j), \delta_{\mathcal{G}}(i,k) + \delta_{\mathcal{G}}(k,j)\}$.
   - Form the geodesic data matrix in source space: $\Delta_{\mathcal{G}} = (\delta_{\mathcal{G}}(i,j))_{N \times N}$.
3. **Construct p-dimensional embedded coordinates with cMDS**
   - Convert the geodesic data matrix, $\Delta_{\mathcal{G}}$, into its corresponding Gram matrix, G.
   - Conduct spectral decomposition: $G_{N \times N} = V_{N \times N} \Sigma_{N \times N} V_{N \times N}^T = \sum_{i=1}^{N} \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i^T$.
   - Produce p-dimensional embedded coordinates: $Z^* = V_p^T \Sigma_p^{\frac{1}{2}}$, where Σ_p is a diagonal matrix of top p eigenvalues and V_p is the matrix of the corresponding eigenvectors. In the vector-wise or element-wise notation: $\boldsymbol{z}_i^* = \sqrt{\lambda_i} \boldsymbol{v}_i$ or $z_{ij}^* = \sqrt{\lambda_i} v_{ij}$, $i = 1, \ldots, p$; $j = 1, \ldots, N$. (PoV can be used to decide proper p)

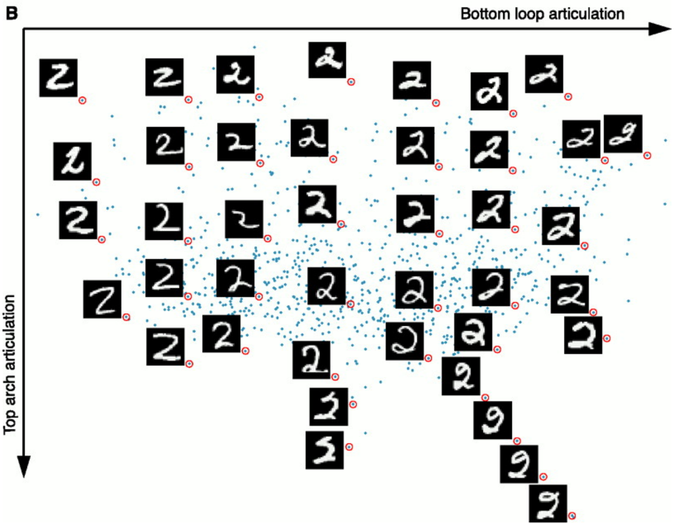
Synthetic dataset: Swissroll

- K-ISOMAP versus PCA: $N = 2,000$, $K = 13$, $d = 3$, $p = 2$



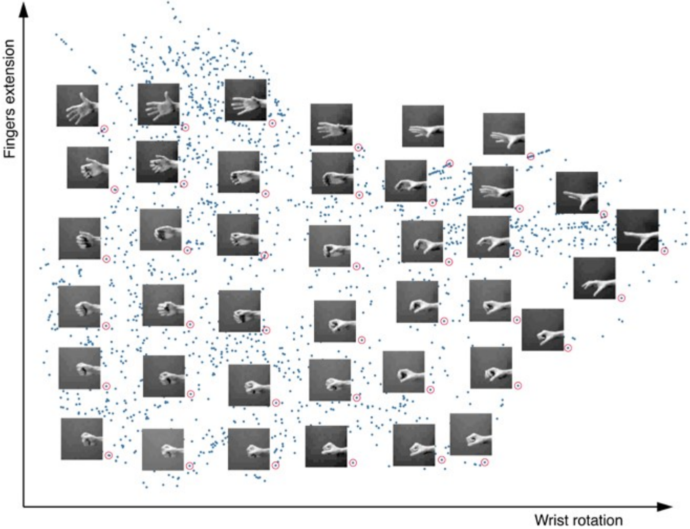
MNIST digit

- ϵ-ISOMAP: $N = 1,000$, $\epsilon = 4.2$, $d = 28 \times 28$, $p = 2$



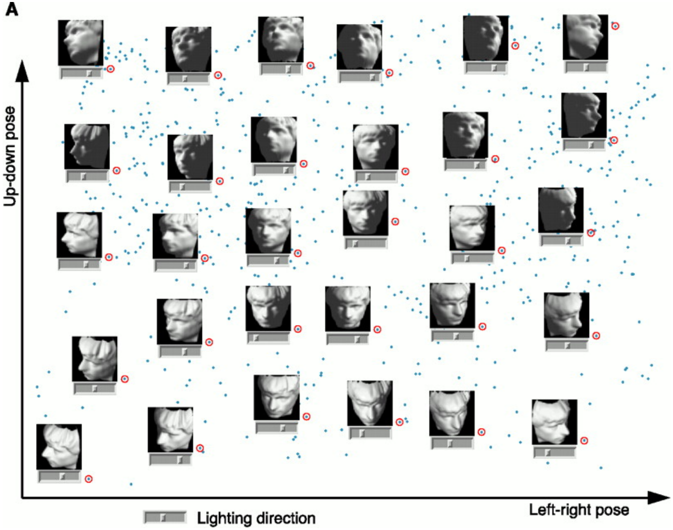
Visual perception: hand images

- K-ISOMAP: $N = 2,000$, $K = 6$, $d = 64 \times 64$, $p = 2$



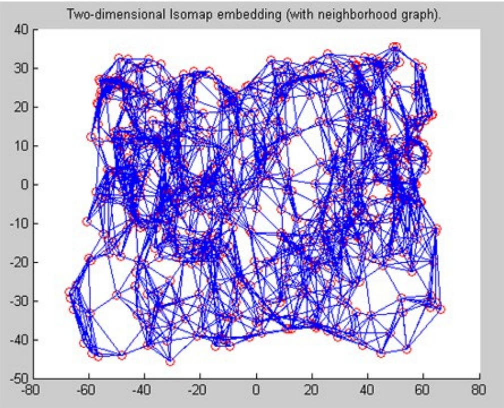
Visual perception: facial images

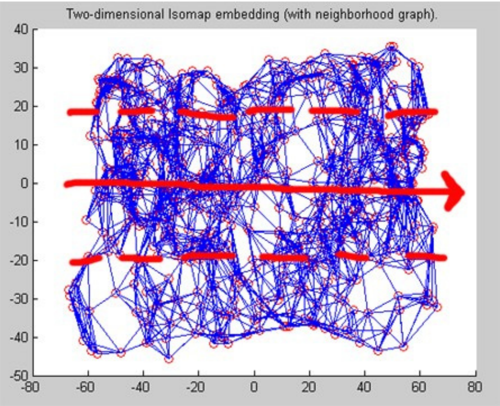
- K-ISOMAP: $N = 698$, $K = 6$, $d = 64 \times 64$, $p = 2$ (3)



Visual perception: facial images

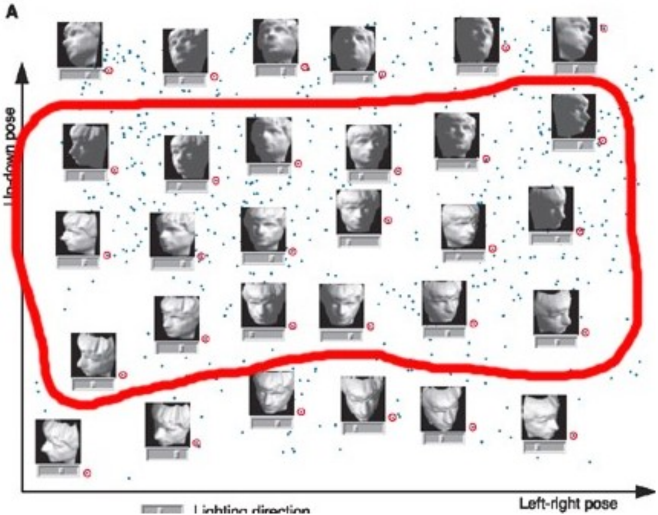
- **Traversing the manifold**
  - draw a smooth line through the manifold
  - only add images within a certain manifold distance from this line.





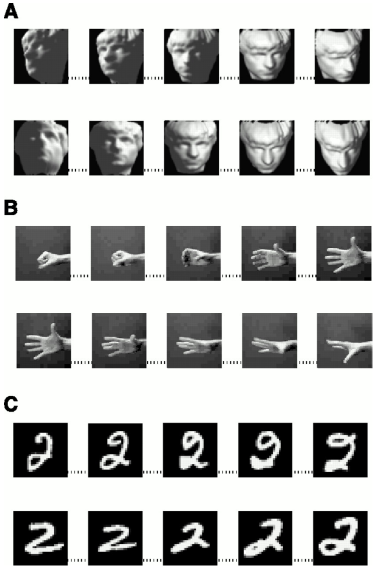
Visual perception: facial images

- **Traversing the manifold**

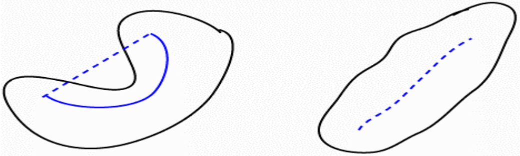


Interpolations in different spaces

- 3 image datasets
  A) Faces, B) Wrists, C) Digits
- Linear interpolations in low dimensional ISOMAP feature space
- Nonlinear interpolations in high dimensional pixel space



- Limitation
  - To work, sufficient data points have to be sampled from the manifold smoothly.
  - Work only on convex Euclidean manifolds to recover the intrinsic geometry, e.g. for 2-D manifolds coming from any physical transformations such as folding, twisting and curving a sheet of paper without tears, holes, or self-intersections.



- Out-of-sample extension
  - ISOMAP (MDS) does not provide any mapping function: $\boldsymbol{z} = f(\boldsymbol{x})$.
  - For extension to unseen data, however, we can use the known raw data and their embedded coordinates, $(X, Z) = \{(\boldsymbol{x}_n, \boldsymbol{z}_n)\}_{n=1}^{N}$, as training examples to learn a parametric mapping function: $\boldsymbol{z} = f(\Theta, \boldsymbol{x})$, where Θ refers to all the parameters in a learning model, e.g., support vector regressor or deep neural networks.

Extension

- **Conformal ISOMAP**: relax the convex manifold assumption by preserving manifold orientation instead of geodesic distance.
- **C ISOMAP**: allow for magnifying the regions of high density but shrinking the regions of low density of data points in manifold.
- **Incremental ISOMAP**: allow for online ISOMAP learning by embedded points one by one instead of training in a batch manner.
- **Landmark ISOMAP**: overcome high computational burden in learning by using landmarks, only a subset of representative data.
- **Robust ISOMAP**: replace Dijkstra path-based geodesic distance estimates with parallel transport unfolding approximation for robustness to noise, a fundamental weakness of ISOMAP.

Reference

If you want to deepen your understanding and learn something beyond this lecture, you can self-study the optional references below.

[Alpaydin, 2014] Alpaydin E. (2014): *Introduction to Machine Learning* (3rd Ed.), MIT Press. (Section 6.10)

[Tenenbaum et al., 2000] Tenenbaum J.B., de Silva V. and Langford J.C. (2000): A global geometirc framework for nonlinear dimensionality reduction. *Science*, Vol 290, pp. 2319-2323.