Visualizing Data using t-SNE

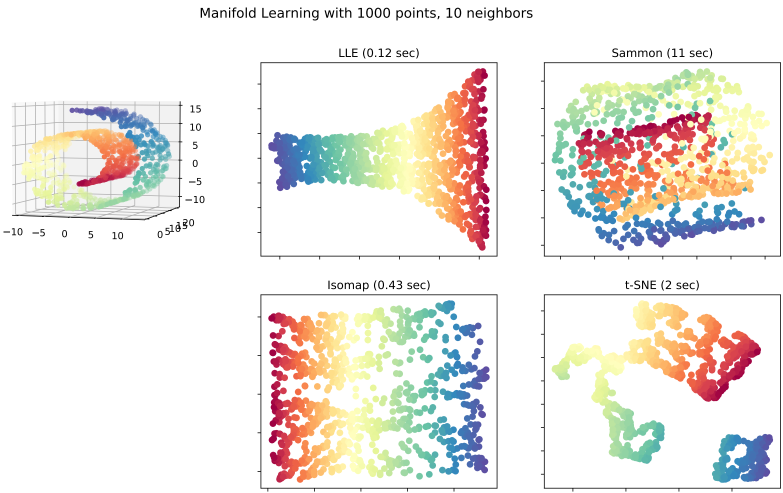
Section 3 Comparison of Dimensionality Reduction Methods

Taran Lynn, Xiaoli Yang, Xiaoxing Chen

October 27, 2020

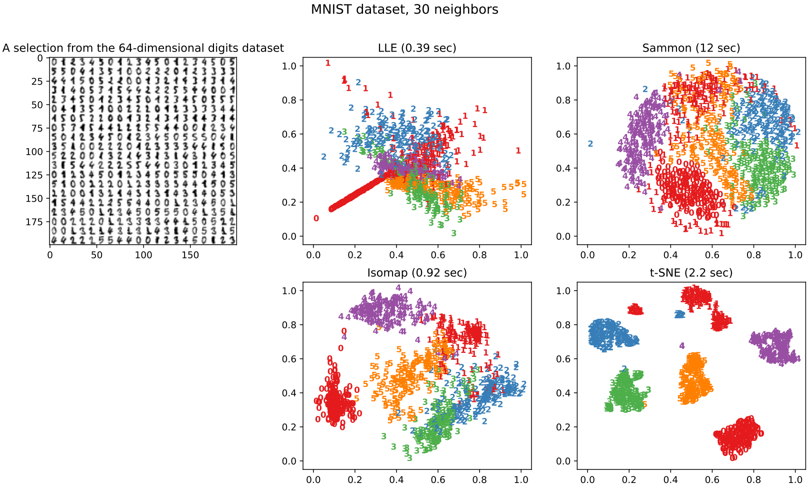
Algorithm Comparison

Criteria: similarity preservation, overlapping, distortion, time



Algorithm Comparison

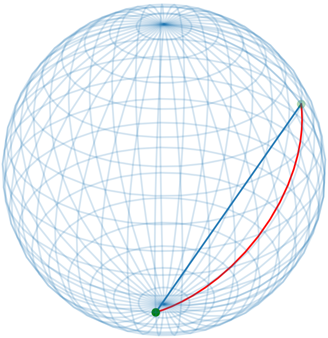
Criteria: similarity preservation, overlapping, distortion, time



Isometric Mapping (Isomap)

a non-linear dimensionality reduction method which tries to preserve the geodesic distances in the lower dimension

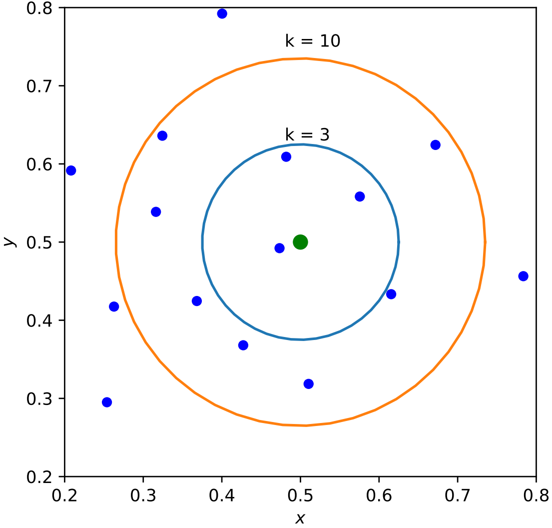
In geometry, a **geodesic** is commonly a curve representing in some sense the shortest path between two points in a surface, or more generally in a Riemannian manifold.



Isometric Mapping (Isomap)

a non-linear dimensionality reduction method which tries to preserve the geodesic distances in the lower dimension

Nearest neighbor search



Isometric Mapping (Isomap)

a non-linear dimensionality reduction method which tries to preserve the geodesic distances in the lower dimension

1. Nearest neighbor search

   Isomap starts by creating a neighborhood network.

2. Shortest-path graph search

   Isomap uses graph distance to the approximate geodesic distance between all pairs of points.

3. Partial eigenvalue decomposition

   And then, through eigenvalue decomposition of the geodesic distance matrix, it finds the low dimensional embedding of the dataset.

Isometric Mapping (Isomap)

Complexity

$$\underbrace{O[D\log(k)N\log(N)]}_{\text{nearest neighbors search}} + \underbrace{O[N^2(k+\log(N))]}_{\text{shortest-path graph search}} + \underbrace{O[dN^2]}_{\text{partial eigenvalue decomposition}}$$

- N: number of training data points
- D: input dimension
- k: number of nearest neighbors
- d: output dimension

Locally Linear Embedding (LLE)

A topology preserving manifold learning method

Assumptions:

- Data is well sampled i.e. density of the dataset is high.
- Dataset lies on a smooth manifold.

1. Nearest neighbor search

A distance metric is needed to measure the distance between the two points and classify them as neighbors. For example Euclidean, Mahalanobis, hamming and cosine. Either e-neighborhood or K-nearest neighbors will be used to create a neighborhood matrix.

2. Weight Matrix Construction

Each point of the dataset is reconstructed as a linear weighted sum of its neighbors.

3. Partial Eigenvalue Decomposition

Create each point in lower dimension using its neighbors and local W matrix. The neighborhood graph and the local Weight matrix capture the topology of the manifold.

Locally Linear Embedding (LLE)

A topology preserving manifold learning method

Complexity

$$\underbrace{O[D\log(k)N\log(N)]}_{\text{nearest neighbors search}} + \underbrace{O[DNk^3]}_{\text{weight matrix construction}} + \underbrace{O[dN^2]}_{\text{partial eigenvalue decomposition}}$$

- N: number of training data points
- D: input dimension
- k: number of nearest neighbors
- d: output dimension

Weakness: Sensitive to outliers and noise

Datasets have a varying density and it is not always possible to have a smooth manifold.

Sammon Mapping

Cost Function

$$E_s = \frac{1}{\sum_{ij} \|x_i - x_j\|} \sum_{i \neq j} \frac{(\|x_i - x_j\| - \|y_i - y_j\|)^2}{\|x_i - x_j\|}$$

Steps

- Distance calculation
- Distance matrix construction
- Minimizing the projection error

Main Weakness

the importance of retaining small pairwise distances in the map is largely dependent on small differences in these pairwise distances. In particular, a small error in the model of two high-dimensional points that are extremely close together results in a large contribution to the cost function.

t-SNE

Keep the pairwise similarity in lower dimension

Steps

- Nearest neighbor search
- Pairwise similarity calculation
- Minimize dissimilarity cost function

Complexity

- Computational complexity: $O(N^2)$
- Memory complexity: $O(N^2)$

Weakness

- Dimensionality reduction for other purposes (reduce to dimension $d > 3$)
- Curse of intrinsic dimensionality
- Non-convexity of the t-SNE cost function

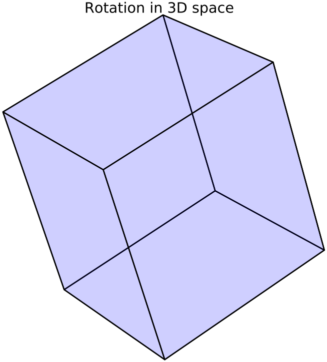
Test with 3D Rotation Group

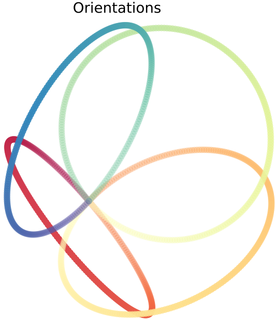
Rotation Axis:

$\alpha_1 = (1,1,1), \alpha_2 = (-1,1,1), \alpha_3 = (1,-1,1), \alpha_4 = (1,1,-1)$

Circles intersect at:

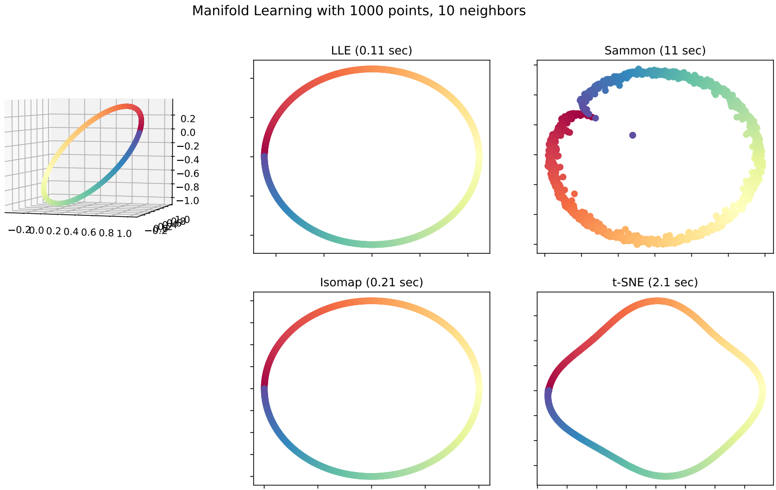
$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$





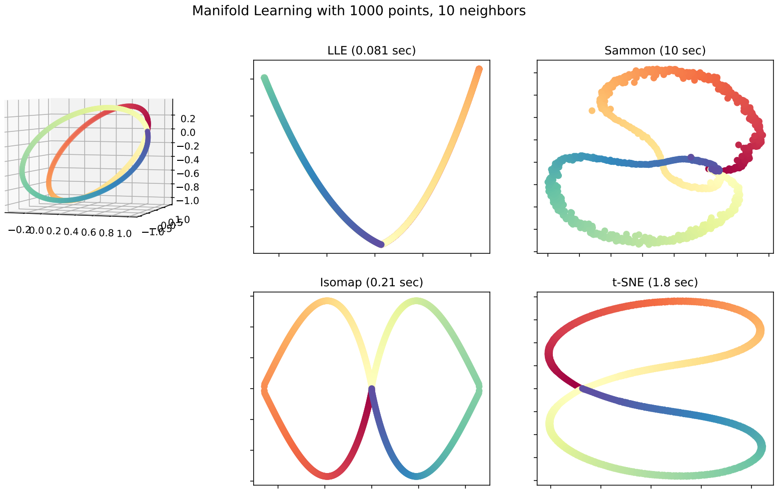
Algorithm Comparison

Criteria: similarity preservation, overlapping, distortion, time



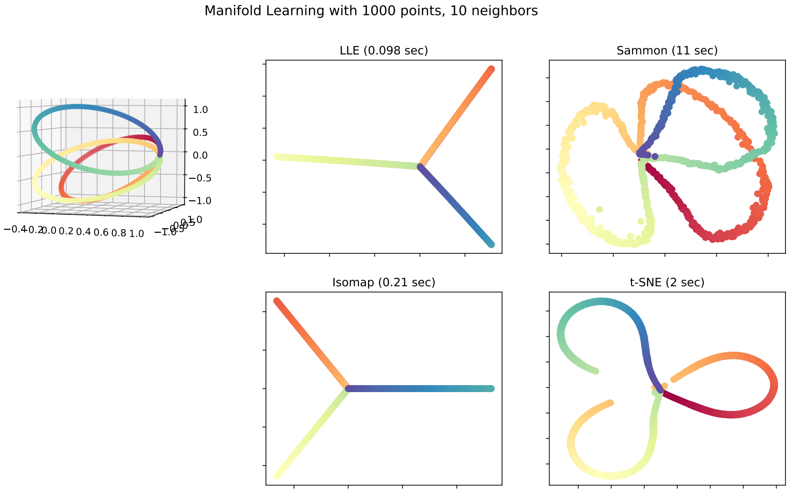
Algorithm Comparison

Criteria: similarity preservation, overlapping, distortion, time



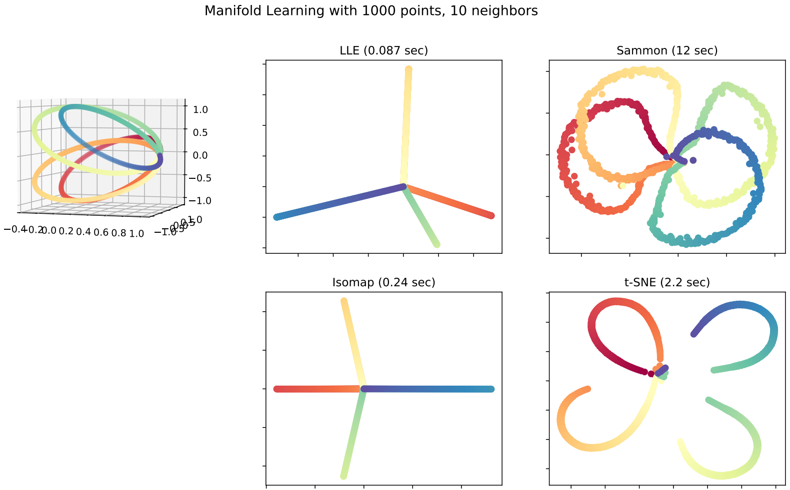
Algorithm Comparison

Criteria: similarity preservation, overlapping, distortion, time



Algorithm Comparison

Criteria: similarity preservation, overlapping, distortion, time



Comparison and Conclusion

1. Nearest neighbor search

All methods except Sammon mapping do NN Search. Sammon mapping scales the weight of closer neighbors by dividing a factor.

2. Local structure preservation

- Isomap: preserve geodesic distance (represented by graph distance)
- LLE: preserve manifold topology
- Sammon mapping: preserve weighted distances
- t-SNE: minimize pairwise dissimilarity(represented by conditional probabilities)

3. Solving methods

Isomap and LLE use eigenvalue decomposition, Sammon mapping and t-SNE use gradient descent.