Isomap

1. Introduction:

- **Isomap**, or **Isometric Mapping**, is a **non-linear dimensionality reduction technique** and a **manifold learning algorithm**.
- Its core purpose is to **preserve** the **intrinsic geometric structure** and **global data structure** of **high-dimensional data** by focusing on the **geodesic distance** between **data points** when projecting them into a **lower-dimensional space**.
- This approach, based on **spectral theory**, enables Isomap to capture **non-linear relationships** crucial for datasets with **complex structures**, a key distinction from **linear techniques** like PCA.

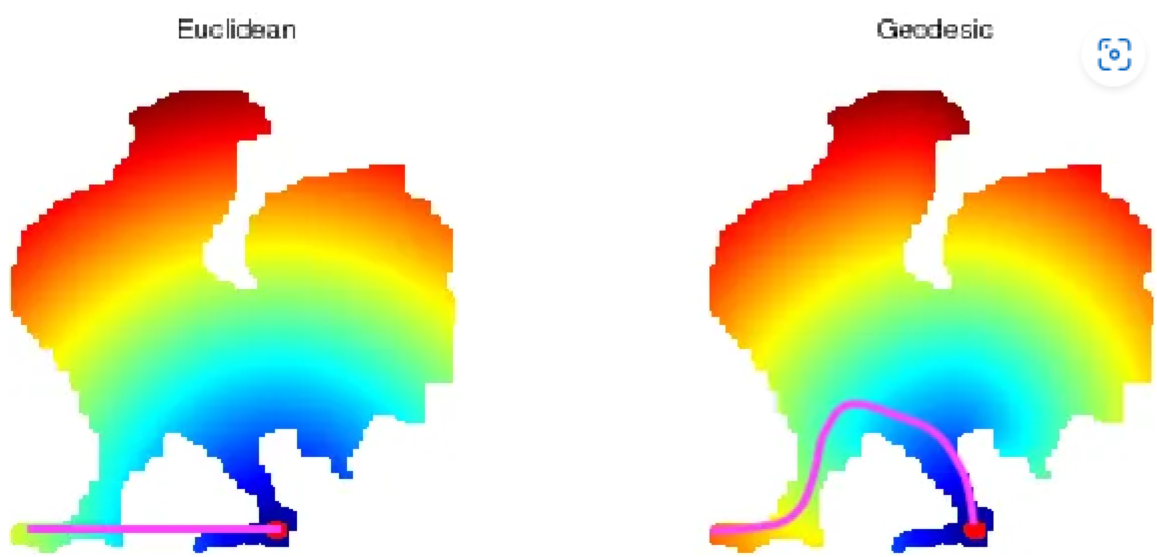
Need of Manifold Learning:

We often suspect that high-dim may actually lie on or near a low-dim manifold (often much lower!); It would be useful if we could reparametrize the data in terms of this manifold, yielding a low-dim embedding; BUT - we typically don't know the form of this manifold.

---

2. Jargons:

2.1 Geodesic Distance:



2.2 Double Centering a matrix:

It means to transform a matrix such that

- mean for any row = 0
- mean for any column = 0

2.2.1 This is computed as follows:

Step 1: For a Matrix A, prepare Matrix B and C such that
Step 2: Matrix B

Mean(col_1)	**Mean(col_2)**	**Mean(col_3)**
Mean(col_1)	**Mean(col_2)**	**Mean(col_3)**
Mean(col_1)	**Mean(col_2)**	**Mean(col_3)**

Step 3: Matrix C

Mean(row_1)	**Mean(row_2)**	**Mean(row_3)**
Mean(row_1)	**Mean(row_2)**	**Mean(row_3)**
Mean(row_1)	**Mean(row_2)**	**Mean(row_3)**

Step 4: Double centered Matrix A = A-B-C+Mean(A)

2.3 Dissimilarity Matrix:

It is a matrix that represents dissimilarity between points in a dataset. This dissimilarity can be calculated using any measure. Though the most common measure is the distance between points. The more the distance, the more dissimilar the samples are.

	A	B	C	D	E	F
A	0	16	47	72	77	79
B	16	0	37	57	65	66
C	47	37	0	40	30	35
D	72	57	40	0	31	23
E	77	65	30	31	0	10
F	79	66	35	23	10	0

2.4 Manifold:

Manifold is any space that is locally Euclidean. For example, the Earth is round but it looks flat to us. The Earth is a manifold: locally it is flat, but globally we know it is a sphere. Then, manifold learning performs dimensionality reduction by representing data as low-dimensional manifolds embedded in a higher-dimensional space.

---

3. Isometric Mapping:

Step 1: Calculating Geodesic Distance:

1. We can calculate the adjacency matrix for all the points in the dataset using kNN where K= [3,4].
2. Then using this weighted matrix, we apply dijkstra algorithm to find the shortest distance.
3. Hence we can calculate the geodesic distance between 2 points.

Step 2: Dissimilarity matrix:

Step 3: Square the Dissimilarity matrix and double center it.

Step 4: Eigendecomposition & choosing 'k' eigenvectors:

This is something similar to what we do in PCA after calculating the correlation matrix.

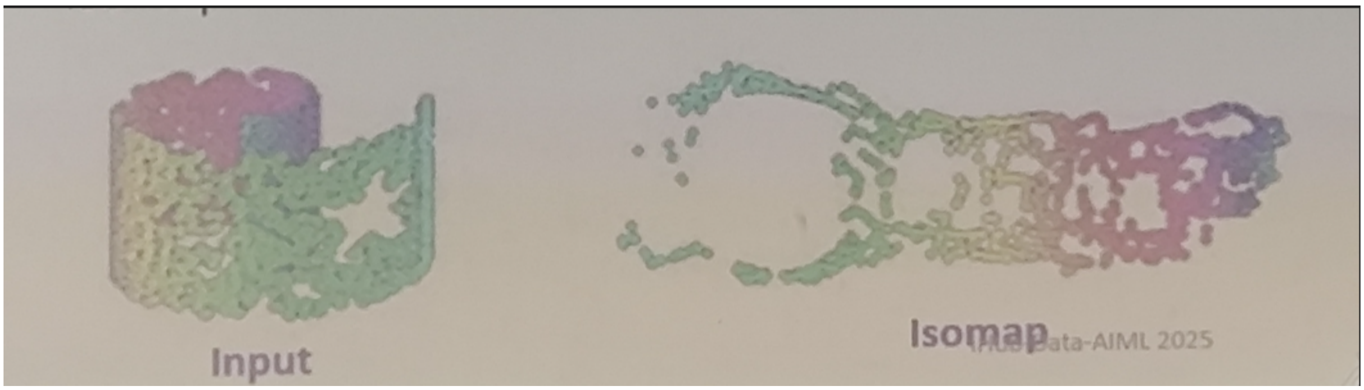
---

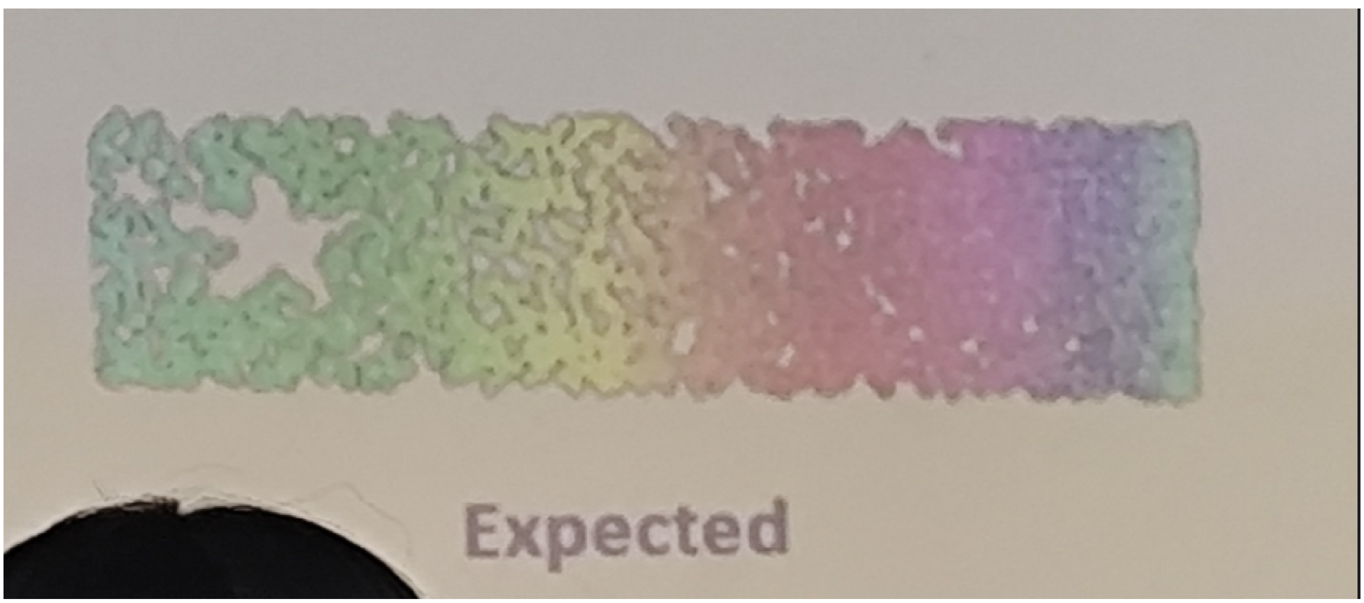
4. Drawbacks:

A few drawbacks always exist

1. In manifold learning, there is no good framework for handling missing data. In contrast, there are straightforward iterative approaches for missing data in PCA.
2. In manifold learning, the presence of noise in the data can "short-circuit" the manifold and drastically change the embedding. In contrast, PCA naturally filters noise from the most important components.
3. The manifold embedding result is generally highly dependent on the number of neighbors chosen, and there is generally no solid quantitative way to choose an optimal number of neighbors. In contrast, PCA does not involve such a choice.

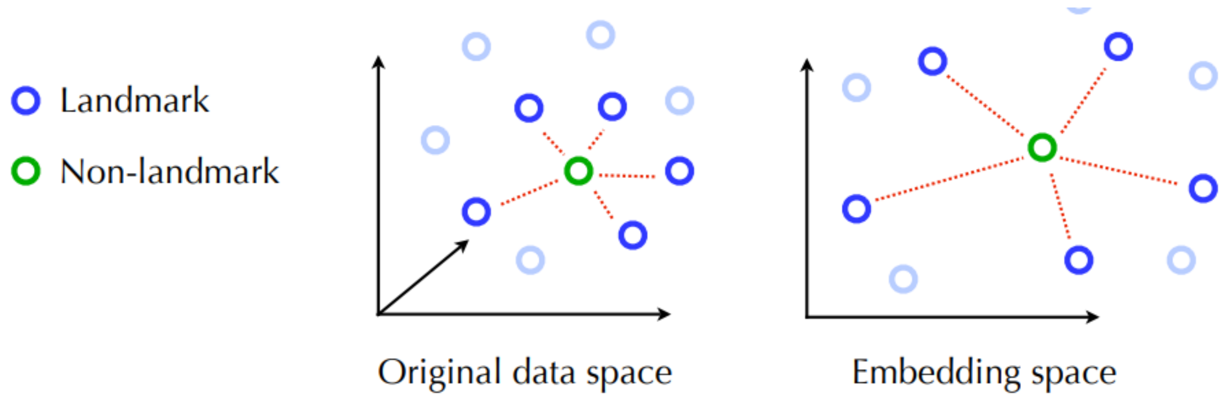
4. In manifold learning, the globally optimal number of output dimensions is difficult to determine. In contrast, PCA lets you find the output dimension based on the explained variance.
5. In manifold learning, the meaning of the embedded dimensions is not always clear. In PCA, the principal components have a very clear meaning.
6. In manifold learning the computational expense of manifold methods scales as $O[N^2]$ or $O[N^3]$. For PCA, there exist randomized approaches that are generally much faster (though see the megaman package for some more scalable implementations of manifold learning).
7. Isomap suffers from non-convexity such as holes on manifolds.





Solution to Isomap scaling:

For Large N, all-pairs for shortest path is computationally expensive. Hence, the solution is to compute an embedding of the subset of data(landmarks). Embed non-landmarks by convex triangulation.



---

5. References:

1. Dimension Reduction using Isomap. Something you need for nonlinear data | by Mehul Gupta | Data Science in Your Pocket | Medium
2. David_NDR_lecture.pdf
3.