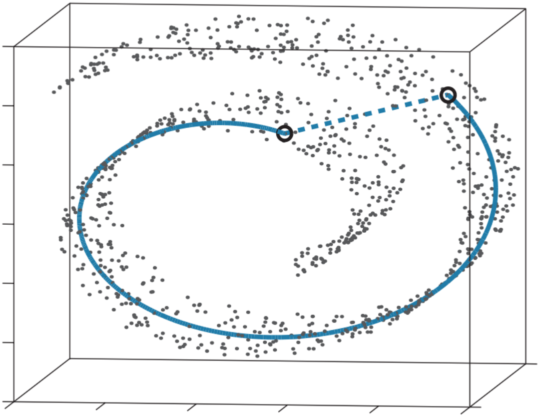
NON-LINEAR DIMENSION REDUCTION

ISOMAP

110024516 邱繼賢 109024513 趙唯毓
110024521 邱婉如 106020015 羅靖雅 107020010 張育豪

11, May, 2022

Geodesic distance

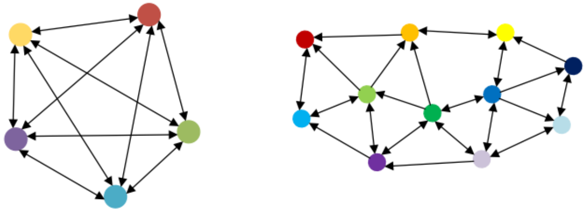


ISOMAP idea

- Approximate pairwise geodesic distances in a manifold.
- Apply MDS on the distance matrix to find a 2-dim embedding that preserves geodesic distances.

Construct neighborhood graph

Create graph G of $\{x_i\}$ by either using a K-NN rule or ϵ-rule where each edge $e = (x_i, x_j)$ is weighted by the Euclidean distance between the two points.



(a) local complete graph (b) global KNN graph

Compute Graph Distances

There are two ways to compute the shortest path of two nodes

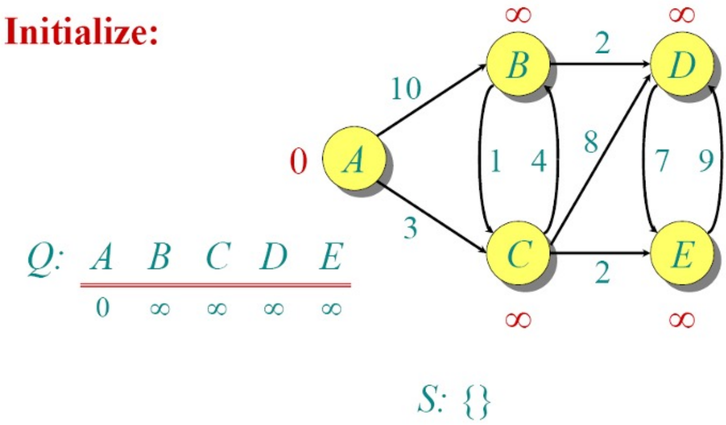
- Floyd's Algorithm: Can find the shortest path between all of the pairs
- Dijkstra's Algorithm: Better for sparse graph

Dijkstra Algorithm

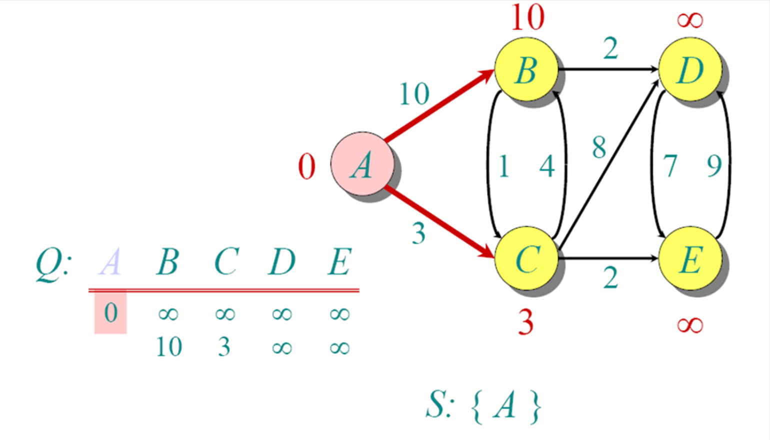
- Dijkstra Algorithm is a greedy algorithm for solving the single source shortest path problem
- Method

  1. It divided sets of vertex into two sets S and V\S where S contains the i nearest neighbors that have been find in first ith steps
  2. $L(\omega) = min\{L(\omega), L(u) + c(u, \omega))\} where\, u \in S, \omega \in V \backslash S$

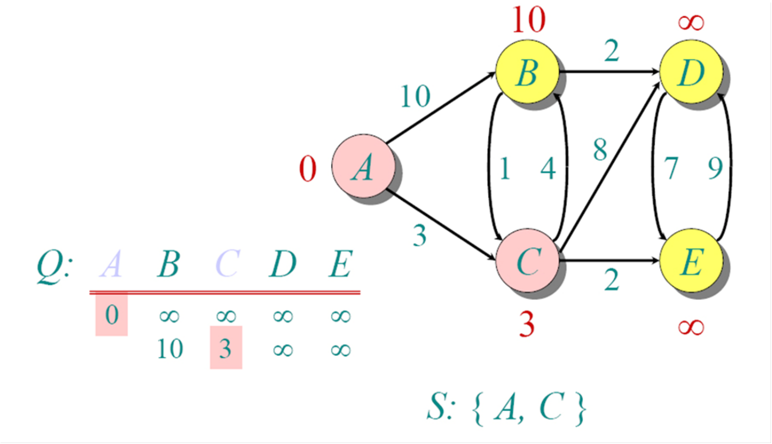
Dijkstra Algorithm Example



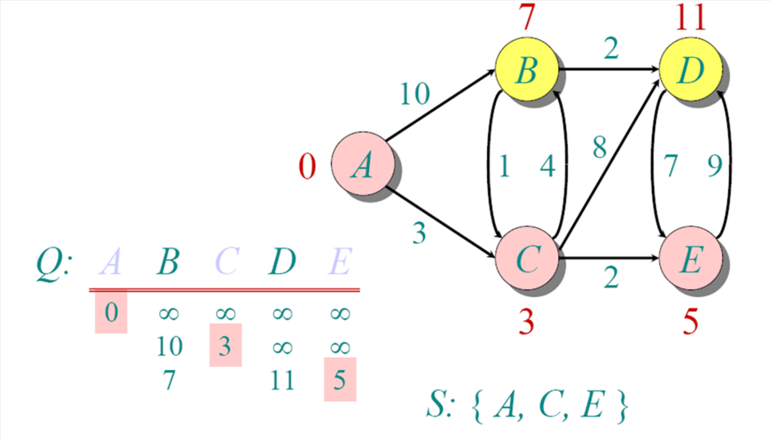
Dijkstra Algorithm Example



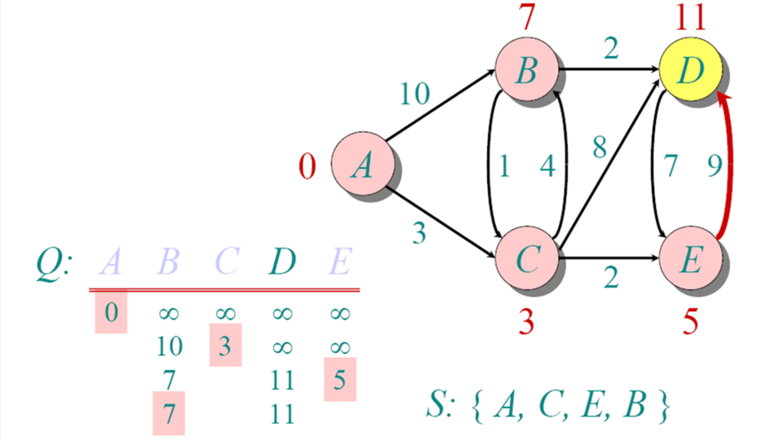
Dijkstra Algorithm Example



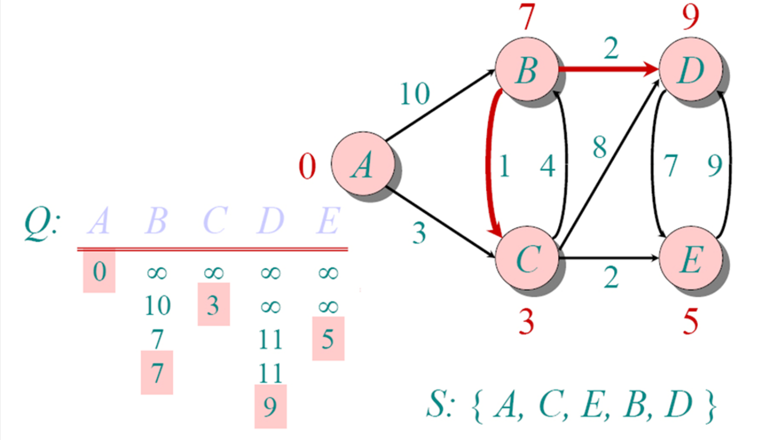
Dijkstra Algorithm Example



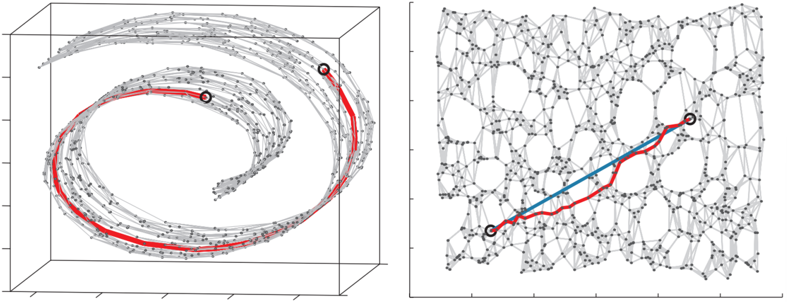
Dijkstra Algorithm Example



Dijkstra Algorithm Example

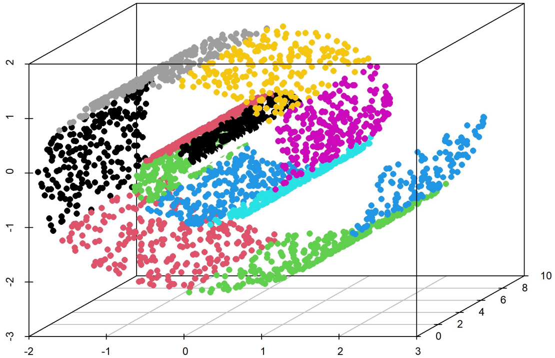


MDS with Graph Distance



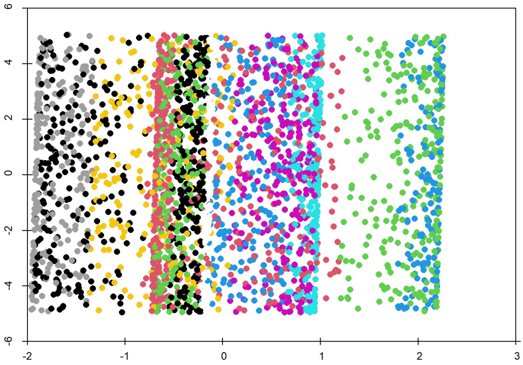
Swiss Roll

$$(X_i\ ,\ Y_i) \overset{iid}{\sim} U\left((0,4\pi)\ \times\ (0,4\pi)\right) \rightarrow \left(\frac{X_i}{5}cos(X_i)\ ,\ \frac{X_i}{5}sin(X_i)\ ,\ Y_i\right)$$



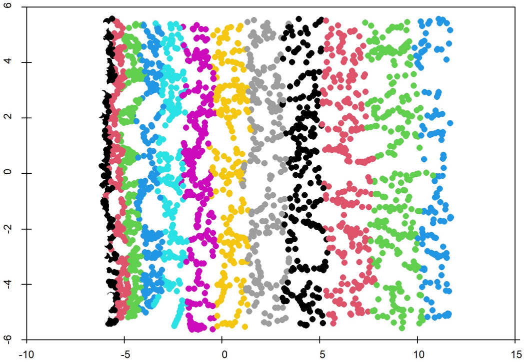
MDS dimension reduction

```
D = dist(data, method = "euclidean")
mds = cmdscale(D, k=2, eig=T)
```



Isomap dimension reduction

```
library(vegan)
iso = isomap(D, ndim = 2, k = 5)
```



Euclidean / Graph Distance Matrix

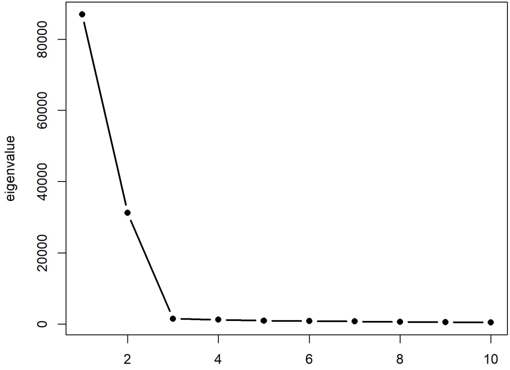
```
round(as.matrix(D)[1:6,1:6],2)
```

	1	2	3	4	5	6
1	0.00	7.80	7.21	3.82	3.11	5.87
2	7.80	0.00	1.02	4.05	10.29	2.66
3	7.21	1.02	0.00	3.47	9.60	2.75
4	3.82	4.05	3.47	0.00	6.52	2.45
5	3.11	10.29	9.60	6.52	0.00	8.70
6	5.87	2.66	2.75	2.45	8.70	0.00

```
graph.dist = isomapdist(D, k = 5)
round(as.matrix(graph.dist)[1:6,1:6],2)
```

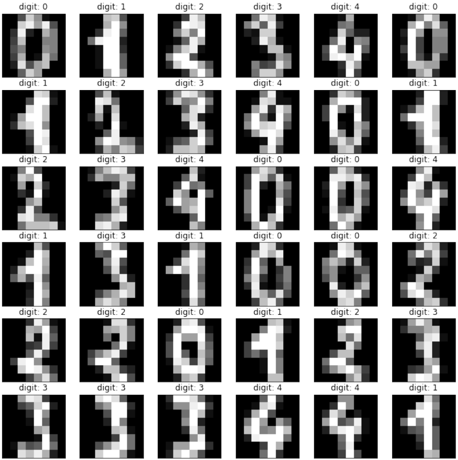
	1	2	3	4	5	6
1	0.00	8.95	8.65	4.58	7.68	14.56
2	8.95	0.00	1.43	4.60	12.82	11.53
3	8.65	1.43	0.00	4.30	11.61	10.77
4	4.58	4.60	4.30	0.00	8.78	11.98
5	7.68	12.82	11.61	8.78	0.00	10.31
6	14.56	11.53	10.77	11.98	10.31	0.00

Scree Plot

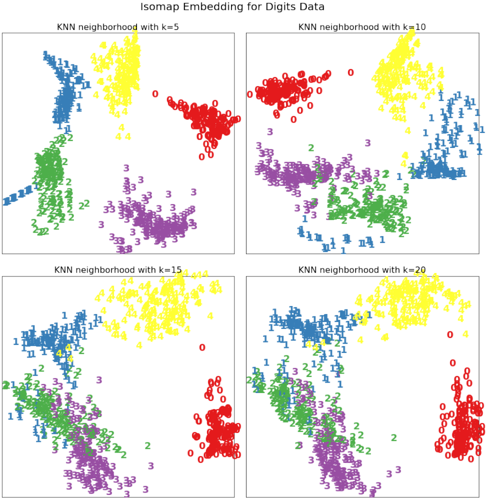


Handwritten Digits

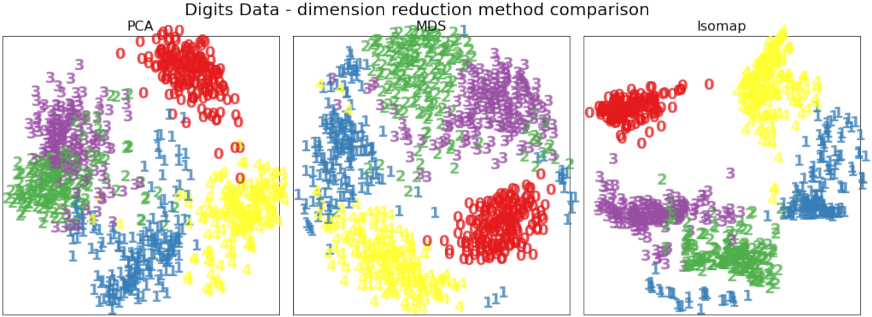
A selection from the 64-dimensional digits dataset



Comparison of different number of neighbors



Comparison with different Dimension Reduction Method



Compared to PCA and MDS, Isomap has done a good job in reducing dimensions from 64 to 2.

Conclusion

- Use graph distance to approximate geodesic distance.
- Apply MDS on the graph distance matrix.
- Deficiency : time consuming (Landmark ISOMAP)

Questions

- How to obtain the graph distance matrix? (see 25/36)
- What is the difference between standard MDS and ISOMAP? (see 5/36)

Reference

- Izenman, A. J. Modern multivariate statistical techniques. Regression, classification and manifold learning. (2008).
- J.B. Tenenbaum, V. de Silva and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, vol. 290, pp. 2319—2323, 2000.

Reference

- TD 1 - Dimensionality Reduction : https://reurl.cc/M0NVR4
- Scikit-Learn Manifold learning on handwritten digits : https://scikit-learn.org/
- Dijkstra's Algorithm by Laksman Veeravagu and Luis Barrera : https://reurl.cc/j1G201

Thank You