

# Isomap

Minh Nguyễn, Thọ Phan, Linh Võ, Thuận Dương

PiMA 2021



Trình bày: Nhóm 2, Isomap

August 8, 2021

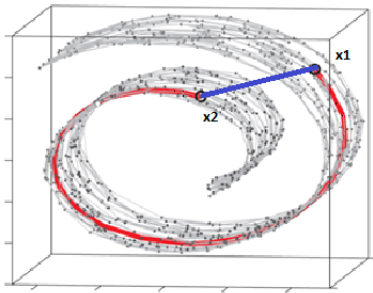
- 1 Tổng quan về Isomap
- 2 Thuật toán
- 3 Áp dụng mô hình
  - Tập dữ liệu Swiss Roll
  - Tập dữ liệu S Curve
- 4 Đánh giá & Cải tiến

- 1 Tổng quan về Isomap
- 2 Thuật toán
- 3 Áp dụng mô hình
  - Tập dữ liệu Swiss Roll
  - Tập dữ liệu S Curve
- 4 Đánh giá & Cải tiến



# Tổng quan

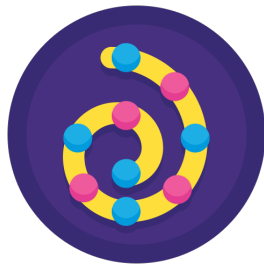
**Isomap** (isometric mapping) là thuật toán giảm chiều dữ liệu phi tuyến tính với mục tiêu bảo toàn khoảng cách đồ thị giữa các điểm dữ liệu trên đa tạp một cách tốt nhất.



# Các định nghĩa liên quan



Đa tạp



Khoảng cách đồ thị



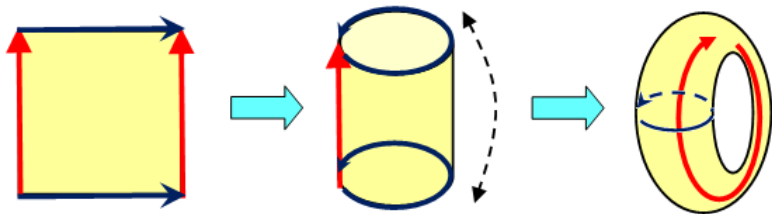
# Đa tạp / Manifold

## Definition

**Đa tạp** là một không gian topo  $n$ -chiều sao cho với mỗi điểm, ta có thể xác định được một lân cận  $U$  xung quanh điểm đó và tồn tại  $V$  thuộc không gian Euclide  $n$ -chiều sao cho  $U$  đồng phôi với  $V$ .



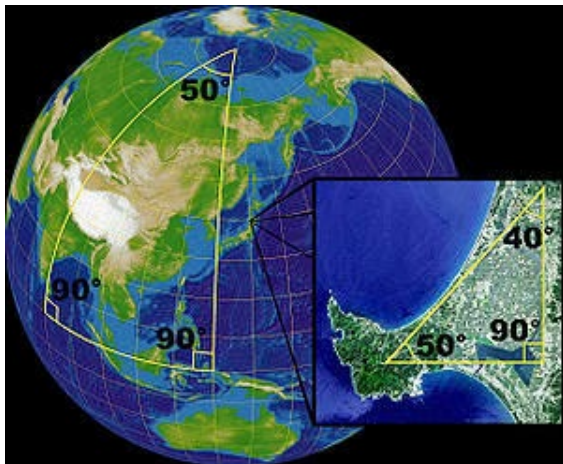
# Mô tả đa tạp



Hình: Phép đồng phôi



# Mô tả đa tạp



Hình: Bề mặt trái đất



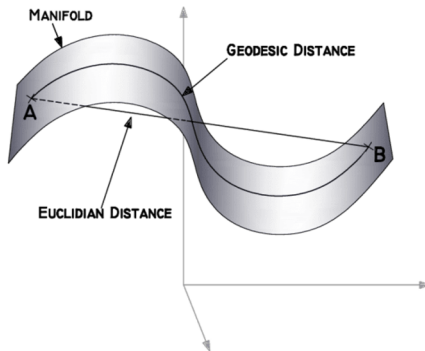
# Khoảng cách đồ thị / Geodesic distance

## Definition

**Khoảng cách đồ thị** (còn gọi là khoảng cách trên đa tạp) là đường đi ngắn nhất giữa 2 điểm trên cùng 1 đa tạp khi đi dọc theo đa tạp đó.



# Mô tả geodesic distance



# Ý tưởng chính của thuật toán Isomap

Tìm các điểm trên không gian chiều thấp sao cho khoảng cách Euclidean giữa chúng xấp xỉ với khoảng cách Geodesic trên manifold mà ta đo được ở chiều cao.



**Q:** Vì sao khoảng cách đồ thị giữa các điểm lại quan trọng?



**Q:** Vì sao khoảng cách đồ thị giữa các điểm lại quan trọng?

**A:** Nó phản ánh cấu trúc dữ liệu trên manifold ở chiều cao (dữ liệu đầu vào).

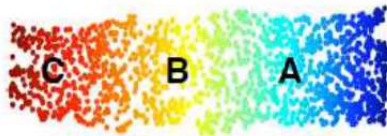


# Geodesic vs Euclidean

The Mathematics of Data Science



$$d(A,C) < d(A,B)$$



$$d(A,C) > d(A,B)$$



- 1 Tổng quan về Isomap
- 2 Thuật toán
- 3 Áp dụng mô hình
  - Tập dữ liệu Swiss Roll
  - Tập dữ liệu S Curve
- 4 Đánh giá & Cải tiến



# Input / Output

**Dữ liệu đầu vào (Input):**

$$X = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_n]$$

với  $\mathbf{x}_i \in \mathbb{R}^q$  và chiều của dữ liệu sau khi giảm là  $p$ .

**Dữ liệu đầu ra (Output):**

$$Y = [\mathbf{y}_1 \quad \mathbf{y}_2 \quad \cdots \quad \mathbf{y}_n]$$

với  $\mathbf{y} \in \mathbb{R}^p$  sao cho hàm mất mát

$$\mathcal{L}(Y) = \sum_{1 \leq i, j \leq n} (d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$

với  $d_{ij}$  là khoảng cách geodesic giữa hai điểm  $\mathbf{x}_i, \mathbf{x}_j$ .



# Thuật toán

- B1. Xây dựng đồ thị vùng lân cận (neighbor graph).
- B2. Tính khoảng cách đồ thị cho dataset.
- B3. Giảm chiều dữ liệu sử dụng cMDS.



# Bước 1: Xây dựng đồ thị

Một số thuật toán xây dựng đồ thị near neighbor graph:

- Brute - force.
- KD - tree.
- Ball tree.



# Kết quả

- Brute - force: 0.0721s
- KD - tree: 0.0166s
- Ball tree: 0.0217s



## Bước 2: Ma trận khoảng cách

Ta có thể xây dựng ma trận khoảng cách bằng các thuật toán shortest path:

- Floyd Warshall.
- Dijkstra.



## Bước 3: cMDS

Sau hai bước trên, ta có được ma trận chứa thông tin geodesic distances giữa các điểm dữ liệu:  $D = (d_{ij})_{1 \leq i, j \leq n}$ .

Nhắc lại bài toán giảm chiều dữ liệu: tìm

$$Y = [\mathbf{y}_1 \quad \mathbf{y}_2 \quad \cdots \quad \mathbf{y}_n]$$

với  $\mathbf{y} \in \mathbb{R}^p$  minimize hàm mất mát

$$\mathcal{L}(Y) = \sum_{1 \leq i, j \leq n} (d_{ij} - \|\mathbf{y}_i - \mathbf{y}_j\|)^2$$



## Bước 3: MDS

Áp dụng **Classical Multidimensional Scaling (cMDS)** với ma trận  $D$ , gồm 4 bước.

- 1 Lập ma trận bình phương khoảng cách  $D_{(2)} = (d_{ij}^2)_{1 \leq i, j \leq n}$ .
- 2 Áp dụng double centering  $K = -\frac{1}{2}HD_{(2)}H$  ( $H = I_n - \frac{1}{n}J_n$ ).
- 3 Xác định trị riêng và vector riêng tương ứng của  $K$ :

$$K = V\Lambda V^\top.$$

- 4 Chọn  $p$  trị riêng lớn nhất và các vector riêng tương ứng, được ma trận output.

$$Y_p = \Lambda_p^{1/2} V_p^\top.$$



## Bước 3: MDS

Chứng minh thuật toán:

- 1 Xét  $Y = [\mathbf{y}_1 \ \mathbf{y}_2 \ \cdots \ \mathbf{y}_n]$ ,  $Y \in \mathbb{R}^{p \times n}$  sao cho  $d_{ij} \sim \|\mathbf{y}_i - \mathbf{y}_j\|$  với  $\sum \mathbf{y}_i = \mathbf{0}$ .
- 2  $Y^\top Y \sim -\frac{1}{2} H D_{(2)} H = K$ , trong đó  $D_{(2)} = (d_{ij}^2)$ ,  $H = I_n - \frac{1}{n} J_n$ .
- 3  $K$  có biểu diễn  $K = V \Lambda V^\top$  (vì  $K$  là ma trận đối xứng).



## Bước 3: cMDS

### Bài toán tối ưu (Giảm chiều dữ liệu)

*Tìm  $Y_p$  sao cho:*

$$Y_p = \operatorname{argmin}_{Y_p} \sum_{1 \leq i, j \leq n} (d_{ij} - \|y_i - y_j\|)^2 = \operatorname{argmin}_{Y_p} \|Y_p^\top Y_p - K\|_F.$$

Áp dụng định lý Low-rank approximation, xấp xỉ tối ưu của bài toán là

$$Y_p^\top Y_p = V_p \Lambda_p V_p^\top \implies Y_p = \Lambda_p^{1/2} V_p^\top.$$

(Q.E.D)



## Bước 3: cMDS

Trường hợp thuật toán không có nghiệm tối ưu: ma trận  $K$  (*Kernel matrix*) không phải bao giờ cũng là ma trận bán xác định dương, nếu  $K$  có trị riêng âm thì  $\Lambda_p^{1/2}$  không xác định thực.

**Giải pháp (Kernel Isomap):** Chuyển đổi ma trận  $K$  thành **Mercer kernel matrix**  $K'$  bán xác định dương bằng cách sử dụng phương pháp constant-shifting.

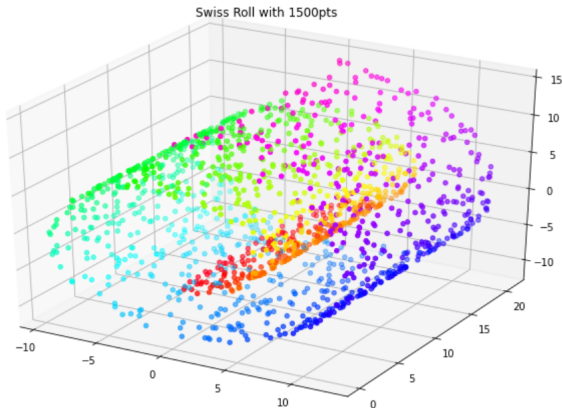


# Contents

- 1 Tổng quan về Isomap
- 2 Thuật toán
- 3 Áp dụng mô hình
  - Tập dữ liệu Swiss Roll
  - Tập dữ liệu S Curve
- 4 Đánh giá & Cải tiến



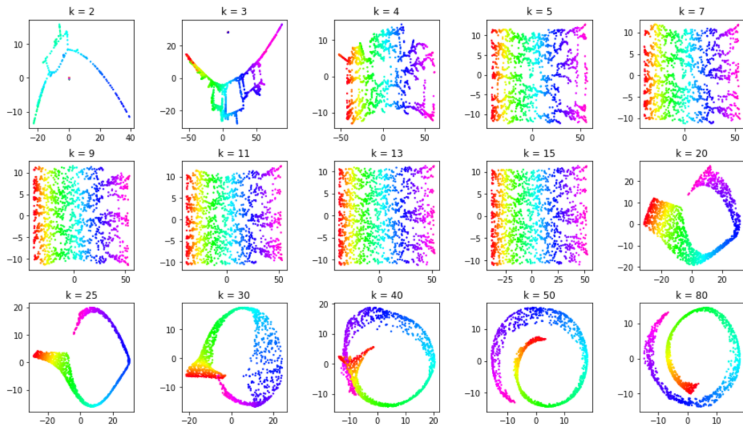
# Tập dữ liệu Swiss Roll



Hình: Tập dữ liệu Swiss Roll với 1500 điểm dữ liệu



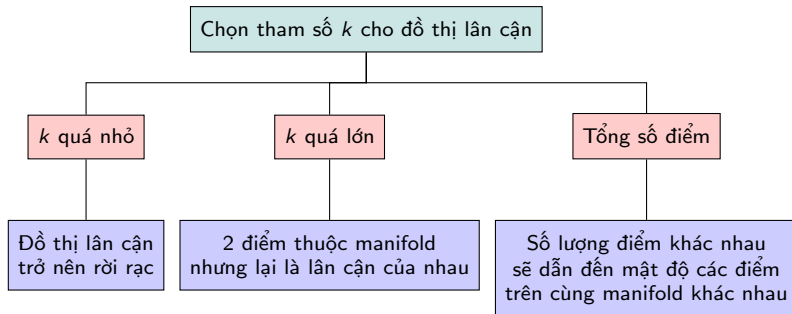
# Giảm chiều Swiss Roll bằng Isomap



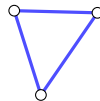
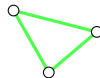
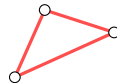
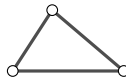
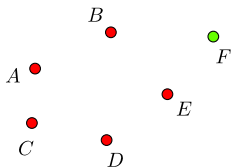
Hình: Dữ liệu sau khi xử lý bằng Isomap với  $k$  khác nhau



# Tập dữ liệu Swiss Roll - Dựng lân cận



# Trường hợp $k$ nhỏ



# Nhận xét - Tập dữ liệu Swiss Roll

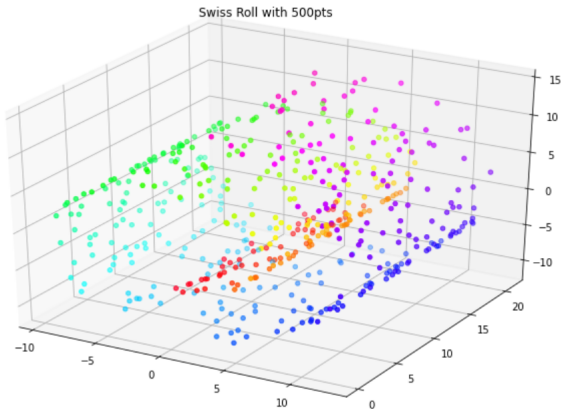
## Nhận xét 1:

Cần chọn khoảng  $k$  *phù hợp* để thuật toán chạy tối ưu

- Nếu  $k$  quá nhỏ đồ thị lân cận sẽ trở nên rời rạc dẫn đến kết quả không mong muốn.
- Nếu  $k$  quá lớn các điểm sẽ bị uốn cong lại do xảy ra tình trạng 2 điểm thuộc manifold khác nhưng lại là lân cận của nhau.



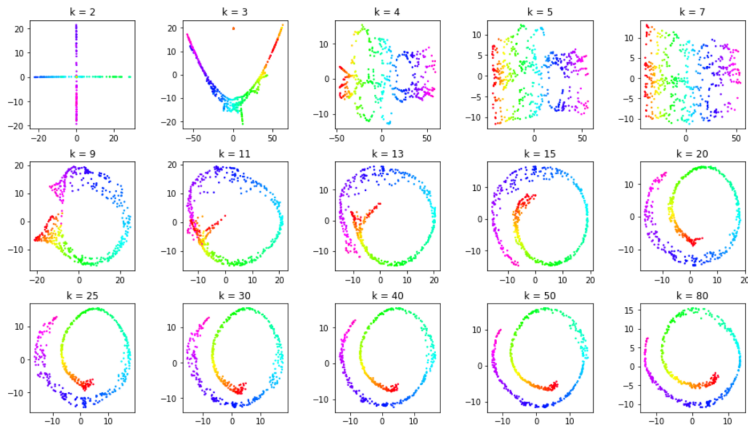
# Tập dữ liệu Swiss Roll - 500 điểm dữ liệu



Hình: Tập dữ liệu Swiss Roll với 500 điểm dữ liệu



# Giảm chiều Swiss Roll bằng Isomap - 500 điểm dữ liệu



**Hình:** Dữ liệu sau khi xử lý bằng Isomap với  $k$  khác nhau



# Tập dữ liệu Swiss Roll

## Nhận xét 2:

Cần chọn khoảng  $k$  phù hợp để thuật toán chạy tối ưu

- Việc đó sẽ phụ thuộc vào số lượng điểm có trong dataset
- Số lượng điểm khác nhau sẽ dẫn đến mật độ các điểm trên cùng manifold sẽ khác nhau

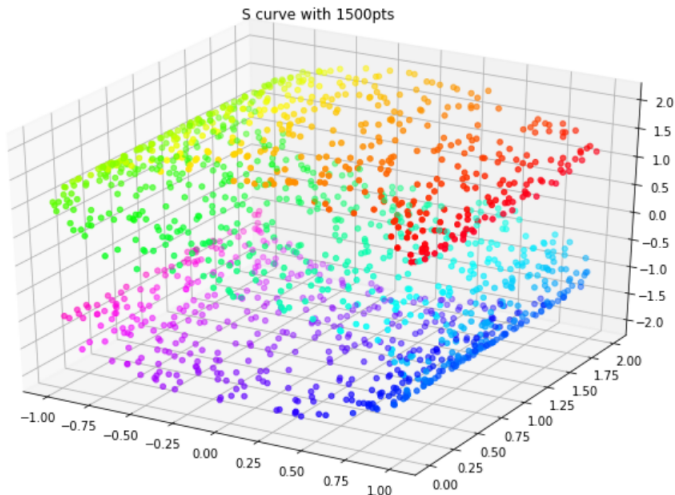


# Contents

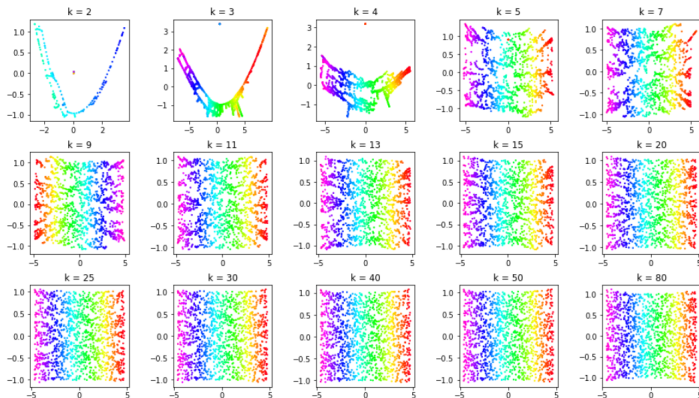
- 1 Tổng quan về Isomap
- 2 Thuật toán
- 3 **Áp dụng mô hình**
  - Tập dữ liệu Swiss Roll
  - **Tập dữ liệu S Curve**
- 4 Đánh giá & Cải tiến



# Tập dữ liệu S Curve



# Xử lý Isomap - S Curve



Hình: Giảm chiều dữ liệu tập S Curve bằng Isomap với  $k$  khác nhau.



# Tập dữ liệu S Curve

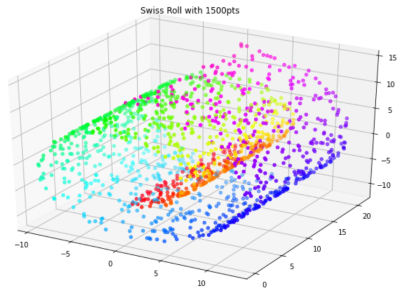
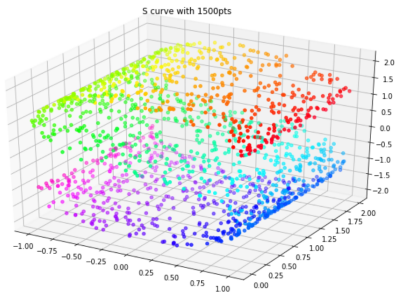
## Nhận xét:

Cần chọn khoảng  $k$  *phù hợp* để thuật toán chạy tối ưu

- 1 Chọn khoảng  $k$  *phù hợp* để thuật toán chạy tối ưu
- 2 Vùng  $k$  sẽ phụ thuộc vào số lượng điểm trong cùng 1 dataset
- 3 Vùng  $k$  sẽ phụ thuộc vào các dataset khác nhau  
Các dataset khác nhau sẽ có sự phân bố các điểm khác nhau, cấu trúc hình thành khác nhau.



# Tập dữ liệu S Curve



- 1 Tổng quan về Isomap
- 2 Thuật toán
- 3 Áp dụng mô hình
  - Tập dữ liệu Swiss Roll
  - Tập dữ liệu S Curve
- 4 Đánh giá & Cải tiến



# Hạn chế

- Nếu lân cận có chứa các lỗi sẽ khiến kết quả không như mong muốn.
- Chọn  $K$  không tốt sẽ ảnh hưởng lớn tới kết quả cuối cùng.



# Landmark - Isomap

- 1 Chọn ngẫu nhiên  $n$  điểm từ  $X$ , gọi là các điểm mốc
- 2 L-Isomap chỉ tính toán đường đi ngắn nhất từ mỗi điểm dữ liệu đến các điểm mốc
- 3 Áp dụng cMDS cho ma trận khoảng cách trắc địa  $n \cdot N$  thu được để tìm cách nhúng chiều thấp của các điểm mốc.
- 4 Việc nhúng các điểm còn lại thu được bằng một phép biến đổi tuyến tính cố định của khoảng cách trắc địa của chúng đến các điểm mốc.

→ Bằng cách này, độ phức tạp về thời gian của đường đi ngắn nhất và tính toán MDS lần lượt được giảm xuống  $O(knN \log(N))$  và  $O(n^2 N)$



# Chọn $k$ - số nearest neighbors

- 1 Chọn khoảng giá trị có thể nhận của  $K \in [K_{min}, K_{max}]$
- 2 Tính hàm mất mát  $\mathcal{L}(K)$  với mỗi  $K \in [K_{min}, K_{max}]$
- 3 Ước tính tất cả các cực tiểu của  $\mathcal{L}(K)$  và các số  $K$  tương ứng để tạo tập  $S_K$  gồm các ứng cử viên ban đầu cho giá trị  $K$  tối ưu.
- 4 Với mỗi  $K \in S_K$ , chạy thuật toán Isomap và xác định  $K_{opt}$  bằng công thức

$$K_{opt} = \underset{K}{\operatorname{argmin}} (1 - p_{D(X)D(Y)}^2)$$

( $p_{D(X)D(Y)}$  là hệ số tương quan tuyến tính tiêu chuẩn giữa  $D(X)$  và  $D(Y)$ )



# So sánh với các thuật toán khác

- 1 **Sammon mapping:** Khoảng cách được chỉnh tùy vào độ lớn các điểm
- 2 **PCA:** Tập trung vào phân bố của một chiều và bỏ qua những chiều còn lại
- 3 **T-SNE:** 2 lớp dữ liệu khác nhau được phân cách bởi các khoảng trống
- 4 **ISOMAP:** Trải phẳng nhờ vào khoảng cách đồ thị



Cảm ơn các bạn đã chú ý lắng nghe!

