

Topic Lecture: Dimensionality Reduction

Minh Thu Nguyen^a, Phuong Dinh^b, Nguyen Nguyen^c, Linh Tran

^aInternational University, VNU

^bHCMUS, VNU

^cDuke University, USA

PiMA 2021



Trình bày: Linh Tran, ****¹ University, USA

July 27, 2021

¹Tên trường bị che đi vì liên quan tới chế độ nô lệ thời xưa ở Mỹ

1 Giới thiệu

- Ký hiệu cơ bản
- Bài toán giảm chiều dữ liệu

2 Phân loại giảm chiều dữ liệu

- Phân loại theo mục đích
- Phân loại theo tính chất thuật toán

3 Thông tin thêm

- Một số ví dụ
- So sánh

Contents

1 Giới thiệu

- Ký hiệu cơ bản
- Bài toán giảm chiều dữ liệu

2 Phân loại giảm chiều dữ liệu

- Phân loại theo mục đích
- Phân loại theo tính chất thuật toán

3 Thông tin thêm

- Một số ví dụ
- So sánh



Ký hiệu cơ bản

D : Số chiều dữ liệu đầu vào (input data), n : Số điểm input data.

$$\text{Input data: } x_1 = \begin{pmatrix} x_{11} \\ x_{12} \\ \vdots \\ x_{1D} \end{pmatrix}, x_2 = \begin{pmatrix} x_{21} \\ x_{22} \\ \vdots \\ x_{2D} \end{pmatrix}, \dots, x_n = \begin{pmatrix} x_{n1} \\ x_{n2} \\ \vdots \\ x_{nD} \end{pmatrix} \in \mathbb{R}^D.$$

$$\text{Data matrix: } X = (x_{ij})_{n \times D} = \begin{bmatrix} - & - & x_1^T & - & - \\ - & - & x_2^T & - & - \\ & & \vdots & & \\ - & - & x_n^T & - & - \end{bmatrix}.$$



Thuật ngữ cơ bản

$$X = \begin{bmatrix} - & - & x_1^T & - & - \\ - & - & x_2^T & - & - \\ & & \vdots & & \\ - & - & x_n^T & - & - \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nD} \end{bmatrix}$$

Data points

Features

Các dòng (điểm x_1, x_2, \dots, x_n): Các điểm dữ liệu (data points).
 Các cột: các features, hoặc các chiều/dimension.



Contents

1 Giới thiệu

- Ký hiệu cơ bản
- Bài toán giảm chiều dữ liệu

2 Phân loại giảm chiều dữ liệu

- Phân loại theo mục đích
- Phân loại theo tính chất thuật toán

3 Thông tin thêm

- Một số ví dụ
- So sánh



Bài toán giảm chiều dữ liệu

Mục tiêu cơ bản: từ các điểm data x_1, x_2, \dots, x_n thuộc không gian nhiều chiều (high-dimensional data), tìm cho mỗi điểm x_i một điểm y_i thuộc không gian ít chiều (low-dimensional data), sao cho một số tính chất tối ưu được thỏa mãn.

Định nghĩa: y_i được gọi là biểu diễn (representation) hoặc embedding của x_i .

$$\{x_i\}_{i=1}^n \subset \mathbb{R}^D \xrightarrow{\text{Low-dimensional embedding}} \{y_i\}_{i=1}^n \subset \mathbb{R}^d$$



Mô hình giảm chiều dữ liệu trong thực tế

Input:

- High-dimensional data $x_1, x_2, \dots, x_n \in \mathbb{R}^D$.
- Số nguyên dương $d < D$: số chiều cho biểu diễn của x_i .
- Các siêu tham số (hyperparameter) đặc trưng từng mô hình.
- Một số thông tin khác (nếu có).

Output:

- Low-dimensional data $y_1, y_2, \dots, y_n \in \mathbb{R}^d$.
- (Một vài thuật toán) Hàm nhúng (embedding): $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^d$.
- Một số thông tin khác (nếu có).



Tại sao cần giảm chiều dữ liệu?

- Giúp tăng tốc độ học và tránh overfit.
- Tìm những feature ẩn, quy luật giúp giải thích data.
- Biểu diễn lại data lên mặt phẳng 2, 3 chiều \implies phân tích bằng mắt thường.
- Giảm dung lượng lưu trữ (nén).
- Tiền xử lý data để làm input cho thuật toán khác.



Phân loại bài toán giảm chiều dữ liệu

Theo mục đích bài toán: Có giám sát vs Không giám sát.

Theo tính chất của thuật toán:

- Của embedding: Tuyến tính vs Phi tuyến tính.
- Của bước tiền xử lý: Graph-based vs Non-graph-based.
- Của tính chất bất biến: có bất biến với các phép biến hình hay không?



Contents

1 Giới thiệu

- Ký hiệu cơ bản
- Bài toán giảm chiều dữ liệu

2 Phân loại giảm chiều dữ liệu

- Phân loại theo mục đích
- Phân loại theo tính chất thuật toán

3 Thông tin thêm

- Một số ví dụ
- So sánh



Phân loại theo mục đích

Giảm chiều dữ liệu có giám sát: (Feature Selection)

- Thường đi kèm một thuật toán học có giám sát.
- Tìm và giữ lại những features quan trọng nhất cho việc học có giám sát, bỏ đi những features gây nhiễu (noise).
- Giúp tăng tốc độ học và tránh overfit.



Phân loại theo mục đích bài toán

Giảm chiều dữ liệu không giám sát:

- Tìm những feature ẩn, quy luật giúp giải thích data.
- Biểu diễn lại data lên mặt phẳng 2, 3 chiều \implies phân tích bằng mắt thường.
- Giảm dung lượng lưu trữ (nén).
- Tiền xử lý data để làm input cho thuật toán khác.



Giảm chiều dữ liệu có giám sát

Input: $X = (x_{ij})_{n \times D}$ gồm các feature (cột) $\{f_i\}_{i=1}^D$, data nhãn y .

Thành phần:

- mô hình học có giám sát \mathcal{M} ,
- Hàm đánh giá $\text{Score}(\mathcal{M}, X, y)$.

Với mỗi nhóm features S , gọi X^S là data tạo từ X bằng cách chỉ lấy các features trong S .

$$\text{Ví dụ: } X^{\{1,2\}} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix}.$$

Mục tiêu: tìm S sao cho $|S| = d$ và $\text{Score}(\mathcal{M}, X^S, y)$ cao nhất
 \implies output X^S .



Giảm chiều dữ liệu không giám sát

Input: $X = (x_{ij})_{n \times D}$.

Thành phần: hàm mục tiêu $L(X, Y)$.

Mục tiêu:

- Tìm $Y = (y_{ij})_{n \times d}$ sao cho $L(X, Y)$ đạt giá trị lớn nhất/nhỏ nhất tùy thuật toán.
- [Nếu có] Tìm hàm embedding $\Phi : \mathbb{R}^D \rightarrow \mathbb{R}^d$ thỏa $y_i = \Phi(x_i)$.



Contents

1 Giới thiệu

- Ký hiệu cơ bản
- Bài toán giảm chiều dữ liệu

2 Phân loại giảm chiều dữ liệu

- Phân loại theo mục đích
- Phân loại theo tính chất thuật toán

3 Thông tin thêm

- Một số ví dụ
- So sánh



Phân loại theo tính chất thuật toán

Phân loại theo tính chất của hàm nhúng (embedding):

- Tuyến tính (Linear): Hàm embedding là hàm tuyến tính.
- Phi tuyến tính (non-linear): Hàm embedding không tuyến tính hoặc không tìm được.



Phân loại theo tính chất thuật toán

Phân loại theo tính chất của phép tiền xử lý:

- Graph-based: Xây dựng đồ thị với các đỉnh thuộc $\{x_1, \dots, x_n\}$ với nguyên lý: 2 điểm đủ gần nhau thì được nối bởi 1 cạnh.
- Non-graph-based: Tiền xử lý cách khác hoặc không tiền xử lý.



Phân loại theo tính chất thuật toán

Phân loại theo tính bất biến: Thuật toán có tính bất biến với phép biến hình F nếu $L(X, Y)$ tối ưu $\implies L(F(X), Y)$ tối ưu.

- Rotational-invariant: Bất biến với mọi phép quay.
- Translational-invariant (Bất biến với mọi phép tịnh tiến):
 $\forall c, L(X, Y)$ tối ưu $\implies L(X + c1_{n \times D}, Y)$ tối ưu.
- Scaling-invariant (Bất biến với phép vị tự):
 $\forall c, \exists c', L(X, Y)$ tối ưu $\implies L(cX, c'Y)$ tối ưu.



Contents

1 Giới thiệu

- Ký hiệu cơ bản
- Bài toán giảm chiều dữ liệu

2 Phân loại giảm chiều dữ liệu

- Phân loại theo mục đích
- Phân loại theo tính chất thuật toán

3 Thông tin thêm

- Một số ví dụ
- So sánh



Giảm chiều dữ liệu có giám sát

Cho hàm đánh giá $\text{Score}(\mathcal{M}, X, y)$, có thể:

- Forward Selection: đặt $S_0 = \emptyset$.

Lần lượt chọn các feature tốt nhất để thêm vào:

$$f_i^+ = \operatorname{argmax}_{f \notin S_{i-1}} \text{Score}(M, S_{i-1} \cup \{f\}, y).$$

Đặt $S_i := S_{i-1} \cup \{f_i^+\}$. Dừng tại S_d .

- Backwards Elimination: đặt $S_D = \{f_i\}_{i=1}^D$.

Lần lượt loại các features kém nhất:

$$f_i^- = \operatorname{argmax}_{f \in S_i} \text{Score}(M, S_i \setminus \{f\}, y).$$

Đặt $S_{i-1} := S_i \setminus \{f_i^-\}$. Dừng tại S_d .



Giảm chiều dữ liệu không giám sát

Một số ví dụ:

- **TruncatedSVD:** loại bỏ features thừa (\approx tổ hợp tuyến tính của các features khác) \implies các features mới ít phụ thuộc tuyến tính vào nhau. Tương tự: PCA.
- **Manifold Learning:** tìm một đa tạp (manifold) d chiều trong \mathbb{R}^D mà biểu diễn được data một cách tốt nhất. Ví dụ: t-SNE, Isomap, Locally Linear Embedding.



Giảm chiều dữ liệu không giám sát

	TruncatedSVD	Manifold Learning
Linear?	Y	N
Graph-based?	N	Y
Rotational-invariant?	Y	Y
Translational-invariant?	N	Y
Scaling-invariant?	Y	Y
Reconstructible?	Y	N



Contents

1 Giới thiệu

- Ký hiệu cơ bản
- Bài toán giảm chiều dữ liệu

2 Phân loại giảm chiều dữ liệu

- Phân loại theo mục đích
- Phân loại theo tính chất thuật toán

3 Thông tin thêm

- Một số ví dụ
- So sánh



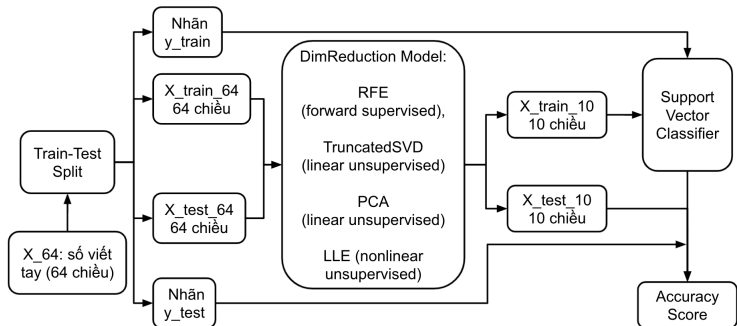
Có và không giám sát

Dataset: Số viết tay (https://scikit-learn.org/stable/auto_examples/datasets/plot_digits_last_image.html)

Phương pháp: dùng mỗi phương pháp trong RFE, TruncatedSVD, PCA và LLE để giảm chiều dữ liệu về 10 chiều, sau dùng SVC để tìm accuracy score cho từng phương pháp và so sánh.



Có và không giám sát



Kết quả: PCA và TruncatedSVD dù không giám sát vẫn có độ chính xác cao hơn RFE (95% so với 89%), trong khi LLE có độ chính xác 80%.

