

	VIETTEL AI RACE	Public 119
	LINEAR DISCRIMINANT ANALYSIS (LDA) — GIỚI THIỆU & GHI CHÚ	Lần ban hành: 1

## 1. Giới thiệu

Trong hai bài viết trước, PCA (unsupervised) giữ lại tổng phương sai lớn nhất nhưng không dùng nhãn. Trong phân lớp (supervised), tận dụng nhãn thường cho kết quả tốt hơn. Ví dụ chiếu lên các hướng  $d_1$  (gần PC1) và  $d_2$  (gần thành phần phụ):  $d_1$  có thể làm hai lớp chồng lấn, trong khi  $d_2$  tách tốt hơn cho classification. Điều này cho thấy giữ lại nhiều phương sai nhất không phải lúc nào cũng tốt cho phân lớp. LDA ra đời để tìm phép chiếu tuyến tính (projection matrix) tối đa hóa khả năng phân biệt (discriminant). Với  $C$  lớp, số chiều mới không vượt quá  $C-1$ .

## 2. LDA cho bài toán với 2 classes

### 2.1 Ý tưởng cơ bản

Discriminant tốt khi: (i) mỗi lớp tập trung (within-class variance nhỏ), (ii) các lớp cách xa nhau (between-class variance lớn).

### 2.2 Xây dựng hàm mục tiêu

Ký hiệu: dữ liệu  $x_n$ , phép chiếu  $y_n = w^T x_n$ .

Kỳ vọng mỗi lớp:  $m_k = (1/N_k) \sum_{n \in C_k} x_n$ ,  $k=1,2$ . (1)

Hiệu kỳ vọng sau chiếu:  $m_1 - m_2 \Rightarrow w^T(m_1 - m_2)$ . (2)

Within-class variances (không lấy trung bình):  $s_k^2 = \sum_{n \in C_k} (y_n - m_k)^2$ . (3)

Ma trận between-class:  $S_B = (m_1 - m_2)(m_1 - m_2)^T$ . (5)

Ma trận within-class:  $S_W = \sum_{k=1}^2 \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T$ . (6)

**Hàm mục tiêu Fisher (2 lớp):**

$$J(w) = (w^T S_B w) / (w^T S_W w). \quad (4,7)$$

### 2.3 Nghiệm tối ưu (Fisher's linear discriminant)

Đạo hàm và sắp xếp lại:

$$S_W^{-1} S_B w = J(w) w. \quad (10)$$

**Chọn nghiệm tỷ lệ:**

$$w = \alpha S_W^{-1} (m_1 - m_2), \quad \alpha \neq 0. \quad (11)$$

## 3. LDA cho multi-class classification

Phép chiếu tuyến tính:  $y = W^T x$ ,  $W \in \mathbb{R}^{D \times D'}$ . Không dùng bias.

**Within-class tổng quát:**

	VIETTEL AI RACE	Public 119
	LINEAR DISCRIMINANT ANALYSIS (LDA) — GIỚI THIỆU & GHI CHÚ	Lần ban hành: 1

$s_W = \text{trace}(W^T S_W W)$ , với  $S_W = \sum_{k=1}^C \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T$ . (17–19)

**Between-class tổng quát:**

$s_B = \text{trace}(W^T S_B W)$ , với  $S_B = \sum_{k=1}^C N_k (m_k - m)(m_k - m)^T$ . (21–22)

**Hàm mục tiêu multi-class:**

$$J(W) = \text{trace}(W^T S_B W) / \text{trace}(W^T S_W W).$$

Điều kiện tối ưu bậc nhất:

$$S_W^{-1} S_B W = J W. \quad (24)$$

Các cột của  $W$  là các eigenvectors ứng với các trị riêng lớn nhất của  $S_W^{-1} S_B$ .

Bổ đề:  $\text{rank}(S_B) \leq C - 1 \Rightarrow$  số chiều tối đa sau LDA  $\leq C - 1$ .

#### 4. Ví dụ nhanh (Python, phác thảo)

- Tạo dữ liệu 2 lớp:  $X_0, X_1 \in R^{N \times D}$ ; tính  $m_0, m_1$ .
- $S_B = (m_0 - m_1)(m_0 - m_1)^T$ ;  $S_W = \sum (x - m_k)(x - m_k)^T$ .
- $W$  từ eigenvectors của  $\text{inv}(S_W) @ S_B$ ; so sánh `sklearn.discriminant_analysis.LDA`.

#### 5. Thảo luận

LDA là phương pháp supervised giảm chiều và/hoặc phân lớp: tối ưu small within-class & large between-class. Số chiều tối đa sau LDA là  $\leq C-1$ . Giả định thường gặp: phân phối gần Gaussian, các ma trận hiệp phương sai giữa các lớp gần nhau. LDA tốt khi các lớp gần linearly separable; kém hiệu quả nếu không tách tuyến tính.