1 Linear regression

Mô hình: $\hat{y}(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T\mathbf{x}$

Hợp lý cực đại

$p(t_n|\mathbf{x}_n, \mathbf{w}, \beta) = \mathcal{N}(t_n|\mathbf{w}^T\mathbf{x}_n, \beta^{-1})$

$L(\mathbf{w}, \beta) = \beta E_D(\mathbf{w}) - \frac{N}{2}\log\beta + \frac{N}{2}\log(2\pi)$

với: $E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^T\mathbf{x}_n)^2$

Nghiệm giải tích

Cực tiểu $E_D(\mathbf{w})$: $\mathbf{w}_{ML} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}$

$\beta_{ML}^{-1} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \mathbf{w}_{ML}^T\mathbf{x}_n)^2$

Giải thuật lặp (Gradient Descent)

$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta\nabla E_D(\mathbf{w})$

Hồi quy cho quan hệ phi tuyến

$\mathbf{x} \to \boldsymbol{\phi}(\mathbf{x}) = [\phi_0(\mathbf{x}), \ldots, \phi_{M-1}(\mathbf{x})]$

Dự báo: $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}_{ML}$

Các độ đo: $\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}(t_n - \hat{y}_n)^2$

$\text{RMSE} = \sqrt{\text{MSE}}$

Hạn chế quá khớp: Ridge Regression

$L(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^T\mathbf{x}_n)^2 + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w}$

Nghiệm: $\mathbf{w}_{ridge} = (\lambda\mathbf{I} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t}$

LASSO: $L(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}(t_n - \mathbf{w}^T\mathbf{x}_n)^2 + \lambda\sum_{m=1}^{M}|w_m|$

Dự báo cho nhiều biến

Với K biến đầu ra: $\mathbf{W} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{T}$

Trong đó: $\mathbf{T} \in \mathbb{R}^{N\times K}, \mathbf{W} \in \mathbb{R}^{M\times K}$

2 Logistic regression

Hàm sigmoid: $\sigma(z) = \frac{1}{1+e^{-z}}$

Mô hình: $\hat{y} = p(C_1|x, w) = \sigma(w^Tx)$

Nếu $\hat{y} \geq \lambda$ thì $x \in C_1$. Ngược lại, $x \in C_0$

Xây dựng hàm mục tiêu

Với một điểm dữ liệu (x, y):

$p(y|x, w) = \hat{y}^y(1-\hat{y})^{1-y}$

Với N điểm dữ liệu:

$p(t|X, w) = \prod_{n=1}^{N}\hat{y}_n^{y_n}(1-\hat{y}_n)^{1-y_n}$

Neg log likelihood (BCE): $L(w) = -\sum_{n=1}^{N}[y_n\log\hat{y}_n + (1-y_n)\log(1-\hat{y}_n)]$

Tìm hệ số của mô hình

$\nabla L(w) = \sum_{n=1}^{N}(\hat{y}_n - y_n)x_n = X^T(\hat{y} - y)$

Giải thuật lặp với đạo hàm bậc 2

Ma trận Hessian: $H = \nabla^2 L(w) = \sum_{n=1}^{N}\hat{y}_n(1-\hat{y}_n)x_nx_n^T = X^TRX$

R là ma trận đường chéo: $R_{nn} = \hat{y}_n(1-\hat{y}_n)$

Method: GD, Newton–Raphson, IRLS

3 Softmax regression

Mỗi nhãn được mã hóa dưới dạng véctơ one-hot kích thước K.

Mô hình dự báo

Ma trận tham số của mô hình:

$W = [w_1, w_2, \ldots w_K]^T \in \mathbb{R}^{K\times M}$

Các bước tính toán: $Z = XW^T \quad (N\times K)$, $\hat{Y} = \text{softmax}(Z)$

Hàm softmax: $\hat{y}_k = \frac{\exp(z_k)}{\sum_{i=1}^{K}\exp(z_i)}$

Nhãn dự đoán: prediction $= \arg\max_k \hat{y}_k$

Hàm hợp lý

Xác suất của tập nhãn:

$p(t|X, W) = \prod_{n=1}^{N}\prod_{k=1}^{K}\hat{y}_{n,k}^{y_{n,k}}$

Hàm mất mát Cross-Entropy

Minimize negative log-likelihood function

$L(W) = -\sum_{n=1}^{N}\sum_{k=1}^{K}y_{n,k}\log(\hat{y}_{n,k})$

Gradient Descent

Gradient loss với softmax: $\frac{\partial L}{\partial z} = (\hat{y} - y)^T$

Gradient theo tham số: $\Delta W = (\hat{y} - y)^T x^T$

Cập nhật trọng số: $W \leftarrow W - \eta\Delta W$

4 MLP (ANN)

Forward Pass

Let $h^{(0)} = x$; $h^{(l)} = \phi\left(W^{(l)}h^{(l-1)} + b^{(l)}\right)$, $l = 1, \ldots, L$; $\phi(\cdot)$ is activation function.

Output Layer

Regression: $\hat{y} = W^{(L+1)}h^{(L)} + b^{(L+1)}$

Classification: $\hat{p}_k = \frac{\exp(w_k^\top h^{(L)} + b_k)}{\sum_j \exp(w_j^\top h^{(L)} + b_j)}$

Function Composition View

$f(x; \theta) = f^{(L+1)} \circ \phi \circ f^{(L)} \circ \cdots \circ \phi \circ f^{(1)}(x)$

Fully Connected (Linear) Layer

Single sample: $y = Wx + b$

Mini-batch $X \in \mathbb{R}^{B\times N}$: $Y = XW^\top + \mathbf{1}b^\top$

Activation Functions

Sigmoid (Vanishing, no 0-centered): $\sigma(z) = \frac{1}{1+e^{-z}}$; $\sigma' = \sigma(1-\sigma)$

$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ $\tanh' = 1 - \tanh^2$ (Saturation)

$\text{ReLU}(z) = \max(0, z)$; $\text{ReLU}' = 0$ $(z < 0)$, $= 1$ $(x > 0)$ (Dead neuron)

$\text{SiLU}(z) = z\sigma(z)$,

$\text{SiLU}' = \sigma(z) + z\sigma(z)(1-\sigma(z))$

$$\text{LReLU}(z) = \begin{cases} z, & z \geq 0 \\ \alpha z, & z < 0 \end{cases}$$

5 Training ANN

Problem Setup: Dataset $\{(x_i, y_i)\}_{i=1}^{n}$, a model $\hat{y}_i = f_\theta(x_i)$, training aims to solve: $\min_\theta \; L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, \hat{y}_i)$.

Regression Losses:

$L_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. (outliner sensitive)

$L_{\text{MAE}} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$. (non 0 differentiable, converge slower.)

Huber Loss: $e_i = y_i - \hat{y}_i$

$$L_\delta(e_i) = \begin{cases} \frac{1}{2}e_i^2, & |e_i| \leq \delta, \\ \delta(|e_i| - \frac{1}{2}\delta), & |e_i| > \delta, \end{cases}$$

Classification Losses

BCE: For $y_i \in \{0, 1\}$ and $p_i = \sigma(z_i)$: $L_{\text{BCE}} = -\frac{1}{n}\sum_{i=1}^{n}[y_i\log p_i + (1-y_i)\log(1-p_i)]$.

Categorical CE: For K classes one-hot: $L_{\text{CE}} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}y_{ik}\log p_{ik}$.

Training Process: Shuffle dataset

Forward: compute predictions and loss.

Backward: compute grads via backprop.

Update: update param with optimizer.

Training Algorithm (SGD)

Initialize parameters θ.

For epoch $= 1$ to E do:

Shuffle dataset and create mini-batches.

For each mini-batch (X, y) do:

Perform a forward pass.

Compute loss L.

Perform a backward pass and compute gradient $\nabla_\theta L$.

Update parameters: $\theta \leftarrow \theta - \eta\nabla_\theta L$.

SGD with Momentum:

$v_t = \mu v_{t-1} + g_t, \quad \theta \leftarrow \theta - \eta v_t$.

Adam: $m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$, $v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2$, $\theta \leftarrow \theta - \eta\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$.

AdamW: decouples weight decay from gradient update

6 Layers

Fully connected Layer

$y = W \times x + b, \Delta X = W^T \times \Delta y$

$\Delta W = \Delta y \times x^T, \Delta b = \Delta y$

CNN

$o_1 = \lfloor\frac{i_1 + 2p_1 - k_1}{s_1}\rfloor + 1$

$Y = X * W, *$ is convolution, not matmul.

$\Delta X = Rot180°(W) * \Delta Y$, full padding;

$\Delta W = Rot180°(\Delta Y) * X$, no padding;

7 SVM primal problem

Ma trận dữ liệu: $X \in \mathbb{R}^{N\times(M-1)}$.

Nhãn: $\mathbf{t} = [t_1, t_2, \ldots, t_N]^T, t_n \in \{-1, +1\}$

Xác định boundary $\mathbf{w}^T\mathbf{x} + b = 0$ sao cho **lề (margin)** giữa hai lớp là lớn nhất.

Siêu phẳng $(M-1)$ chiều: $\mathbf{w}^T\mathbf{x} + b = 0$

Từ $\mathbf{x}$ đến siêu phẳng: $d(\mathbf{x}) = \frac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$

Hàm quyết định: $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$

Quy tắc phân lớp: $\text{class}(\mathbf{x}) = \text{sign}(y(\mathbf{x}))$

Lề (Margin): $m_\mathbf{w} = \min_n \frac{t_n(\mathbf{w}^T\mathbf{x}_n + b)}{\|\mathbf{w}\|}$

Chuẩn hóa: $m_\mathbf{w} = \frac{1}{\|\mathbf{w}\|}$

Cực đại lề: chuẩn hóa $t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$

Hàm mục tiêu Cực đại lề tương đương: $\min_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2$ với ràng buộc:

$t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \quad n = 1, \ldots, N$

Bài toán gốc:

$\mathbf{w}^*, b^* = \arg\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2$

s.t. $\quad t_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \quad \forall n$

Hàm mục tiêu và các hàm ràng buộc: lồi và khả vi

CVXOPT: $\arg\min_\mathbf{x}\frac{1}{2}\mathbf{x}^TK\mathbf{x} + \mathbf{p}^T\mathbf{x}$

s.t. $G\mathbf{x} \leq \mathbf{h}$; $\mathbf{x} = [w_1, w_2, \ldots, w_{M-1}, b]^T$

$K_{M\times M}$ là ma trận đơn vị với $K_{M,M} = 0$.

$G_{N\times M}$ có $G_{i,j} = -t_ix_{i,j}$ và $G_{i,M} = -t_i$.

$p_{N\times 1}$ chỉ chứa số 0; $h_{N\times 1}$ chỉ chứa số -1.

8 SVM dual problem

Hàm Lagrangian $L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N}\alpha_n\left[t_n(w^Tx_n + b) - 1\right]$, với $\alpha_n \geq 0$

Convex theo w, b ($\|w\|^2$ bán định dương)

Concave theo α, là hàm affine theo α

Là chặn dưới của hàm mục tiêu gốc

(KKT-1) ĐK dừng: $\nabla_{w,b}L(w, b, \alpha) = 0$

(KKT-2) Ràng buộc gốc

(KKT-3) Ràng buộc đối ngẫu: $\alpha_n \geq 0$

(KKT-4) ĐK bù: $\alpha_n[1 - t_n(w^Tx_n + b)] = 0$

Xây dựng hàm đối ngẫu

$\frac{\partial L}{\partial w} = w - \sum_{n=1}^{N}\alpha_nt_nx_n = 0$

$\frac{\partial L}{\partial b} = \sum_{n=1}^{N}\alpha_nt_n = 0$

Suy ra: $w = \sum_{n=1}^{N}\alpha_nt_nx_n$.

Thay Lagrangian, hàm đối ngẫu: $g(\alpha) = \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{r=1}^{N}\sum_{c=1}^{N}\alpha_r\alpha_ct_rt_cx_r^Tx_c$.

Ta cần argmax hàm đối ngẫu

Bài toán đối ngẫu: $(K_{rc} = t_rt_cx_r^Tx_c.)$

$\alpha^* = \arg\min_\alpha \; \frac{1}{2}\alpha^TK\alpha - \mathbf{1}^T\alpha$

s.t. $\alpha_n \geq 0, n = 1, \ldots, N, \sum_{n=1}^{N}\alpha_nt_n = 0$

CVXOPT $\boldsymbol{\alpha}^* = \text{argmin}_\alpha\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{K}\boldsymbol{\alpha} + \boldsymbol{p}^T\boldsymbol{\alpha}$

s.t. $G\alpha \leq h$, $A\alpha = b$. Với $K = K_{\text{Gram}} \odot Y$ $(K_{Gram} = XX^T, Y = yy^T)$, có shape $(N \times N)$ - rất lớn

$p_{N\times 1}$ chỉ chứa -1; $h_{N\times 1}$ chỉ chứa 0

$G_{N\times N}, G_{ii} = -1$; $A_{1\times N} = y^T$; $b_{1\times 1} = [0]$

Tiêu chuẩn Slater: Vì tồn tại (w, b) : $t_n(w^Tx_n + b) > 1$, $\forall n$, nên bài toán thỏa Slater. Do đó: $\min_{w,b}\max_\alpha L(w, b, \alpha) = \max_\alpha\min_{w,b}L(w, b, \alpha)$.; duality gap = 0.

Với tập véc-tơ hỗ trợ $S = \{n : \alpha_n > 0\}$, $y(x) = \sum_{n\in S} \alpha_n t_n x_n^T x + b$.
$b = \frac{1}{N_S} \sum_{m\in\mathcal{S}} \left(t_m - \sum_{n\in\mathcal{S}} \alpha_n t_n \boldsymbol{x}_n^T \boldsymbol{x}_m\right)$

9 SVM soft margin

Ràng buộc mới: $t_n(w^T x_n + b) \geq 1 - \xi_n, \quad n = 1,\ldots,N, \quad \xi_n \geq 0$.
Hàm mục tiêu mới:
$f_0(w, b, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{n=1}^N \xi_n$,
$C > 0$ siêu tham số: C càng nhỏ thì mức phạt càng thấp, và lề rộng hơn với kỳ vọng tổng quát hóa tốt hơn
Bài toán: $\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{n=1}^N \xi_n$ s.t. $\quad t_n(w^T x_n + b) \geq 1 - \xi_n, \xi_n \geq 0, \forall n$.
Hàm Lagrangian: $L(w, b, \xi, \alpha, \mu) = \frac{1}{2}\|w\|^2 + C\sum_{n=1}^N \xi_n - \sum_{n=1}^N \alpha_n[t_n(w^T x_n + b) - 1 + \xi_n] - \sum_{n=1}^N \mu_n \xi_n$.
Bài toán đối ngẫu
$\min_\alpha \frac{1}{2}\sum_{r=1}^N \sum_{c=1}^N \alpha_r \alpha_c t_r t_c x_r^T x_c - \sum_{n=1}^N \alpha_n$ s.t. $0 \leq \alpha_n \leq C, \sum_{n=1}^N \alpha_n t_n = 0$
Công thức dự báo: Sau khi tìm được α và b: $y(x) = \sum_{n\in S} \alpha_n t_n x_n^T x + b$
$b = \frac{1}{N_\mathcal{M}} \sum_{m\in\mathcal{M}} \left(t_m - \sum_{n\in\mathcal{S}} \alpha_n t_n \boldsymbol{x}_n^T \boldsymbol{x}_m\right)$, M (Margin SupVec Set): $\{t_m y(x_m) = 1\}$
CVXOPT: $\min_\alpha \frac{1}{2}\alpha^T K\alpha + p^T\alpha$
s.t. $G\alpha \leq h,\ A\alpha = b$.
$G_{2N\times N}$: đường chéo nửa trên -1, đường chéo nửa dưới 1, còn lại 0; $H_{2N\times 1}$: nửa trên 0, nửa dưới C
Các ràng buộc hộp $0 \leq \alpha_n \leq C$ được mã hóa trong ma trận G và h.

10 SVM kernel

Bài toán đối ngẫu của SVM lề mềm:
$\alpha^* = \arg\min_\alpha \frac{1}{2}\alpha^T K\alpha + p^T\alpha$
Trong đó, ma trận kernel K được xác định bởi tích vô hướng giữa các điểm dữ liệu.
Tính $\langle\Phi(x_i), \Phi(x_j)\rangle$ qua kernel: $k(x_i, x_j)$
Mercer: $k(x_i, x_j)$ là kernel hợp lệ nếu: Đối xứng: $k(x_i, x_j) = k(x_j, x_i)$; Bán định dương: $\sum_{i=1}^N \sum_{j=1}^N c_i c_j k(x_i, x_j) \geq 0$
Các kernel thông dụng
Linear: $k(x, x') = x^T x'$
Polynomial: $k(x, x') = (\gamma x^T x' + r)^d$
RBF (Radial Basic Function - Gaussian): $k(x, x') = \exp(-\gamma\|x - x'\|^2)$
Sigmoid: $k(x, x') = \tanh(\gamma x^T x' + r)$
Thiết kế Kernel: $k(x_i, x_j)$ lớn nếu x_i, x_j cùng lớp; $k(x_i, x_j)$ nhỏ nếu khác lớp

11 PCA

Chiếu x lên u: $l_{xou} = \frac{u^T x}{||u||}$
Tập dữ liệu $X \in \mathbb{R}^{N\times D}$; Giảm số chiều từ D xuống M với $M \ll D$; Các đặc trưng mới không còn tương quan tuyến tính.
Phương sai và hiệp phương sai: Trung bình của dữ liệu: $\mu = \frac{1}{N}\sum_{n=1}^N x_n$
Dữ liệu được chuẩn hóa: $z_n = x_n - \mu$
Ma trận hiệp phương sai:
$S = \frac{1}{N}\sum_{n=1}^N (x_n - \mu)(x_n - \mu)^T$
$= Z^T Z = (X - \mu)^T(X - \mu)$ (size $D \times D$)
$Au = \lambda u$, u là eigenvector, λ là eigenvalue.
Cực đại hóa phương sai: vectơ đơn vị u, phương sai dữ liệu chiếu lên u là: $\sigma^2 = u^T Su$
Bài toán tối ưu: $\max_u u^T Su \;$ s.t. $u^T u = 1$
Dùng nhân tử Lagrange dẫn đến: $Su = \lambda u$
Trục chính của PCA là các eigenvector của S; Phương sai tương ứng là các eigenvalue.
Det: $|A_{2\times 2}| = a_{11}a_{22} - a_{12}a_{21}$;
$|A_{3\times 3}| = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{32}a_{21} - a_{11}a_{23}a_{32} - a_{22}a_{13}a_{31} - a_{12}a_{21}a_{33}$
$U = [u_1, u_2, \ldots u_D], D = [\lambda_1, \lambda_2, \ldots, \lambda_D]I$
Tính chất: $\mathbf{U}^T = \mathbf{U}^{-1}$; $\mathbf{SU} = \mathbf{UD}$; $\mathbf{S} = \mathbf{UDU}^{-1} = \mathbf{UDU}^T = \sum_{k=1}^D \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^T$; $\mathbf{U}^{-1}\mathbf{SU} = \mathbf{U}^T\mathbf{SU} = \mathbf{D}$
Thu giảm số chiều Chọn M eigenvector tương ứng với M eigenvalue lớn nhất, tạo thành ma trận: $\hat{U} = [u_1, u_2, \ldots, u_M]$
Chiếu dữ liệu: $X_{\text{PCA}} = (X - \mu^T)\hat{U}$
Phục hồi xấp xỉ: $\hat{X} = \mu^T + X_{\text{PCA}}\hat{U}^T$
SVD: Phân rã SVD: $X = USV^T$
$\mathbf{U}_{N\times N}$; cột là eigenvector của $\mathbf{XX}^T$
$\mathbf{V}_{D\times d}$; cột là eigenvector của $\mathbf{X}^T\mathbf{X}$
Ma trận đường chéo $\mathbf{S}_{N\times D}$; giá trị là các singular values từ lớn đến nhỏ
Giải thuật SVD: Tính $\mathbf{Z} = \mathbf{X} - \mathbf{m}^T$, với $\mathbf{m}$ là **total mean**, $\mathbf{m} = \frac{1}{N}\sum_{n=1}^N \mathbf{x}_n$.; Dùng SVD phân rã $\mathbf{Z} = \mathbf{USV}^T$; Chọn M véc-tơ đầu tiên của $\mathbf{V}$ được ma trận $\hat{\mathbf{V}}$; Chiếu dữ liệu trong $\mathbf{Z}$ lên M eigenvector: $\mathbf{X}_{\text{pca}} = \mathbf{Z}\hat{\mathbf{V}}$.

12 LDA

Tập dữ liệu: $X \in \mathbb{R}^{N\times D}$, với D thường rất lớn. $t_k \in \{1, 2, \ldots, C\}$ là nhãn lớp của điểm dữ liệu thứ $k \Rightarrow$ Giảm số chiều từ D xuống M, với $M \leq C - 1$. Dữ liệu có độ phân tách giữa các lớp là lớn nhất.
Tâm của mỗi lớp
Với lớp k: $\mathbf{m}_k = \frac{1}{N_k}\sum_{n\in C_k} \mathbf{x}_n$
Between-class scatter matrix
$S_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$
Within-class scatter matrix
$S_W = \sum_{k=1}^C \sum_{n\in C_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T$
Hàm mục tiêu của Fisher
$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$
Mục tiêu: $\mathbf{w}^* = \arg\max_\mathbf{w} J(\mathbf{w})$
Tìm nghiệm: Giải bài toán tối ưu dẫn đến phương trình trị riêng: $S_W^{-1} S_B \mathbf{w} = \lambda\mathbf{w}$
Hướng chiếu tối ưu là: $\mathbf{w} \propto S_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1)$
Cần chuẩn hóa w có độ dài bằng 1.
Trường hợp C lớp:
$\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_M]$
Hàm mục tiêu: $J(W) = \frac{\text{trace}(W^T S_B W)}{\text{trace}(W^T S_W W)}$
Số chiều tối đa: $M \leq C - 1$
Do ma trận S_B có hạng tối đa là $C - 1$.
Giải thuật LDA: Tính S_B và S_W. Tính $A = S_W^{-1} S_B$. Thực hiện SVD hoặc eigen-decomposition. Chọn M eigenvector tương ứng với eigenvalue lớn nhất. Chiếu dữ liệu: $\hat{X} = (X - m^T)W$.

13 Ensemble

Variance of ensemble regression: $\text{Var}\left(\frac{1}{M}\sum_{m=1}^M \hat{y}^{(m)}\right) \approx \frac{1}{M^2}\sum_{m=1}^M \text{Var}(\hat{y}^{(m)})$
Bagging (Bootstrap Aggregating)
Training dataset $D = \{(x_i, y_i)\}_{i=1}^n$,
(1) Draw M bootstrap datasets by sampling n points with replacement from D. (2) Train base learner on each bootstrap to obtain models $h_1, h_2, \ldots, h_M$. (3) Combine predictions of all models:
$\hat{y}(x) = \begin{cases} \frac{1}{M}\sum_{m=1}^M h_m(x), & \text{regression,} \\ \text{majority vote,} & \text{classification.} \end{cases}$
Random Forest: (1) Sample bootstrap dataset from train set. (2) Grow tree by recursively splitting nodes. (3) At each split, randomly select subset of features $F_{\text{sub}} \subset \{1, \ldots, d\}$. (4) Choose best split using only features in F_{sub}. $|F_{\text{sub}}| = \sqrt{d}$ for classification, $|F_{\text{sub}}| = d/3$ for regression.
Boosting: train models sequentially, where each model focuses on samples misclassified by previous ones. Reduce both bias and variance.
AdaBoost: For binary classification with $y_i \in \{-1, +1\}$, AdaBoost maintains a weight distribution over training samples. At iteration t, a weak learner h_t is trained using weighted data. final classifier is $H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$, where α_t is determined by weighted classification error of h_t.
Gradient Boosting: views ensemble construction as gradient descent in function space. Each iteration, new weak learner is fitted to negative gradient of loss function with respect to current predictions.
Voting: majority or probability averaging
Stacking: trains a meta-learner on predictions of base models. To avoid overfitting, cross-validation is used to generate out-of-fold predictions, which are then used as inputs for meta-model.

14 Genetic algorithm

Solution encoded as chromosome.
Encoding: Binary; Real-valued; Tree Permutation;
Fitness Function: Eval quality of solution x. For min problems, transformation e.g. $f_{\max}(x) = \frac{1}{1+f_{\min}(x)}$
Selection: Roulette wheel selection; Rank selection; Tournament selection; Elitism
Crossover: Single-point crossover; Two-point crossover; Uniform crossover; Arithmetic crossover (for real-valued encoding)
Mutation: Bit flipping for binary encoding; Gaussian or uniform noise for real-valued encoding; Swap or inversion for permutation encoding
GA: (1) Init population of N individuals (2) Eval fitness each individual (3) Repeat: (a) Select parents based on fitness (b) Apply crossover to generate offspring (c) Apply mutation to offspring (d) Evaluate fitness of new individuals (e) Form next generation (with optional elitism)
Break conditions: max # of gen, fitness convergence, stagnation, or time limits.
Parameters and Tuning: Population size (N = 20–200); Crossover probability (p_c = 0.6–0.9); Mutation probability (p_m = 0.001–0.1)
Variants and Extensions: RCGA; DE; GP; NSGA-II