SOẠN NHẬN DẠNG MẪU

Mục lục

Câu 1:

- A là ma trận vuông cấp n, phương trình eigenvector (trị riêng) được định nghĩa:

$$Au_i = \lambda_i u_i \ (i = 1,2, \dots, n)$$

Trong đó: u_i là một vecto riêng và λ_i là giá trị riêng tương ứng.

$$\Rightarrow \begin{cases} Au_1 = \lambda_1 u_1 & (1) \\ Au_2 = \lambda_2 u_2 & (2) \end{cases}$$

- Nhân bên trái (1) với u_2^T:

$$u_2^T A u_1 = \lambda_1 u_2^T u_1$$

(vì λ_i là số vô hướng nên $u_j^T(\lambda_i u_i) = \lambda_i u_j^T u_i$)

Tương tự đối với (2), nhân bên trái với u_1^T:

$$u_1^T A u_2 = \lambda_2 u_1^T u_2$$

- Vì A là ma trận đối xứng $A^T = A \rightarrow u_2^T A u_1 = u_1^T A u_2$

Và $u_2^T u_1 = u_1^T u_2$ do u_1, u_2 là các vecto thực.

$\Rightarrow (\lambda_1 - \lambda_2) u_1^T u_2 = 0$

- Do $\lambda_1 \neq \lambda_2$ nên: $u_1^T u_2 = 0$

$\Rightarrow$ u_1, u_2 trực giao. (đpcm)

Câu 2:

CMR tính xác định dương của A dẫn đến $\lambda_i > 0$

- A là ma trận đối xứng được gọi là xác định dương nếu: $v^T A v > 0$ cho mọi vecto $v \neq 0$. Vì A đối xứng nên:

$$\begin{cases} \text{Mọi giá trị riêng } \lambda_i \text{của A đều là số thực.} \\ \text{Có thể chọn một hệ vecto riêng trực giao } \{u_1, u_2, \dots, u_M\}, \text{trong đó } Au_i = \lambda_i u_i \end{cases}$$

- Giả sử ngược lại: tồn tại một $\lambda_k \leq 0$:
  - Chọn $v = u_k$ (một vecto riêng). Đây là vector khác không (vì vector riêng không thể là vector 0).
  - $v^T A v = u_k^T A u_k = u_k^T(\lambda_k u_k) = \lambda_k(u_k^T u_k)$
    - TH1: $\lambda_k < 0$, thì $\lambda_k(u_k^T u_k) < 0 \Rightarrow v^T A v < 0$

      TH2: $\lambda_k = 0$, thì biểu thức bằng 0.

- Cả hai trường hợp đều mẫu thuẫn với giả thuyết xác định dương $v^T Av > 0$.

- Vậy, tính xác định dương của A dẫn đến $\lambda_i > 0$

CMR nếu $\lambda_i > 0$ thì A là ma trận xác định dương($v^T Av > 0$) .

- Vì A đối xứng, luôn tồn tại ma trận trực giao U (các cột là vector riêng chuẩn trực giao) và ma trận đường chéo P chứa các λ_i, sao cho:

$$A = UPU^T, \text{ với } P = diag(\lambda_1, \lambda_2, \dots, \lambda_m),\ \lambda_i > 0$$

- Với $v \neq 0$, đặt $z = U^T v$ $(z \neq 0)$

$$v^T Av = v^T (UPU^T)v$$
$$= (v^T U)P(U^T v) = z^T Pz$$

- Vì P là ma trận đường chéo với các phần tử $\lambda_i > 0$ nên:

$$z^T Pz = \sum_{i=1}^{M} \lambda_i z_i^2$$

Vì $\lambda_i > 0, z_i \neq 0$ nên $\sum_{i=1}^{M} \lambda_i z_i^2 > 0$

⇨ $z^T Pz > 0$ hay $v^T Av > 0$

- Vậy, $\lambda_i > 0, \forall i \Rightarrow$ A là xác định dương

Kết luận rằng một ma trận dương luôn khả nghịch.

- Nếu A xác định dương, thì mọi $\lambda_i > 0$. Khi đó

$$\det(A) = \lambda_1 . \lambda_2 \dots \lambda_M > 0$$

- Vì $\det(A) \neq 0$, nên A là khả nghịch (tồn tại A^{-1})
- Vậy A xác định dương ⇒ $\det(A) \neq 0$ ⇒ A khả nghịch.

Câu 3:

1. ***Tính entropy ban đầu dựa trên cột Play***

   - <Định nghĩa Entropy và công thức: cái này ko cần chép vào bài thi>
     Entropy là một đại lượng toán học dùng để đo lượng tin không chắc chắn hay lượng ngẫu nhiên của một sự kiện hay phân phối ngẫu nhiên cho trước.

$$\boldsymbol{H(x) = -\sum_{x=1}^{n} p_x \, log_2 \, p_x}$$

n: số lượng giá trị khác nhau của biến đang xét.

p_x: xác suất xuất hiện của giá trị i trong tập dữ liệu.

- Cột Play:
  - Số lượng mẫu: 6, số lượng giá trị khác nhau: 2 (Y/N)
  - Xác suất: $P(Yes) = P(No) = \frac{3}{6} = 0.5$
  - Tính entropy:

$$H(x) = -\sum_{x=1}^{n} p_x \, log_2 \, p_x$$
$$= -[P(Yes) \cdot log_2 \, P(Yes) + P(No) \cdot log_2 \, P(No)]$$
$$= -[0.5 \log(0.5) + 0.5 \log(0.5)] \; = \; 1$$

2. Tính Information Gain (IG) nếu sử dụng weather để chia tập data

- <Đ/n + CT: ko cần chép vào thi>

  Information Gain (IG) là mức giảm kỳ vọng của entropy khi phân chia các mẫu dựa trên một thuộc tính.

$$\boldsymbol{Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)}$$

  - $Gain(S, A)$*:* mức giảm entropy của tập data S khi sử dụng thuộc tính A để chia rập dữ liệu.
  - $Entropy(S)$: entropy ban đầu của tập S
  - $\sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$:
    - Tổng tất cả các giá tị v có thể có của thuộc tính A.
    - $\frac{|S_v|}{|S|}$: tỷ lệ số lượng phần tử trong tập con S_v (phân hoạch dựa trên giá trị v) / tổng số phần tử trong tập S.
    - $Entropy(S_v)$: entropy của tập con S_v (những phần tử trong S có giá trị A = v)

- **Tính IG**
    - Tập data được chia thành 3 giá trị của Weather: sunny, overcast, rainy.
    - Tính entropy từng nhóm:
        - Entropy(Sunny): 2 bản ghi đều là No (đối chiếu với cột target – Play)

$$Entropy(Sunny) = -1 \cdot log_2 1 = 0$$

        - Entropy(Overcast): 1 bản ghi là Yes

$$Entropy(Overcast) = -1 \cdot log_2 1 = 0$$

        - Entropy(Rainy): 2 bản ghi là Yes, 1 No

$$Entropy(Rainy) = -\left[\frac{2}{3} \cdot log_2 \frac{2}{3} + \frac{1}{3} \cdot log_2 \frac{1}{3}\right] \approx 0.9183$$

    - Tính trọng số $\frac{|S_v|}{|S|} Entropy(S_v)$ cho từng nhóm:
        - Sunny: $\frac{|S_{sunny}|}{|S|} Entropy(S_{sunny}) = \frac{2}{6} \cdot 0 = 0$
        - Overcast: $\frac{|S_{overcast}|}{|S|} Entropy(S_{overcast}) = \frac{1}{6} \cdot 0 = 0$
        - Rainny: $\frac{|S_{rainny}|}{|S|} Entropy(S_{rainny}) = \frac{3}{6} \cdot 0.9183 = 0.45915$
    - Tính IG

$$Gain(S, Weather) = Entropy(S) - \sum_{v \in Values(Weather)} \frac{|S_v|}{|S|} Entropy(S_v) = 1 - (0 + 0 + 0.45915) = 0.54085$$

3. Giải thích cách chọn thuật toán tốt nhất dựa trên IG trong thuật toán cây quyết định.

- Mục tiêu của cây quyết định (decision tree):
    - Phân chia dữ liệu sao cho các nhóm kết quả (leaf nodes) trở nên đồng nhất nhất có thể.
    - Thuộc tính được chọn tại mỗi bước là thuộc tính "tốt nhất" để giảm độ không chắc chắn (entropy) của dữ liệu.

- IG là thước đo mức độ giảm **độ không chắc chắn** (entropy) khi chia dữ liệu theo một thuộc tính, giúp tập dữ liệu trở nên đồng nhất hơn.
- Thuộc tính nào có IG cao nhất sẽ được chọn làm thuộc tính tốt nhất để phân chia dữ liệu.
- Ví dụ ở bài này:
  - Tính IG cho cả Weather và Temperature.
  - Nếu$Gain(S, Weather) > Gain(S, Temperature)$,chọn Weather để chia dữ liệu ở bước đầu tiên.
  - Lặp lại tại các bước tiếp theo cho đến khi cây hoàn thiện.

Câu 4:

- Entropy của phân phối đồng thời $H(X,Y)$:

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)\log\big(p(x,y)\big) \quad (1)$$

Trong đó $p(x,y)$là phân phối xác suất đồng thời của X và Y

- Entropy của từng biến:

$$H(X) = -\sum_{x\in X} p(x)\log\big(p(x)\big) \quad (2)$$

$$H(Y) = -\sum_{y\in Y} p(y)\log\big(p(y)\big) \quad (3)$$

- Nếu x và Y độc lập: $p(x,y) = p(x).p(y)$
- Thay vào (1):

$$H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x).p(y)\log\big(p(x).p(y)\big)$$

$$= -\sum_{x\in X}\sum_{y\in Y} p(x).p(y)\big[\log\big(p(x)\big) + \log\big(p(y)\big)\big]$$

$$= -\sum_{x\in X}\sum_{y\in Y} p(x).p(y)\log\big(p(x)\big) - \sum_{x\in X}\sum_{y\in Y} p(x).p(y)\log\big(p(y)\big)$$

$$= A + B$$

- Tính A:

$$-\sum_{x\in X}\sum_{y\in Y} p(x).p(y)\log\big(p(x)\big) = -\sum_{x\in X} p(x)\log\big(p(x)\big)\sum_{y\in Y} p(y)$$

$$= -\sum_{x \in X} p(x) \log\big(p(x)\big) * 1 = H(X)$$

- Tính B <tương tự>

$$-\sum_{x \in X}\sum_{y \in Y} p(x).p(y) \log\big(p(y)\big) = -\sum_{y \in Y} p(y) \log\big(p(y)\big) \sum_{x \in X} p(x)$$

$$= -\sum_{y \in Y} p(y) \log\big(p(y)\big) * 1 = H(Y)$$

⇨ $H(X,Y) = H(X) + H(Y)$

Câu 5:

Chứng minh rằng:

$$D_{KL}(P||Q) \geq 0,$$

Với P và Q là hai phân phối xác suất và dấu "=" xảy ra khi P = Q. KL Divergence được định nghĩa:

$$D_{KL}(P||Q) = \sum_{x} P(x) \, log \frac{P(x)}{Q(x)}$$

Ta có:

$$D_{KL}(P||Q) = \sum_x P(x) \, log \frac{P(x)}{Q(x)} = \sum_x P(x) logP(x) - \sum_x P(x) logQ(x) \qquad (1)$$

Bất đẳng thức Gibbs phát biểu rằng entropy của một phân phối xác suất luôn nhỏ hơn hoặc bằng cross-entropy với một phân phối xác suất khác, và dấu "=" chỉ xảy ra khi hai phân phối là giống hệt nhau. Do đó:

$$-\sum_{i=1}^{n} P(x_i) logP(x_i) \leq -\sum_{i=1}^{n} P(x_i) logQ(x_i)$$

$$\Leftrightarrow \quad \sum_{i=1}^{n} P(x_i) logP(x_i) - \sum_{i=1}^{n} P(x_i) logQ(x_i) \geq 0 \qquad (2)$$

Từ (1) và (2) suy ra, $D_{KL}(P||Q) \geq 0$ và dấu "=" xảy ra khi và chỉ khi P = Q

Câu 6:

Cho hai phân phối Bernoulli P và Q với xác suất p và q. Chứng minh rằng:

$$D_{KL}(P||Q) = p.log\frac{p}{q} + (1 - p).log\frac{1 - p}{1 - q}$$

Ta có độ phân kỳ KL giữa 2 phân phối $P(x)$ và $Q(x)$ được định nghĩa như sau:

$$D_{KL}(P||Q) = -\int P(x)\, ln\frac{Q(x)}{P(x)}dx$$

Đối với một biến ngẫu nhiên rời rạc, công thức tổng quát sẽ trở thành:

$$D_{KL}(P||Q) = -\sum_{x} P(x)\, log\frac{Q(x)}{P(x)} \quad (*)$$

Với phân phối Bernoulli P có xác suất thành công là p:

- $P(x) = p$ khi $x = 1$
- $P(x) = 1 - p$ khi $x = 0$

Tương tự, phân phối Bernoulli Q có xác suất thành công là q:

- $Q(x) = q$ khi $x = 1$
- $Q(x) = 1 - q$ khi $x = 0$

Do đó (*) trở thành:

$$D_{KL}(P||Q) = -\left[P(1).log\frac{Q(1)}{P(1)} + P(0).log\frac{Q(0)}{P(0)}\right]$$

$$= -\left[p.log\frac{q}{p} + (1-p).log\frac{1-q}{1-p}\right]$$

$$= \boldsymbol{p.log\frac{p}{q} + (1-p).log\frac{1-p}{1-q}} \qquad (\text{đ}\boldsymbol{pcm})$$

Câu 7: Chứng minh rằng:

$$\boldsymbol{H(X,Y) = H(X) + H(Y|X),}$$

Và sử dụng điều này để suy ra:

$$I(X;Y) = H(X) + H(Y) - H(X,Y)$$

Trong đó $I(X:Y)$ là Mutual Information.

Entropy kết hợp của hai biến ngẫu nhiên:

$$H(X,Y) = -\sum_{x,y} P(x,y)\log P(x,y)$$

Ta có: $P(x,y) = P(x).P(y|x)$, suy ra:

$$H(X,Y) = -\sum_{x,y} P(x,y)\log[P(x).P(y|x)]$$

$$= -\sum_{x,y} P(x,y)\log P(x) - \sum_{x,y} P(x,y)\log P(y|x)$$

$$= -\sum_{x} P(x)\log P(x) - \sum_{x} P(x)\sum_{y} P(x,y)\log P(y|x)$$

$$= H(X) + H(Y|X)$$

$$\Rightarrow H(X,Y) = H(X) + H(Y|X)$$

Mutual Information: $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$

$I(X;Y) = H(Y) - H(Y|X) = H(Y) - [H(X,Y) - H(X)] = H(X) + H(Y) - H(X,Y)$

Vậy bài toán đã được chứng minh

Câu 8: Chứng minh rằng phân phối P (biến ngẫu nhiên liên tục) có entropy lớn nhất trong tất cả các phân phối với kỳ vọng $\mathrm{E}[X] = \mu$ là phân phối đồng đều trên tập hợp các giá trị của X.

Tính entropy H(X) của một biến ngẫu nhiên liên tục X:

$$H(X) = -\int_{-\infty}^{\infty} f(x)\log f\,(x)\,dx$$

Ràng buộc:

1. $\mathrm{E}[\mathrm{X}] = \mu$:

$$E[X] = \int_{-\infty}^{\infty} x\,f(x)\,dx = \mu$$

$$\Rightarrow \int_{-\infty}^{\infty} x\,f(x)\,dx - \mu = 0$$

2. $\int_{-\infty}^{\infty} f(x)dx = 1$: Hàm mật độ xác suất f(x) phải chuẩn hoá

   $\Rightarrow \int_{-\infty}^{\infty} f(x)dx - 1 = 0$

Hàm Lagrange:

$\mathcal{L}(f,\lambda_1,\lambda_2) = -\int_{-\infty}^{\infty} f(x)\log f\,(x)\,dx + \lambda_1\left(\int_{-\infty}^{\infty} f(x)dx - 1\right) + \lambda_2\left(\int_{-\infty}^{\infty} x\,f(x)\,dx - \mu\right)$

Đạo hàm theo f(x):

$$\frac{\partial \mathcal{L}}{\partial \mathrm{f(x)}} = -log f(\mathrm{x}) - 1 + \lambda_1 + \lambda_2 x$$

Cho đạo hàm bằng 0:

$$-log f(\mathrm{x}) - 1 + \lambda_1 + \lambda_2 x = 0$$

$$\Rightarrow log f(\mathrm{x}) = \lambda_2 \mathrm{x} + \lambda_1 - 1$$

⇨ $f(\mathrm{x}) = e^{\lambda_2 \mathrm{x} + \lambda_1 - 1}$

Ta có ràng buộc: $\int_{-\infty}^{\infty} f(x)dx = 1$

⇨ $\int_{-\infty}^{\infty} \left(e^{\lambda_2 \mathrm{x} + \lambda_1 - 1}\right)dx = 1$

Nếu $\lambda_2 \neq 0$, $e^{\lambda_2 x}$ sẽ tăng đến vô hạn khi $\mathrm{x} \to \infty$ hoặc $\mathrm{x} \to -\infty$, khiến tích phân hội tụ. Thế nên bắt buộc $\lambda_2 = 0$.

⇨ $f(\mathrm{x}) = e^{\lambda_1 - 1}$

Đây là một hằng số trên toàn bộ trục số thực, nhưng để f(x) là hàm mật độ xác suất hợp lệ, nó chỉ có thể khác 0 trên một khoảng hữu hạn **[a,b]** và bằng $\frac{1}{b-a}$ để thỏa mãn chuẩn hóa xác suất:

$$\int_a^{\mathrm{b}} f(x)dx = 1$$

Khi đó f(x) có dạng:

$$f(x) = \begin{cases} \dfrac{1}{b - a}, nếu\ x \in [a, b] \\ 0, nếu\ x \notin [a, b] \end{cases}$$

Đây là phân phối đồng đều (Uniform).

Vậy: Phân phối đồng đều có entropy lớn nhất trong các phân phối biến ngẫu nhiên liên tục với kỳ vọng $\mathrm{E}[X] = \mu$.

Entropy lúc này có giá trị (làm thêm chứ không có trong phần chứng minh):

$$H(X) = -\int_a^{\mathrm{b}} f(x) \log f\,(x)\, dx$$

$$⇨\ H(X) = -\int_a^{\mathrm{b}} f(x) \log f\,(x)\, dx$$

$$⇨\ H(X) = -\int_a^{\mathrm{b}} \frac{1}{b-a} \log\left(\frac{1}{b-a}\right) dx$$

$$⇨\ H(X) = -\frac{1}{b-a}.\log\left(\frac{1}{b-a}\right).(\mathrm{b} - \mathrm{a})$$

$$⇨\ H(X) = -\log\left(\frac{1}{b-a}\right)$$

$$⇨\ H(X) = \log(b - a)$$

Câu 9: Chứng minh rằng: **H(X|Y) ≤ H(X), với dấu "=" xảy ra khi X và Y độc lập.**

❖ Chứng minh: H(X|Y) ≤ H(X)

Ta có: H(X|Y) = H(X, Y) – H(Y) (1)

- Mà $H(X, Y) \leq H(X)$ vì $H(X, Y)$ chứa cả thông tin về X và Y, trong khi $H(X)$ chỉ chứa thông tin về X.

⇨ $H(X|Y) = H(X, Y) - H(Y) \leq H(X) - H(Y)$

- Mà $H(Y) \geq 0$ vì tính không âm của entropy

⇨ $H(X) \geq H(X) - H(Y)$

⇨ $H(X|Y) \leq H(X)$ (đpcm)

➢ Dấu "=" xảy ra khi X và Y độc lập:

Với X, Y độc lập ta có: $H(X, Y) = H(X) + H(Y)$

Thế vào (1): $H(X|Y) = H(X) + H(Y) - H(Y) = H(X)$

Vậy: **$H(X|Y) \leq H(X)$ với dấu "=" xảy ra khi X, Y độc lập**

Câu 10:

Cho mô hình hồi quy tuyến tính đa biến:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon,$$

trong đó:

- $\mathbf{y} \in \mathbb{R}^n$: vector quan sát (biến phụ thuộc),
- $\mathbf{X} \in \mathbb{R}^{n \times p}$: ma trận đặc trưng (biến độc lập),
- $\mathbf{w} \in \mathbb{R}^p$: vector trọng số cần ước lượng,
- $\epsilon \in \mathbb{R}^n$: vector nhiễu (giả định $\epsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$).

Với hàm mất mát (loss function) dựa trên sai số bình phương (Mean Squared Error - MSE):

$$L(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 = (\mathbf{y} - \mathbf{X}\mathbf{w})^\top(\mathbf{y} - \mathbf{X}\mathbf{w}).$$

Hãy tìm công thức nghiệm đóng (closed-form solution) $\mathbf{w}^*$ bằng cách tối thiểu hóa hàm mất mát $L(\mathbf{w})$.

Tìm w* bằng cách tối thiểu hoá hàm mất mát L(w):

$$w^* = \arg\min_w L(w)$$

Hàm mất mát L(w) là hàm lồi nên chỉ có một cực trị duy nhất và chính là điểm cực tiểu toàn cục.

Ta có: $L(w) = (y - Xw)^T(y - Xw)$

- Khai triển:

  Ta có công thức nhân tích vô hướng:

  $(A - B)^T(A - B) = A^TA - 2A^TB + B^TB$

  ⇨ $(y - Xw)^T(y - Xw) = y^Ty - 2y^TXw + (Xw)^TXw$

  ⇨ $(y - Xw)^T(y - Xw) = y^Ty - 2y^TXw + w^TX^TXw$

- Lấy đạo hàm: $\frac{\partial L(w)}{\partial w}$

  $\frac{\partial}{\partial w}(y^Ty) = 0$

  $\frac{\partial}{\partial w}(-2y^TXw) = -2X^Ty$ (Ta có công thức: $\frac{\partial}{\partial w}(A^Tw) = A$, $A^T = y^TX$)

  $\frac{\partial}{\partial w}(w^TX^TXw) = 2X^TXw$ (Ta có công thức: $\frac{\partial}{\partial w}(w^TAw) = 2Aw$ với A là ma trận vuông đối xứng, $A = X^TX$ đối xứng vì $(X^TX)^T = X^TX$)

  ⇨ $\frac{\partial L(w)}{\partial w} = -\mathbf{2X^Ty} + \mathbf{2X^TXw}$

  Đặt đạo hàm bằng 0: $-2X^Ty + 2X^TXw = 0$

- Giải tìm w*:

  $X^TXw = X^Ty$

  ⇨ $\mathbf{w} *= (\mathbf{X^TX})^{-1}\mathbf{X^Ty}$

Câu 11:

- Xét bài toán

$$\max_{\mathbf{u}} \mathbf{u}^\top \mathbf{C}\mathbf{u}, \quad \text{với ràng buộc } \|\mathbf{u}\| = 1.$$

  Chứng minh rằng nghiệm của bài toán này chính là vector riêng tương ứng với giá trị riêng lớn nhất của ma trận $\mathbf{C}$.

- Bài toán này liên quan gì đến PCA?

❖ Chứng minh:

Ta có: Hàm mục tiêu: $\max_u u^TCu$

Ràng buộc: $\|u\| = 1$ $(u^Tu = 1) \rightarrow u^Tu - 1 = 0$

Hàm Lagrange được xây dựng như sau:

$\mathcal{L}(u, \lambda)$ = Hàm mục tiêu $- \lambda \cdot$ (Ràng buộc)

$\mathcal{L}(u, \lambda) = u^TCu - \lambda \cdot (u^Tu - 1)$

Đạo hàm theo u:

- Ta có công thức: $\frac{\partial(\mathrm{u}^{\mathrm{T}}Cu)}{\partial \mathrm{u}} = 2Cu$, với C là ma trận vuông đối xứng
- $\frac{\partial(\lambda \mathrm{u}^{\mathrm{T}}u)}{\partial \mathrm{u}} = 2\lambda u$
- $\frac{\partial \lambda}{\partial \mathrm{u}} = 0$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \mathbf{2Cu} - \mathbf{2\lambda u}$$

Đặt đạo hàm bằng 0 để tìm nghiệm:

$$2\mathrm{Cu} - 2\lambda \mathrm{u} = 0$$
$$\mathrm{Cu} = \lambda \mathrm{u}$$

Phương trình $\mathbf{Cu} = \boldsymbol{\lambda}\mathbf{u}$ là phương trình vector riêng với:

- u là vector riêng của ma trận C
- λ là trị riêng tương ứng

Để tối đa hoá $\mathrm{u}^{\mathrm{T}}\mathrm{Cu}$, giá trị λ phải lớn nhất. Thế nên, nghiệm của bài toán này chính là vector riêng v_1 tương ứng với giá trị riêng lớn nhất λ_1 của ma trận C.

❖ Bài toán này liên quan đến PCA:

C thường là **ma trận hiệp phương sai** của dữ liệu.

Vector riêng v_1 tương ứng với giá trị riêng lớn nhất λ_1:

- v_1 xác định **thành phần chính đầu tiên**, là hướng trong không gian đặc trưng chứa nhiều phương sai nhất của dữ liệu (tối đa hóa phương sai).
- Giá trị riêng λ_1 đại diện cho lượng phương sai dữ liệu được giải thích bởi thành phần chính đầu tiên.

Câu 12:

1. **Mô tả ý tưởng chính của Boosting.**

   Boosting là một phương pháp học máy nhằm kết hợp nhiều mô hình yếu (weak learners) thành một mô hình mạnh hơn (strong learner). Ý tưởng chính của Boosting:

   - Các mô hình con được xây dựng tuần tự.
   - Mỗi mô hình mới cố gắng sửa lỗi của mô hình trước đó.
   - Trọng số của các mẫu dữ liệu được điều chỉnh để tập trung hơn vào các mẫu khó phân loại chính xác.

- Kết hợp dự đoán của tất cả các mô hình con bằng cách gán trọng số cho chúng, tạo ra một dự đoán cuối cùng.

2. Viết công thức để cập nhật trọng số của các mẫu sau mỗi lần huấn luyện một mô hình con của thuật toán Adaboost (phân loại nhị phân).

Khởi tạo trọng số ban đầu:

Gán trọng số ban đầu cho mỗi mẫu dữ liệu:

$$w_n^{(1)} = \frac{1}{N}, \qquad n = 1, \dots, N$$

Lặp qua từng mô hình con $m = 1, \dots, M$:

a. Huấn luyện mô hình con $y_m(x)$

- Huấn luyện mô hình $y_m(x)$ trên tập dữ liệu với trọng số $w_n^{(m)}$, bằng cách tối thiểu hóa hàm lỗi:

$$\boldsymbol{J_m = \sum_{n=1}^{N} w_n^{(m)} . 1(y_m(x_n) \neq t_n)}$$

Trong đó $\mathbf{1(.)}$ là hàm chỉ thị, trả về 1 nếu đúng, 0 nếu sai.

b. Tính lỗi mô hình:

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^{(m)} . 1(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^{(m)}}$$

c. Tính trọng số của mô hình α_m

$$\alpha_m = \ln\left(\frac{1 - \epsilon_m}{\epsilon_m}\right)$$

d. Cập nhật trọng số của mẫu dữ liệu:

$$w_n^{(m+1)} = w_n^{(m)} . e^{\alpha_m . 1(y_m(x_n) \neq t_n)}$$

Kết hợp các mô hình con để đưa ra dự đoán cuối cùng:

$$Y_M(x) = sign\left(\sum_{m=1}^{M} \alpha_m . y_m(x)\right)$$

Với

$$\text{sign}(x) = \begin{cases} 1, & \text{nếu } x > 0, \\ 0, & \text{nếu } x = 0, \\ -1, & \text{nếu } x < 0. \end{cases}$$

3. Trình bày thuật toán Gradient Boosting, và giải thích.

Gradient Boosting là một kỹ thuật học máy mạnh mẽ để xây dựng các mô hình dự đoán bằng cách kết hợp các mô hình con (weak learners), thường là các cây quyết định, theo cách tuần tự. Mỗi mô hình con được huấn luyện để giảm gradient của hàm mất mát tại bước hiện tại.

Thuật toán:

1. Khởi tạo mô hình ban đầu:
   - Khởi tạo mô hình $F_0(x)$ bằng cách tối thiểu hóa hàm mất mát tổng thể $L(y, F(x))$:

$$F_0(x) = arg\min_c \sum_{i=1}^{N} L(y_i, c)$$

   Trong đó:

- $L(y, F(x))$: Hàm mất mát (ví dụ: MSE, cross-entropy, ...).
- c: Hằng số tối ưu hóa

   Ví dụ: Trong bài toán hồi quy, $F_0(x)$ là giá trị trung bình của y:

$$F_0(x) = \frac{1}{N}\sum_{i=1}^{N} y$$

2. Lặp từ 1 đến T:
   a. Tính sai số (gradient âm):

$$r_i{}^{(t)} = -\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$$

   b. Xây dựng mô hình con $h_t(x)$ (thường là một cây quyết định nhỏ) để dự đoán sai số $r_i{}^{(t)}$:

$$h_t(x) \approx r_i{}^{(t)}$$

   c. Tính hệ số tối ưu γ_t:

$$\gamma_t = arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{t-1}(x_i) + \gamma h_t(x_i))$$

   d. Cập nhật mô hình:

$$F_t(x) = F_{t-1}(x) + \gamma_t . h_t(x)$$

3. Mô hình cuối cùng: Sau T bước lặp thì mô hình cuối cùng là tổng của tất cả các mô hình con:

$$F_T(x) = F_0(x) + \sum_{t=1}^{T} \gamma_t . h_t(x)$$

Giải thích lý thuyết:

- Gradient Boosting sử dụng gradient để tìm hướng tốt nhất giảm giá trị của hàm mất mát. Gradient âm chính là hướng tối ưu để giảm giá trị của $L(y, F(x))$.
- Hàm mất mát linh hoạt: Gradient Boosting có thể sử dụng nhiều loại hàm mất mát, chẳng hạn:
    - Hồi quy: Sử dụng MSE
    - Phân loại: Sử dụng làm log-likelihood
- Tóm lại: Gradient Boosting là một thuật toán tổng quát và mạnh mẽ, hoạt động qua:
    - **Tính gradient của hàm mất mát** để xác định hướng tối ưu hóa.
    - **Xây dựng các mô hình con tuần tự** để giảm lỗi còn sót lại.
    - **Kết hợp các mô hình con** để tạo ra mô hình cuối cùng.

Câu 13:

1. Trình bày thuật toán Random Forest.

Random Forest là một thuật toán machine learning dựa trên việc kết hợp nhiều cây quyết định (Decision Trees) để tạo ra một mô hình dự đoán mạnh mẽ hơn, sử dụng kỹ thuật "Bagging" và "Random Feature Selection".

Các bước thuật toán:

1. Tạo các tập con (Boostrap Sampling)
    - Từ tập dữ liệu ban đầu có N mẫu, thực hiện lấy mẫu ngẫu nhiên ***có hoàn lại*** để tạo ra B tập dữ liệu con.
    - Mỗi tập dữ liệu con có kích thước bằng kích thước tập ban đầu (nhưng có thể chứa các mẫu lặp lại).
2. Xây dựng các cây quyết định (Decision Trees):
    - Với mỗi tập dữ liệu con:
        - Tạo một cây quyết định.
        - Tại mỗi nút, không sử dụng toàn bộ các đặc trưng d, mà chỉ sử dụng một tập con k đặc trưng được chọn ngẫu nhiên ($k < d$).

- Cây được xây dựng đến khi đạt điều kiện dừng (độ sâu tối đa hoặc không còn mẫu nào để chia)

3. Dự đoán với Random Forest:
   - Phân loại: Sử dụng **đa số phiếu bầu** từ tất cả các cây.

$$\hat{y} = mode\{T_1(x), T_2(x), \dots, T_B(x)\}$$

   - Hồi quy: Tính **trung bình dự đoán** từ tất cả các cây.

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Ý tưởng chính:

- Bagging: Tăng tính đa dạng giữa các cây bằng cách sử dụng các tập dữ liệu khác nhau.
- Random Feature Selection: Mỗi cây sử dụng một tập con ngẫu nhiên các đặc trưng để giảm tương quan giữa các cây.

2. Sự khác nhau giữa Random Forest và Bagging

Tiêu chí	**Bagging**	**Random Forest**
Cách xây dựng cây	Sử dụng toàn bộ đặc trưng tại mỗi nút khi xây dựng cây quyết định.	Tại mỗi nút, chỉ sử dụng một tập con ngẫu nhiên các đặc trưng (Random Feature Selection).
Mục tiêu chính	Tăng tính ổn định và giảm phương sai của các mô hình (variance reduction).	Giảm tương quan giữa các cây và tăng tính đa dạng, ngoài việc giảm phương sai.
Số lượng mô hình con	Tạo B cây quyết định từ các tập dữ liệu con.	Tạo B cây quyết định từ các tập dữ liệu con và với các đặc trưng ngẫu nhiên ở từng nút.
Hiệu quả tổng thể	Tốt cho các bài toán mà tất cả các đặc trưng đều có tầm quan trọng tương đương.	Tốt hơn Bagging khi có các đặc trưng mạnh và yếu, nhờ việc chọn lọc đặc trưng ngẫu nhiên.

Ứng dụng phổ biến	Chủ yếu dùng với cây quyết định đơn giản (Decision Trees) hoặc các mô hình khác (Bagging SVM).	Được thiết kế riêng cho cây quyết định và trở thành thuật toán riêng biệt.

Câu 14:

Cho phân phối Gaussian nhiều chiều $X = (X_1, X_2, \ldots, X_n)$ với kỳ vọng $\mathbb{E}[X] = \mu$ và ma trận hiệp phương sai $\Sigma = Cov(X)$. Giả sử hai thành phần X_i và X_j của X có hiệp phương sai $Cov(X_i, X_j) = 0$, tức là:

$$Cov(X_i, X_j) = \mathbb{E}[(X_i - \mathbb{E}[X_i])(X_j - \mathbb{E}[X_j])] = 0$$

Chứng minh rằng X_i và X_j là độc lập.

Ta có $Cov(X_i, X_j) = \mathbb{E}[\,(X_i - \mathbb{E}[X_i])\,(X_j - \mathbb{E}[X_j])\,]$

$$= \mathbb{E}[\,X_i X_j - X_i\,\mathbb{E}[X_j] - X_j\,\mathbb{E}[X_i] + \mathbb{E}[X_i]\,\mathbb{E}[X_j]\,]$$

$$= \mathbb{E}[\,X_i X_j] - \mathbb{E}[\,X_i\,\mathbb{E}[X_j]\,] - \mathbb{E}[X_j\,\mathbb{E}[X_i]] + \mathbb{E}[\mathbb{E}[X_i]\,\mathbb{E}[X_j]]$$

$$= \mathbb{E}[\,X_i X_j] - \mathbb{E}[X_i]\,\mathbb{E}[X_j] - \mathbb{E}[X_j]\,\mathbb{E}[X_i] + \mathbb{E}[X_i]\,\mathbb{E}[X_j]$$

$$= \mathbb{E}[\,X_i X_j] - \mathbb{E}[X_i]\,\mathbb{E}[X_j]$$

Mà $Cov(X_i, X_j) = 0 \Rightarrow \mathbb{E}[X_i X_j] - \mathbb{E}[X_i]\mathbb{E}[X_j] = 0$

$$\Rightarrow \mathbb{E}[X_i X_j] = \mathbb{E}[X_i]\mathbb{E}[X_j]$$

$\Rightarrow X_i, X_\text{j}$ độc lập

Câu 15: Giải thích ý tưởng cơ bản của t-sne và random projection.

1. **t-SNE**

   Khái niệm: Là thuật toán giảm chiều dữ liệu phi tuyến tính, dùng để khám phá và trực quan hóa dữ liệu trong không gian 2D hoặc 3D.

   Mục tiêu: Bảo toàn mối quan hệ tương đồng giữa các điểm dữ liệu khi chuyển từ không gian cao chiều sang thấp chiều.

   Cách hoạt động:

   - t-SNE tính toán độ tương đồng cặp giữa các điểm dữ liệu trong không gian cao chiều bằng cách sử dụng phân phối Gaussian. Các điểm gần nhau có xác suất cao hơn để được chọn làm hàng xóm.

- Ánh xạ các điểm dữ liệu từ không gian cao chiều sang không gian thấp chiều sao cho mối tương đồng giữa các cặp điểm được bảo toàn một cách tốt nhất.
- Tối ưu hóa bằng gradient descent để giảm thiểu sự khác biệt giữa phân phối xác suất ở hai không gian.

Kết quả: Dữ liệu trong không gian thấp chiều được tổ chức thành các cụm, phản ánh cấu trúc và mối quan hệ của không gian gốc.

2. Random Projection

Khái niệm: Là phương pháp giảm chiều dữ liệu tuyến tính, sử dụng phép chiếu ngẫu nhiên để giảm số chiều dữ liệu.

Mục tiêu: Bảo toàn khoảng cách tương đối giữa các điểm dữ liệu trong không gian thấp chiều.

Cách hoạt động:

- Sử dụng một ma trận chiếu ngẫu nhiên với các giá trị được lấy từ một phân phối xác suất, chẳng hạn phân phối Gaussian. Ma trận này được sử dụng để biến đổi dữ liệu từ không gian cao chiều sang không gian thấp chiều bằng cách nhân ma trận ngẫu nhiên với ma trận dữ liệu gốc.
- Áp dụng lý thuyết Johnson-Lindenstrauss để bảo toàn gần đúng khoảng cách Euclidean giữa các điểm.

Kết quả: Dữ liệu trong không gian thấp chiều duy trì cấu trúc tương đối, giảm số chiều và tăng hiệu quả tính toán.

Câu 16: Giải thích ý tưởng cơ bản của Importance Sampling. Làm sao mà Importance Sampling hữu dụng trong việc ước lượng xác suất của các biến cố hiếm.

1. Ý tưởng cơ bản của Importance Sampling

Khái niệm: Importance Sampling là một kỹ thuật mô phỏng Monte Carlo, được sử dụng để ước lượng kỳ vọng của một hàm số khi phân phối xác suất của biến ngẫu nhiên không dễ lấy mẫu.

Cách hoạt động:

- Ước lượng kỳ vọng của một hàm $f(x)$ dưới phân phối xác suất $p(x)$:

$$\mathbb{E}_p[f(x)] = \int f(x)p(x)dx$$

Tuy nhiên, thay vì lấy mẫu từ phân phối gốc $p(x)$, Importance Sampling sử dụng một phân phối thay thế $q(x)$ (gọi là *proposal distribution*), dễ lấy mẫu hơn.

$$\mathbb{E}_p[f(x)] = \int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx$$

Hệ số trọng số $w(x) = \frac{p(x)}{q(x)}$ được sử dụng để điều chỉnh chênh lệch giữa $p(x)$ và $q(x)$.

- Thực hiện lấy N mẫu $x_1, x_2, \dots, x_N$ từ $q(x)$. Với mỗi mẫu x_i tính trọng số $w(x_i) = \frac{p(x_i)}{q(x_i)}$
- Kỳ vọng của hàm $f(x)$ được ước lượng bằng trung bình trọng số:

$$\mathbb{E}_p[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i)w(x_i)$$

2. Importance Sampling hữu dụng trong ước lượng xác suất của các biến cố hiếm như thế nào?

Biến cố hiếm thường có xác suất rất nhỏ, khiến cho việc lấy mẫu từ phân phối gốc $p(x)$ để ước lượng xác suất trở nên kém hiệu quả (vì rất ít mẫu rơi vào vùng quan tâm).

Importance Sampling giải quyết vấn đề này bằng cách:

- *Tăng cường mẫu trong vùng quan trọng:* Phân phối $q(x)$ được thiết kế để ưu tiên các vùng có khả năng xảy ra biến cố hiếm, từ đó tăng tần suất lấy mẫu tại những vùng này.
- *Điều chỉnh trọng số:* Hệ số trọng số $w(x)$ đảm bảo rằng các mẫu được điều chỉnh để phù hợp với phân phối gốc $p(x)$, dù các mẫu được lấy từ $q(x)$.
- *Cải thiện hiệu quả:* Giảm số lượng mẫu cần thiết để ước lượng chính xác xác suất của các biến cố hiếm so với phương pháp Monte Carlo truyền thống.

Ví dụ:
Ước lượng xác suất $P(X > 5)$ với $X \sim \mathcal{N}(0, 1)$:

1. **Vấn đề:** Phân phối chuẩn $\mathcal{N}(0, 1)$ có xác suất rất nhỏ ở vùng $X > 5$.
2. **Giải pháp với Importance Sampling:**
   - Chọn $q(x) = \mathcal{N}(5, 1)$, tập trung hơn ở vùng $X > 5$.
   - Lấy mẫu $x_1, x_2, \ldots, x_N$ từ $q(x)$.
   - Tính trọng số $w(x_i) = \frac{p(x_i)}{q(x_i)}$, với $p(x) \sim \mathcal{N}(0, 1)$ và $q(x) \sim \mathcal{N}(5, 1)$.
   - Xác suất $P(X > 5) \approx \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}(x_i > 5) w(x_i)$.

Kết quả: Thu thập được nhiều mẫu ở vùng $X > 5$, giúp ước lượng chính xác và hiệu quả hơn.

Câu 17: Trình bày về ảnh hưởng của curse of dimensionality đến các mô hình ML.

❖ **Khái niệm Curse of Dimensionality:**

Lời nguyền chiều dữ liệu là một hiện tượng xảy ra khi số lượng chiều (features) của dữ liệu tăng theo cấp số nhân, dẫn đến việc phân tích, xử lý và huấn luyện mô hình máy học trở nên phức tạp và ít hiệu quả hơn. Điều này ảnh hưởng sâu sắc đến các mô hình ML, đặc biệt là trong học có giám sát, học không giám sát, và học sâu.

❖ **Ảnh hưởng Curse of Dimensionality đến các mô hình máy học:**

1. Sự thưa thớt của dữ liệu:

- Khi số chiều tăng lên, không gian trở nên rất lớn và lượng dữ liệu cần thiết để phủ kín không gian tăng lên theo cấp số nhân, dẫn đến tình trạng "thưa thớt" của các điểm dữ liệu.
- Có thể gây khó khăn cho các thuật toán trong việc nhận diện mẫu và quan hệ giữa các điểm dữ liệu. Đặc biệt trong các thuật toán phân cụm(như KNN hoặc K-Means)trở nên kém hiệu quả vì khoảng cách giữa các điểm trong không gian cao chiều có xu hướng trở nên đồng đều.

2. Tăng cường tính toán:

Khi số chiều càng nhiều, thời gian huấn luyện lâu hơn, khối lượng tính toán lớn và yêu cầu về bộ nhớ sẽ gây khó khăn và tiêu hao nhiều tài nguyên.

3. Overfitting:

Với các chiều cao hơn, mô hình có xu hướng khớp với nhiễu (noise) thay vì mẫu cơ bản. Các mô hình có thể dễ dàng "học thuộc" dữ liệu huấn luyện mà không thực sự học được các mối quan hệ khái quát. Điều này làm giảm khả năng khái quát của mô hình đối với dữ liệu mới. Mô hình có thể ghi nhớ toàn bộ dữ liệu nhưng lại hoạt động kém khi áp dụng vào dữ liệu chưa từng thấy (overfitting). Cách để giảm thiểu là sử dụng PCA hoặc Regularization.

4. Hiện tượng "Concentration of Measure":

Hiện tượng này xảy ra khi phần lớn các điểm dữ liệu có xu hướng trở nên cách xa trung tâm và tập trung gần biên của không gian nhiều chiều khi số chiều tăng lên. Điều này khiến các mô hình bị sai lệch vì chúng sẽ có xu hướng phân biệt dữ liệu ở biên thay vì ở trung tâm, làm giảm độ chính xác và hiệu quả.

5. Mất ý nghĩa khoảng cách:

Trong không gian cao chiều, các khái niệm như mật độ điểm dữ liệu hoặc khoảng cách giữa các điểm gần như tương đương vì không gian bị phân tán quá mức. Do đó, các thuật toán phụ thuộc vào khoảng cách như KNN, Euclid sẽ mất đi ý nghĩa và giảm độ chính xác trong việc đánh giá sự khác biệt giữa các điểm dữ liệu.

6. Khó khăn trong visualization:

- Việc trực quan hóa dữ liệu nhiều chiều trở nên cực kì phức tạp. Chúng ta có thể hình dung tốt trong không gian 2 hoặc 3 chiều, trong khi dữ liệu máy học có hàng chục, hàng trăm chiều.
- Điều này sẽ gây khó khăn trong việc khám phá, phân tích, làm mất khả năng phát hiện trực quan các mẫu và xu hướng tiềm ẩn.