Homework #1

Tuesday, 24 September 2024 18:24

1/

Define: $P(C) = 1/800$; $P(\bar{C}) = 799/800$
having condition / not having

$P(T|C) = 0,98$; $P(T|\bar{C}) = 0,03$
true positive / false positive

a/ $P(T) = P(T|C) \cdot P(C) + P(T|\bar{C}) \cdot P(\bar{C})$
$= 0,98 \cdot 1/800 + 0,03 \cdot 799/800$
$= 0,0312$

b/ $P(C|T) = \dfrac{P(T|C) \cdot P(C)}{P(T)} = 0,0393$

c/ $P(\bar{T}|\bar{C}) = 0,97$

$P(\bar{C}|\bar{T}) = \dfrac{P(\bar{T}|\bar{C}) \cdot P(\bar{C})}{P(\bar{T})} = 0,9999$

2/

a/ $L(u,z,b) = \sum_{i=1}^{3} \| \tilde{x}^i - x^i \|^2$
$= \sum_{i=1}^{3} \| z_{i1} u_1 + b_2 u_2 - x^i \|^2$
$= \sum_{i=1}^{3} z_{i1}^2 + b_2^2 - 2 z_{i1} u_1^T x^i - 2 b_2 u_2^T x^i + x^{iT} x^i$

+) $\dfrac{\partial}{\partial z_{i1}} L(u,z,b) = \sum_{i=1}^{3} 2 z_{i1} - 2 u_1^T x^i = 0$
$\Rightarrow z_{i1}^* = u_1^T x^i$

+) $\dfrac{\partial}{\partial b_2} L(u,z,b) = \sum_{i=1}^{3} 2 b_2 - 2 u_2^T x^i = 0$
$\Rightarrow b_2^* = \dfrac{1}{3} u_2^T \sum_{i=1}^{3} x^i$

b/ $L(u, z^*, b^*) = \sum_{i=1}^{3} \| (u_1^T x^i) u_1 + \underbrace{\tfrac{1}{3} u_2^T \left(\sum_{i=1}^{3} x^i\right) u_2}_{0} - x^i \|^2$

$= \sum_{i=1}^{3} \| (u_1^T x^i) u_1 - (u_1^T x_i) u_1 - (u_2^T x_i) u_2 \|^2 \quad \left(\sum_{i=1}^{3} x^i = 0\right)$

$= \sum_{i=1}^{3} \| (u_2^T x_i) u_2 \|^2 = \sum_{i=1}^{3} [(u_2^T x_i) u_2]^T [(u_2^T x_i) u_2]$

$= \sum_{i=1}^{3} u_2^T x_i x_i^T u_2 = u_2^T S u_2 \Rightarrow$ QED

c/ We have $S = S^T$ (Symmetric) and $S = \sum_{i=1}^{3} x^i x^{iT}$

$\Rightarrow \dfrac{\partial \mathcal{L}}{\partial u_2} = 2 S u_2 - 2 \lambda u_2 = 0$

$\Leftrightarrow S u_2 = \lambda u_2 \Rightarrow \begin{cases} u_2 \text{ is eigen vector} \\ \lambda \text{ is eigen value} \end{cases}$ of S

$\Rightarrow \det(S - \lambda I) = 0$

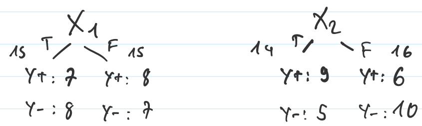
$\Leftrightarrow \lambda^2 - 5\lambda + 3 = 0 \Leftrightarrow \begin{cases} \lambda \approx 0,697 \\ \lambda = 4,303 \end{cases}$

$\Rightarrow u_2 \in \left\{ \begin{bmatrix} \pm 0,570 \\ \pm 0,822 \end{bmatrix} \right\}$

3/

a/ $H(Y) = -(0{,}5 \cdot \log_2 0{,}5 + 0{,}5 \cdot \log_2 0{,}5)$
$= 1$

b/

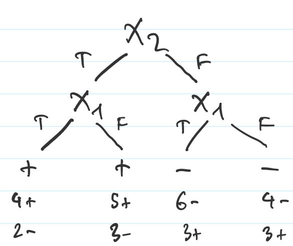


$$H(Y|X_1) = -\frac{15}{30}\left(\frac{7}{15}\log_2\frac{7}{15} + \frac{8}{15}\log_2\frac{8}{15}\right) - \frac{15}{30}\left(\frac{8}{15}\log_2\frac{8}{15} + \frac{7}{15}\log_2\frac{7}{15}\right)$$
$$= 0{,}99679$$

$$H(Y|X_2) = -\frac{14}{30}\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) - \frac{16}{30}\left(\frac{6}{16}\log_2\frac{6}{16} + \frac{10}{16}\log_2\frac{10}{16}\right)$$
$$= 0{,}94783$$

$\Rightarrow IG(X_1) = 1 - 0{,}99679 = 0{,}00321$
$IG(X_2) = 1 - 0{,}94783 = 0{,}05217$

c/



4.1/

a/ $G(Y) = 1 - (0{,}5^2 + 0{,}5^2) = 0{,}5$

$G(Y|X_1 = T) = 1 - \left[\left(\frac{7}{15}\right)^2 + \left(\frac{8}{15}\right)^2\right] = 0{,}49778$

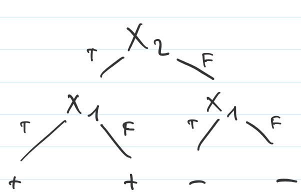
$G(Y|X_1 = F) = 1 - \left[\left(\frac{8}{15}\right)^2 + \left(\frac{7}{15}\right)^2\right] = 0{,}49778$

$\Rightarrow G(Y|X_1) = 0{,}49778$

$G(Y|X_2 = T) = 1 - \left[\left(\frac{9}{14}\right)^2 + \left(\frac{5}{14}\right)^2\right] = 0{,}45918$

$G(Y|X_2 = F) = 1 - \left[\left(\frac{10}{16}\right)^2 + \left(\frac{6}{16}\right)^2\right] = 0{,}46875$

$\Rightarrow G(Y|X_2) = \frac{14}{30} \cdot 0{,}45918 + \frac{16}{30} \cdot 0{,}46875$
$= 0{,}464284$



b/

- Gini Impurity
  - Measures the probability of misclassifying a randomly choosen element
  - Ranges : 0 (pure node) → 0,5 (maximally mixed in binary classification)

  On tree
  - Favors splits that lead to purer nodes early on
    ↳ Often leads to shallower trees with more agressive splitting
  - Less Sensitive to class propotions when they are imbalanced
    ↳ Might split faster even with highly imbalanced classes

  ⇒ Quicker, purer splits ⇒ Shallower trees

- Entropy
  - Measures the uncertainty in the class distribution of a subset
  - Ranges : 0 (pure node) → 1 (maximum uncertainty in binary classification)

  On tree
  - Focus on reducing uncertainty or maximizing information gain.
    ↳ Can lead to deeper trees as it focuses on reducing uncertainty more gradually
  - More sensitive to changes in class distribution, especially when proportions are highly imballanced
    ↳ May result in a more careful, balanced splits in cases of imbalanced data → strategic splitting

  ⇒ Reduce overall uncertainty → deeper trees with balanced splits.

4.2/

c/

$$p_1 = (-4,-2);\ p_2 = (-3,0);\ p_3 = (-2,2);\ p_4 = (2,2);\ p_5 = (3,0);\ p_6 = (4,-2)$$

$$m_c = \frac{1}{6}(-2+0+2+2+0-2) = 0 \Rightarrow S(T) = \{(-2)^2+0^2+2^2+2^2+0^2+2^2] = 8/3$$

+) $1 \mid 2\to 6 \Rightarrow \begin{cases} m_{T_1} = -2,\ m_{T_2} = 0{,}4 \\ S(T_1) = 0;\ S(T_2) = 2{,}24. \\ T = \dfrac{S(T_1)+S(T_2)}{S(T)} = 0{,}84 \end{cases}$

+) $1\to 3 \mid 4\to 6 \Rightarrow \begin{cases} m_{T_1} = 0;\ m_{T_2} = 0 \\ S(T_1) = 8/3;\ S(T_2) = 8/3 \\ T = 2 \end{cases}$

+) $1\to 5 \mid 6 \Rightarrow \begin{cases} m_{T_1} = 0{,}4;\ m_{T_2} = -2 \\ S(T_1) = 2{,}24;\ S(T_2) = 0. \\ T = 0{,}84 \end{cases}$

+) $1,2 \mid 3\to 6 \Rightarrow \begin{cases} m_{T_1} = -1,\ m_{T_2} = 0{,}5 \\ S(T_1) = 1;\ S(T_2) = 2{,}75 \\ T = 1{,}406 \end{cases}$

+) $1\to 4 \mid 5\to 6 \Rightarrow \begin{cases} m_{T_1} = 0{,}5,\ m_{T_2} = -1 \\ S(T_1) = 2{,}75;\ S(T_2) = 1 \\ T = 1{,}406 \end{cases}$

$\Rightarrow T_{min} = 0{,}84 > 0{,}5$, split cannot be performed.

d/ The goal is to find values a and b that split dataset into 3 subsets based on x_1

⇒ Our approach :
1. Sort data by x_1 ⇒ $O(n\log n)$
2. Iterate over all pairs of split points (a,b) ⇒ $O(n^2)$
3. Use prefix sums to compute the sum of squared errors for 3 subsets ⇒ $O(1)$ each pair.

⇒ Total runtime complexity : $O(n^2)$

5/

a/

Classes	Euclidean Distance		
Dogs	$\sqrt{17}$	1	$\sqrt{13}$
Cats	5	$\sqrt{5}$	$\sqrt{2}$
Fish	$\sqrt{34}$	$2\sqrt{5}$	$\sqrt{37}$

b/

- $k = 1$ ⇒ Dogs (1 d)
- $k = 3$ ⇒ Cats (1 d, 2 c)
- $k = 5$ ⇒ Dogs (3 d, 2 c)
- $k = 9$ ⇒ Unclassified (Tie with all classes)

- Low k
  - Sensitive to local patterns
  - Lower bias, capture fine details.
  - Sensitive to noise and outliers
  - Can lead to overfitting.
- High k
  - Robust to noise and outliers
  - Smoother, more general decision boundaries
  - Less sensitive to local patterns
  - Can lead to underfitting

6/

a/ No. The decision boundaries for a 1-NN classifier correspond to the Voronoi cell boundaries of each point, which are not necessarily paralel to the coordinates axes. These boundaries are based on Euclidean distance, creating regions with non-axis-aligned boundaries.
In constrast, DT boundaries are parallel to the coordinate axes because decisions at each node are of the form $x_1 > a$ or $x_2 < b$. It would require large number of axis-aligned splits to approximate a sloped boundary (in 1-NN) with a decision tree, so it isn't feasible.

b/
- DB1: 1-NN because it has jagged and localized bounderies, react strongly to individual points
- DB2: 3-NN ——— smoother — generalized ——— , reflecting the influence of multiple negbors