CHAPTER 3:

Bayesian Decision Theory

Probability and Inference

- Result of tossing a coin is $\in$ {Heads,Tails}
- Random var $X \in \{1,0\}$

  Bernoulli: $P\{X=1\} = p_o^{X}(1-p_o)^{(1-X)}$
- Sample: $\boldsymbol{X} = \{x^t\}_{t=1}^{N}$

  Estimation: p_o = # {Heads}/#{Tosses} = $\sum_t x^t / N$
- Prediction of next toss:

  Heads if $p_o > ½$, Tails otherwise

Classification

- Credit scoring: Inputs are income and savings.
  Output is low-risk vs high-risk
- Input: $\boldsymbol{x} = [x_1,x_2]^T$,Output: C Î {0,1}
- Prediction:

$$\text{choose}\begin{cases} C = 1 \text{ if } P(C=1|x_1,x_2) > 0.5 \\ C = 0 \text{ otherwise} \end{cases}$$

or

$$\text{choose}\begin{cases} C = 1 \text{ if } P(C=1|x_1,x_2) > P(C=0|x_1,x_2) \\ C = 0 \text{ otherwise} \end{cases}$$

Bayes' Rule

prior *likelihood*

posterior

$$P(C|\mathbf{x}) = \frac{P(C)\,p(\mathbf{x}|C)}{p(\mathbf{x})}$$

evidence

$$P(C=0)+P(C=1)=1$$

$$p(\mathbf{x})=p(\mathbf{x}|C=1)P(C=1)+p(\mathbf{x}|C=0)P(C=0)$$

$$p(C=0|\mathbf{x})+P(C=1|\mathbf{x})=1$$

Bayesian rule: Example

- 1% of women have breast cancer. 80% of mammograms detect breast cancer when it is there. 9.6% of mammograms detect breast cancer when it’s **not** there. Now suppose you get a positive test result. What are the chances you have cancer?

Bayes’ Rule: *K*>2 Classes

$$P(C_i \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid C_i)P(C_i)}{p(\mathbf{x})}$$

$$= \frac{p(\mathbf{x} \mid C_i)P(C_i)}{\sum_{k=1}^{K} p(\mathbf{x} \mid C_k)P(C_k)}$$

$$P(C_i) \geq 0 \text{ and } \sum_{i=1}^{K} P(C_i) = 1$$

choose C_i if $P(C_i \mid \mathbf{x}) = \max_k P(C_k \mid \mathbf{x})$

Application: Losses and Risks

- Actions: α_i
- Loss of α_i when the state is C_k but assigned C_i: λ_{ik}
- Expected risk

$$R(\alpha_i \mid \mathbf{x}) = \sum_{k=1}^{K} \lambda_{ik} P(C_k \mid \mathbf{x})$$

$$\text{choose } \alpha_i \text{ if } R(\alpha_i \mid \mathbf{x}) = \min_k R(\alpha_k \mid \mathbf{x})$$

Losses and Risks: 0/1 Loss

- *K* actions α_i, correct decisions have no loss and all errors are equally costly

$$\lambda_{ik} = \begin{cases} 0 \text{ if } i = k \\ 1 \text{ if } i \neq k \end{cases}$$

- Risk of action α_i: $R(\alpha_i \mid \mathbf{x}) = \sum_{k=1}^{K} \lambda_{ik} P(C_k \mid \mathbf{x})$

$$= \sum_{k \neq i} P(C_k \mid \mathbf{x})$$

$$= 1 - P(C_i \mid \mathbf{x})$$

For minimum risk, choose the most probable class

Losses and Risks: Reject

- *In practice, wrong decisions may have high cost, define an additional action of doubt, α_{k+1} with following loss function*

$$\lambda_{ik} = \begin{cases} 0 & \text{if } i = k \\ \lambda & \text{if } i = K+1\,, \quad 0 < \lambda < 1 \\ 1 & \text{otherwise} \end{cases}$$

- Risk of action α_i: $R(\alpha_{K+1} \mid \mathbf{x}) = \sum_{k=1}^{K} \lambda P(C_k \mid \mathbf{x}) = \lambda$

$$R(\alpha_i \mid \mathbf{x}) = \sum_{k \neq i} P(C_k \mid \mathbf{x}) = 1 - P(C_i \mid \mathbf{x})$$

$$\begin{array}{ll} \text{choose } C_i & \text{if } P(C_i \mid \mathbf{x}) > P(C_k \mid \mathbf{x}) \;\; \forall k \neq i \text{ and } P(C_i \mid \mathbf{x}) > 1 - \lambda \\ \text{reject} & \text{otherwise} \end{array}$$

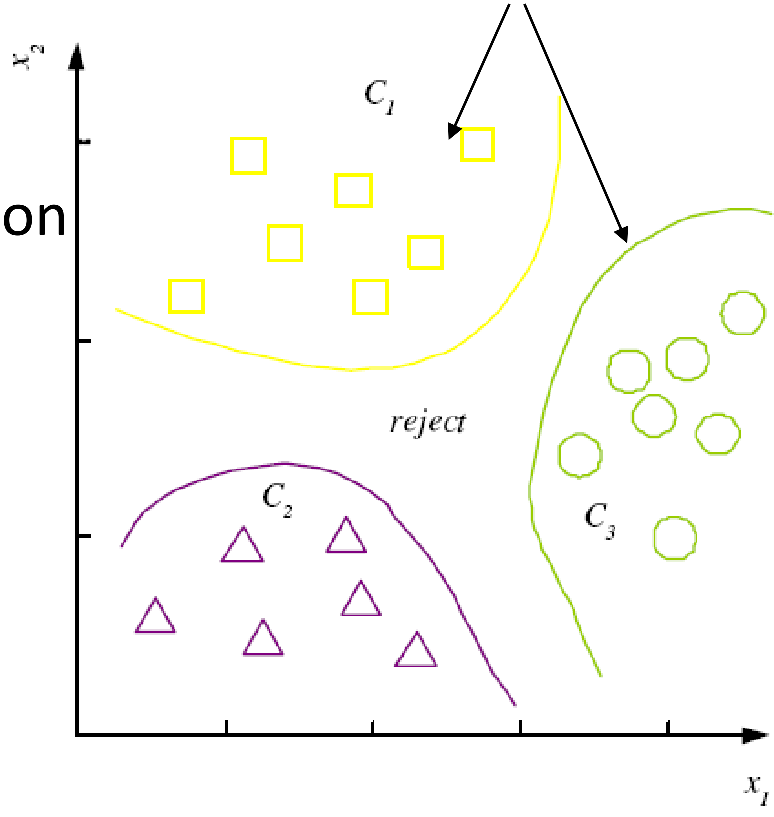
Discriminant Functions

- Classification can be seen as implementing discriminant function

$$\text{choose } C_i \text{ if } g_i(\mathbf{x}) = \max_k g_k(\mathbf{x})$$

$g_i(\mathbf{x}), i = 1,\mathrm{K}\ ,K$

$$g_i(\mathbf{x}) = \begin{cases} -R(\alpha_i \mid \mathbf{x}) & \text{//minimum risk} \\ P(C_i \mid \mathbf{x}) & \text{//for 0/1 loss function} \\ p(\mathbf{x} \mid C_i)P(C_i) & \text{//by ignoring p(}\boldsymbol{x}\text{)} \end{cases}$$



K decision regions $\mathcal{R}_1,\ldots,\mathcal{R}_K$

$$\mathcal{R}_i = \{\mathbf{x} \mid g_i(\mathbf{x}) = \max_k g_k(\mathbf{x})\}$$

Example: Discriminant function, K=2 Classes

- *Define a single discriminant for K =2,*
  - $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$

$$\text{choose} \begin{cases} C_1 \text{ if } g(\mathbf{x}) > 0 \\ C_2 \text{ otherwise} \end{cases}$$

  - *Log odds:* $\log \frac{P(C_1 \mid \mathbf{x})}{P(C_2 \mid \mathbf{x})}$

- When K = 2, classification system is called Dichotomizer
- When K > 2 , classification system is called Polychotomizer

Utility Theory

- Prob of state *k* given exidence ***x***: $P\,(S_k|\mathbf{x})$
- Utility of α_i when state is *k*: U_{ik}
- Expected utility:

$$EU(\alpha_i\,|\,\mathbf{x}) = \sum_k U_{ik} P(S_k\,|\,\mathbf{x})$$

Choose α_i if $EU(\alpha_i\,|\,\mathbf{x}) = \max_j EU(\alpha_j\,|\,\mathbf{x})$

Association Rules

- Association rule: $X \rightarrow Y$
  - *X: Antecedent*
  - *Y: Consequent*
- *People who buy/click/visit/enjoy X are also likely to buy/click/visit/enjoy Y.*
- A rule implies association, not necessarily causation.

Association measures

- Support ($X \rightarrow Y$): $P(X,Y) = \dfrac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers}\}}$

- Confidence ($X \rightarrow Y$): $P(Y \mid X) = \dfrac{P(X,Y)}{P(X)}$

$$= \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers who bought } X\}}$$

- Lift ($X \rightarrow Y$): $= \dfrac{P(X,Y)}{P(X)P(Y)} = \dfrac{P(Y \mid X)}{P(Y)}$
  - aka interest of association

Note: there are more than hundred measures

Association rules: Important points

- Support: Maximize
- Confidence: Should be close to 1 and significantly larger than P(Y)
- Lift:
    - if X and Y are independent lift is close to 1
    - If the ratio differs
        - If > 1, X makes Y more likely
        - If < 1, X makes Y less likely

Typically, minimum support and confidence values are set by the company

Apriority Property

- If (X,Y) is not frequent, none of its supersets can be frequent. ***Or*** All non-empty subsets of frequent item sets are frequent.
  - For (X,Y,Z), a 3-item set, to be *frequent* (have enough support), (X,Y), (X,Z), and (Y,Z) should be frequent.

- Once we find the frequent *k*-item sets, we convert them to rules: X, Y → Z, ...

  and X → Y, Z, ...

Apriori Algorithm: steps

- Frequent item set finding:
  - Start by finding the frequent one-item sets and at each step, inductively, from frequent k-items sets, generate candidate k+1-item sets and then do a pass over the data to check if they have enough support.
- Conversion into rules
  - Spit the k-items into two as antecedent and consequent.
    - Start by putting a single consequent and k-1 items in the antecedent. Check if the rule has enough confidence if not remove
    - Check weather we can move another item from the antecedent to the consequent.
      - For two items to be in consequent, each of the two rules with single consequent should have enough confidence
- ***Implementation Notes***
  - store the frequent itemsets in a hash table: Faster access
  - Candidate item sets will decrease very rapidly as k increases

Apriori Algorithm: Example

- Data:

  1,2,5

  2,4

  2,3

  1,2,4

  1,3

  2,3

  1,3

  2,3

  1,3

  1,2,3,5

  1,2,3

- MinSupport and Confidence are given