CHAPTER 3:

BAYESIAN DECISION THEORY

國立雲林科技大學 資訊工程研究所
張傳育(Chuan-Yu Chang) 博士
Office: EB 212
TEL: 05-5342601 ext. 4516
E-mail: chuany @yuntech.edu.tw
Website: http://MIPL.yuntech.edu.tw

Probability and Inference

- Result of tossing a coin is $\in$ {Heads,Tails}
- Random var X $\in${1,0}
  - X=1 the outcome of a toss is head
  - X=0 the outcome of a toss is tails
- Bernoulli distribution:
  $P\{X=1\} = p_o$, $P(X=0)=1-P(X=1)=1-p_o$
- The probability mass function f of this distribution, over possible outcomes k, is
- $$f(k,p_0) = \begin{cases} p_0 & if\ X = 1 \\ 1-p_0 & if\ X = 0 \end{cases}$$
- $$f(k,p_0) = p_o^{X}(1-p_o)^{(1-X)}$$

Probability and Inference

- Sample: $\boldsymbol{X} = \{x^t\}_{t=1}^N$

  Estimation: p_o = # {Heads}/#{Tosses} = $\sum_t x^t / N$

$$\hat{p}_0 = \frac{\#\{\text{tosses with outcome } heads\}}{\#\{\text{tosses}\}}$$

- Ex: Given the sample {heads, heads, heads, tails, heads, tails, tails, heads, heads}
  X={1,1,1,0,1,0,0,1,1}

$$\hat{p}_0 = \frac{\sum_{t=1}^{N} x^t}{N} = \frac{6}{9}$$

- 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: $\mathbf{x} = [x_1, x_2]^T$,Output: $C \in \{0,1\}$
- Prediction:

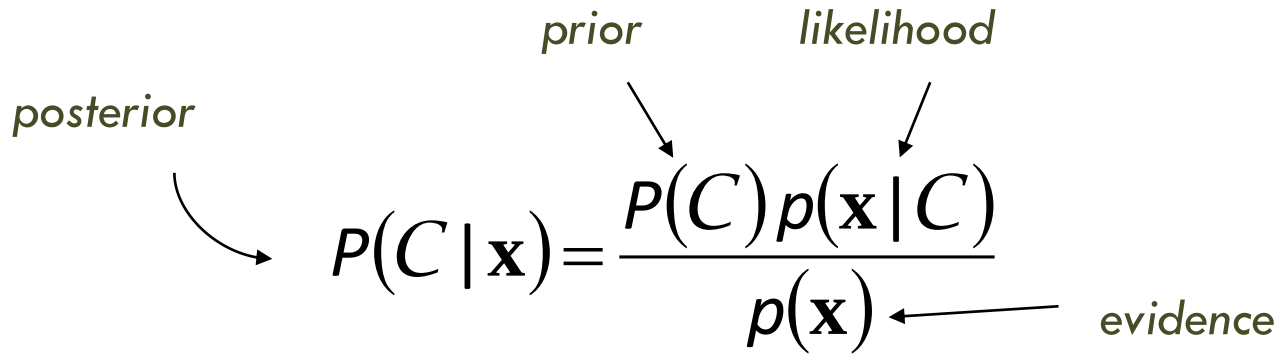
$$\text{choose} \begin{cases} C = 1 \text{ if } P(C = 1 \mid x_1, x_2) > 0.5 \\ C = 0 \text{ otherwise} \end{cases}$$

or

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

Bayes' Rule

Bayes' rule relates the odds of event X_1 to the odds of event X_2, before (prior to) and after (posterior to) conditioning on another event C.



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

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

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

Bayes' Rule: $K>2$ Classes

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

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

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

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

Losses and Risks

- Actions: α_i
- Loss of α_i when the state is C_k : λ_{ik}
- Expected risk (Duda and Hart, 1973)

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

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

Losses and Risks: 0/1 Loss

Define K actions α_i, i=1,…,K, α_i is the action of assigning $\mathbf{x}$ to C_i.

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

The risk of taking action α_i is

$$\begin{aligned} R(\alpha_i \mid \mathbf{x}) &= \sum_{k=1}^{K} \lambda_{ik} P(C_k \mid \mathbf{x}) \\ &= \sum_{k \neq i} P(C_k \mid \mathbf{x}) \\ &= 1 - P(C_i \mid \mathbf{x}) \end{aligned}$$

For minimum risk, choose the most probable class

Losses and Risks: Reject

Define action of reject, α_{K+1}, with α_i, i=1,…,K, being the usual actions of deciding on classes C_i , i=1,…,K.

A possible loss function is

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

The risk of reject is

$$R(\alpha_{K+1} \mid \mathbf{x}) = \sum_{k=1}^{K} \lambda P(C_k \mid \mathbf{x}) = \lambda$$

The risk of choosing class C_i is

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

Losses and Risks: Reject

- The optimal decision rule is to

$$\text{choose } C_i \quad \text{if } R(\alpha_i \mid \mathbf{x}) < R(\alpha_k \mid \mathbf{x}) \text{ for } \forall k \neq i \text{ and } R(\alpha_i \mid \mathbf{x}) < (\alpha_{K+1} \mid \mathbf{x})$$

$$\text{reject} \quad \text{if } R(\alpha_{K+1} \mid \mathbf{x}) < R(\alpha_i \mid \mathbf{x}), i = 1, \ldots, K$$

- Given the loss function of Eq(3.10),

$$\text{choose } C_i \quad \text{if } P(C_i \mid \mathbf{x}) > P(C_k \mid \mathbf{x}) \ \forall k \neq i \text{ and } P(C_i \mid \mathbf{x}) > 1 - \lambda$$

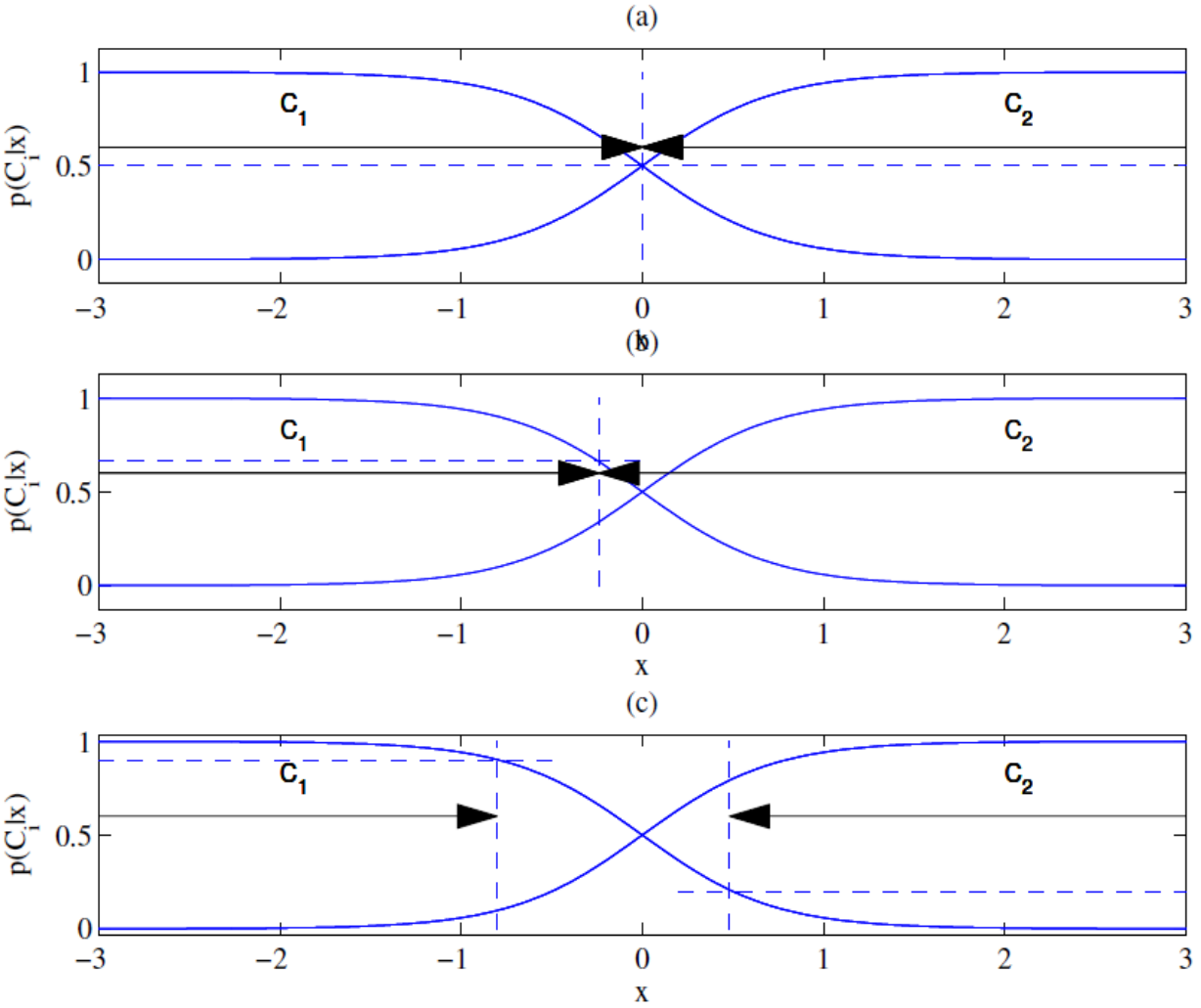
$$\text{reject} \qquad \text{otherwise}$$

Different Losses and Reject

Equal losses

Unequal losses

With reject



Discriminant Functions

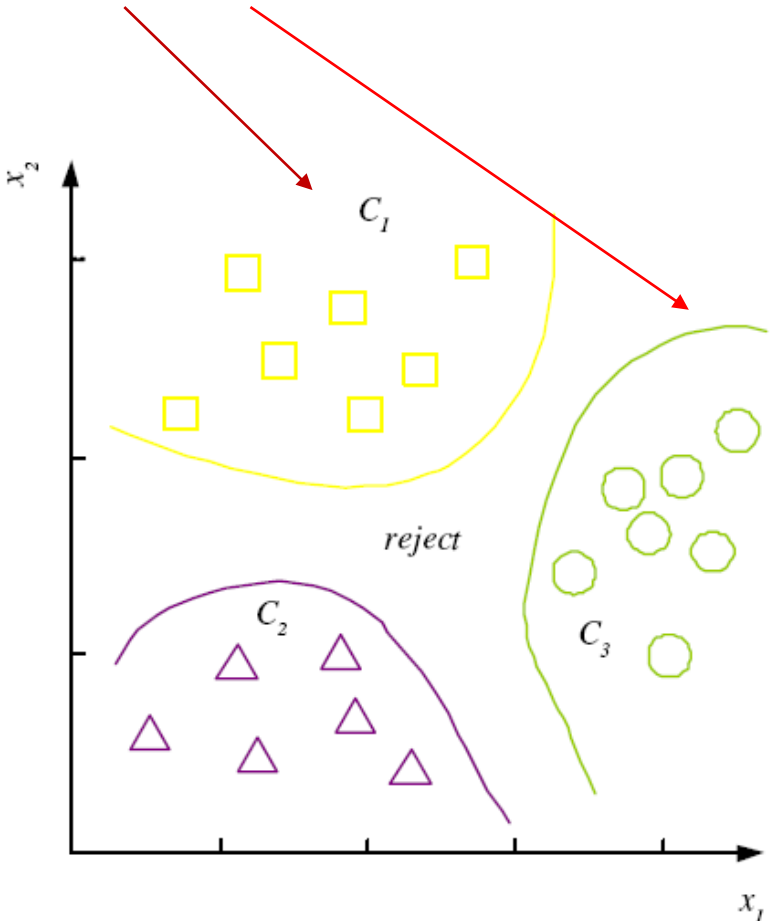
- Classification can be seen as implementing a set of discriminant functions, $g_i(\mathbf{x}), i = 1, \ldots, K$, such that we

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

$$g_i(\mathbf{x}) = \begin{cases} -R(\alpha_i \mid \mathbf{x}) \\ P(C_i \mid \mathbf{x}) \\ p(\mathbf{x} \mid C_i) P(C_i) \end{cases}$$

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

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



$K=2$ Classes

- Dichotomizer ($K=2$) vs Polychotomizer ($K>2$)
- $g(\boldsymbol{x}) = g_1(\boldsymbol{x}) - g_2(\boldsymbol{x})$

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

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

Association Rules

- An association rule is an implication of the form
  $X \rightarrow Y$ where X is the antecedent and Y is the consequent of the rule.
- *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 ($X \rightarrow Y$):

$$P(X,Y) = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers}\}}$$

- Confidence ($X \rightarrow Y$):

$$P(Y \mid X) = \frac{P(X,Y)}{P(X)} = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers who bought } X\}}$$

- Lift ($X \rightarrow Y$):

$$= \frac{P(X,Y)}{P(X)P(Y)} = \frac{P(Y \mid X)}{P(Y)}$$

Example

- Given the following data of transactions at a shop, calculate the support and confidence values of milk→banana, banana →milk, milk →chocolate, chocolate →milk

Transaction	Items in basket
1	milk, bananas, chocolate
2	milk, chocolate
3	milk, bananas
4	chocolate
5	chocolate
6	milk, chocolate

SOLUTION:

milk → bananas : Support = 2/6, Confidence = 2/4

bananas → milk : Support = 2/6, Confidence = 2/2

milk → chocolate : Support = 3/6, Confidence = 3/4

chocolate → milk : Support = 3/6, Confidence = 3/5

Apriori algorithm (Agrawal et al., 1996)

- Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.
- For (X,Y,Z), a 3-item set, to be frequent (have enough support), (X,Y), (X,Z), and (Y,Z) should be frequent.
- If (X,Y) is not frequent, none of its supersets can be frequent.
- Once we find the frequent *k*-item sets, we convert them to rules: X, Y $\rightarrow$ Z, ...

  and X $\rightarrow$ Y, Z, ...

Example

(Ref: https://en.wikipedia.org/wiki/Apriori_algorithm)

- Consider the following database, where each row is a transaction and each cell is an individual item of the transaction:

alpha	beta	epsilon
alpha	beta	theta
alpha	beta	epsilon
alpha	beta	theta

- The association rules that can be determined from this database are the following:
- 100% of sets with alpha also contain beta
- 50% of sets with alpha, beta also have epsilon
- 50% of sets with alpha, beta also have theta

Example

(Ref: https://en.wikipedia.org/wiki/Apriori_algorithm)

- Let the database of transactions consist of following itemsets:
- Using the Apriori to determine the frequent item sets of this database.
  - An item set is frequent if it appears in at least 3 transactions of the database: the value 3 is the support threshold.

Itemsets
{1,2,3,4}
{1,2,4}
{1,2}
{2,3,4}
{2,3}
{3,4}
{2,4}

- The first step of Apriori is to count up the number of occurrences, called the support, of each member item separately, by scanning the database a first time. We obtain the following result

Item	Support
{1}	3
{2}	6
{3}	4
{4}	5

Example

(Ref: https://en.wikipedia.org/wiki/Apriori_algorithm)

- All the itemsets of size 1 have a support of at least 3, so they are all frequent.
- The next step is to generate a list of all pairs of the frequent items:
- The pairs {1,2}, {2,3}, {2,4}, and {3,4} all meet or exceed the minimum support of 3, so they are frequent. The pairs {1,3} and {1,4} are not and can be pruned.
- Look for frequent triples in the database, but we can already exclude all the triples that contain one of these two pairs:

Item	Support
{1,2}	3
{1,3}	1
{1,4}	2
{2,3}	3
{2,4}	4
{3,4}	3

Item	Support
{2,3,4}	2

Utility Theory

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

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

$$\text{Choose } \alpha_i \text{ if } EU(\alpha_i \mid \mathbf{x}) = \max_j EU(\alpha_j \mid \mathbf{x})$$