CENG 463
Machine Learning

Lecture 02 - Bayes Decision Theory

Bayes Decision Theory

- It is a statistical approach to machine learning.
- Assumptions:
  - Decision problem is probabilistic
  - All relevant probability values are known
- We derive decision rules that are optimal in the sense that they either minimize average probability of **error** or overall **risk**.

Bayes Decision Theory

Example: Credit scoring for bank customers

- Inputs are income and savings: $x = [x_1, x_2]^T$
- There are two classes: low-risk customers (C=1) and high-risk customers (C=0).
- Probabilities:
  - $P(C=1 | x_1, x_2)$: probability that input belongs to class 1
  - $P(C=0 | x_1, x_2)$: probability that input belongs to class 0
  - $P(C=0) + P(C=1) = 1$

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

Bayes' Rule (Two-Class Case)

$$\underbrace{P(C \mid \mathbf{x})}_{\text{posterior}} = \frac{\overbrace{P(C)}^{\text{prior}}\;\overbrace{P(\mathbf{x} \mid C)}^{\text{likelihood}}}{\underbrace{P(\mathbf{x})}_{\text{evidence}}}$$

- $P(C=0) + P(C=1) = 1$
- $P(x) = P(x|C=0) \cdot P(C=0) + P(x|C=1) \cdot P(C=1)$
- $P(C=0|x) + P(C=1|x) = 1$

Bayes' Rule: Example

We have found that the word 'Rolex' occurs in 250 of 2000 spam messages, and in 5 of 1000 non-spam messages. What is the probability that a new message containing the word 'Rolex' is spam? Assuming it is equally likely that this new message is spam or non-spam.

Lets denote the class spam by S and non-spam by NS.

Our feature (x) is observing the word Rolex.

$$P(S|x) = \frac{P(S)\,P(x|S)}{P(x)} = \frac{P(S)\,P(x|S)}{P(x|S)P(S) + P(x|NS)P(NS)}$$

Bayes' Rule: Example

We have found that the word 'Rolex' occurs in 250 of 2000 spam messages, and in 5 of 1000 non-spam messages.

P(x|S) = 250/2000   P(x|NS) = 5/1000

We assume S and NS are equally likely:

$$P(S \mid x) = \frac{P(S)\,P(x \mid S)}{P(x)} = \frac{0.5 \cdot 0.125}{0.5 \cdot 0.125 + 0.5 \cdot 0.005} = \frac{125}{130}$$

What is the probability of being non-spam?

Note that the evidence P(x) is only necessary for normalization purposes; it does not affect the decision.

Decision

Example: Classification problem of apple and peach by observing their color.

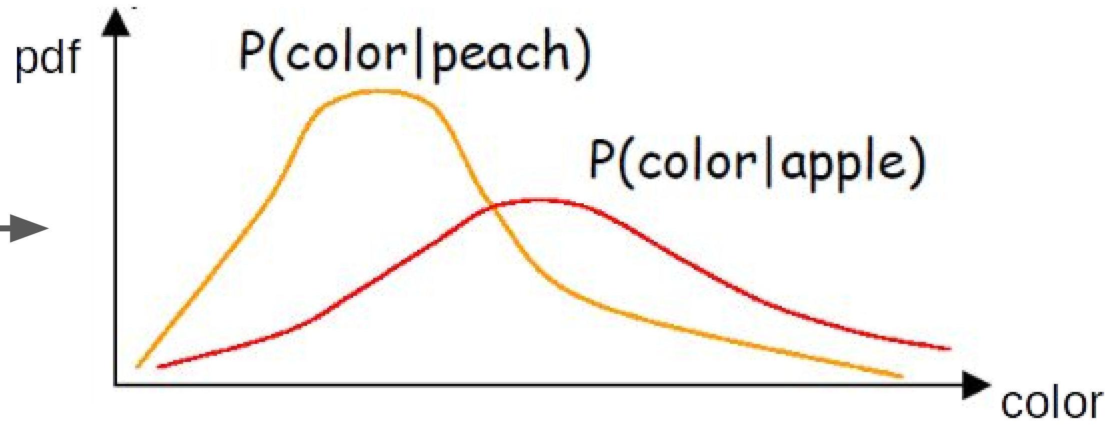
- If we assume initial probabilities are not equal, i.e. if P(apple) > P(peach) and if we do not have a chance to see the fruit, we always predict apple!
- If we see the color of the fruit, we compute posterior (conditional) probabilities:
  P(apple|color)=? , P(peach|color)= ?
- We choose the class with higher conditional probability.
- How to find these probabilities? **Bayes Rule**

Decision

- Posterior (conditional) probabilities are
  - P(apple|color) = P(color|apple)*P(apple) / P(color)
  - P(peach|color) = P(color|peach)*P(peach) / P(color)

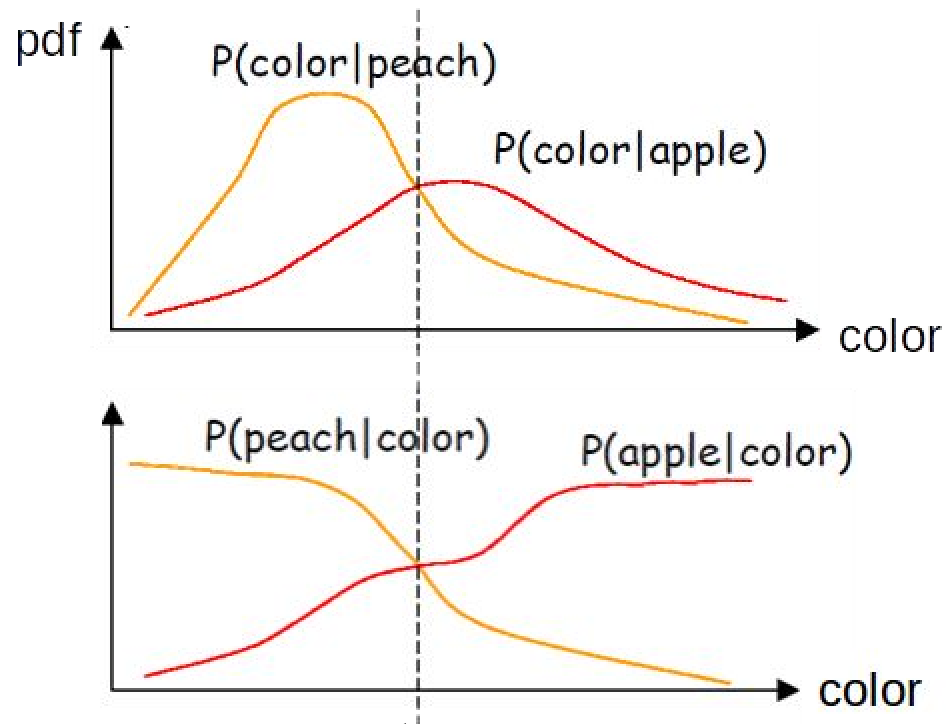
- We need likelihoods: P(color|apple), P(color|peach).
- A way of describing likelihood is **probability distribution functions (pdf)**:

For this example, assume, color means intensity of light, i.e. how dark or bright is the fruit.



Decision

Using probability distribution functions:



Posterior probabilities after applying Bayes rule:

Intersections are co-located only when prior probabilities are equal, P(apple)=P(peach).

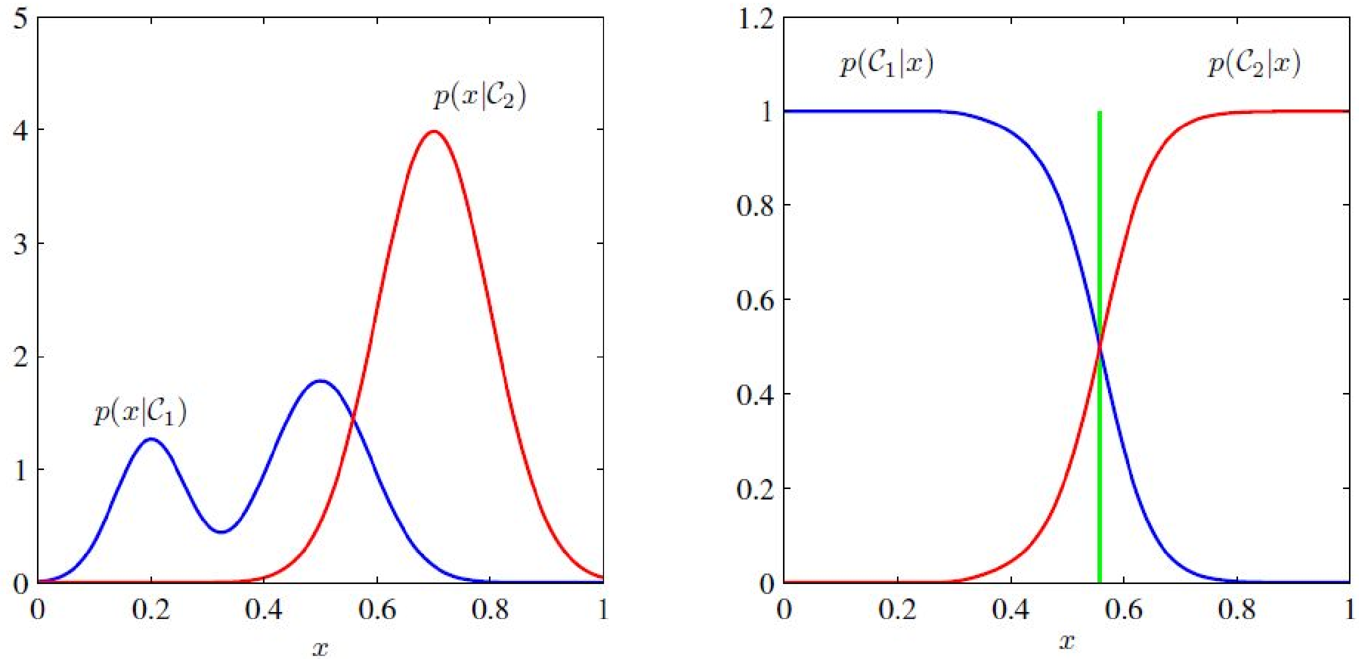
<u>Decision:</u>
if P(apple|color) > P(peach|color)
then choose apple

Two-Class Example

- Assume our color is red and we have the following likelihoods and priors:
  - P(red|apple)=0.5, P(red|peach)=0.2
  - P(apple)=0.4, P(peach)=0.6
- Given a red fruit, would you say it is an apple or a peach?
- Posteriors:
  - P(apple|red)=0.5*0.4=0.2
  - P(peach|red)=0.2*0.6=0.12
- You would say **'apple'**!
- Normalized posteriors:

$$\left.\begin{array}{l} P(apple|red)= \ 0.2/0.32=0.625 \\ P(peach|red)=0.12/0.32=0.375 \end{array}\right\} \text{sum = 1}$$

Another Visual Two-Class Example



Probability distributions (left plot) and their posterior probabilities (right plot). Note that the left plot affects the posterior probabilities only when 0.35<x<0.8.

The vertical green line in the right plot shows the decision boundary.

Figures from Pattern Recognition and Machine Learning, C. Bishop, 2006.

Bayes' Rule (Multi-Class Case)

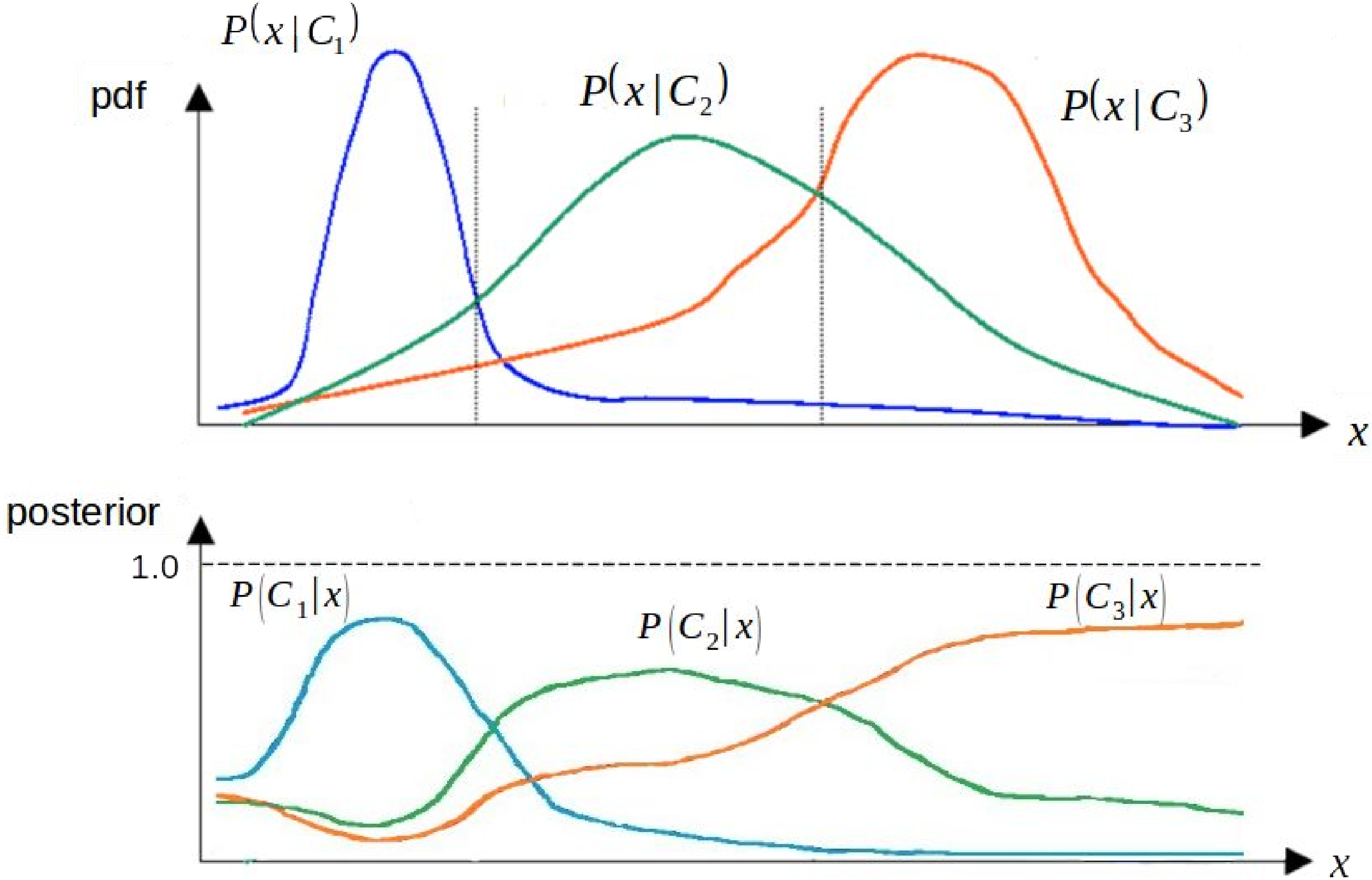
$$P(C_i \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C_i) P(C_i)}{P(\mathbf{x})}$$

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

$$\sum_{i=1}^{K} P(C_i) = 1$$

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

Bayes' Rule (Multi-Class Case)



Losses and Risks

- **Action**, α_i : assigning an input to C_i
- **Loss**, λ_{ik}: Loss of α_i when the actual class is C_k
- Expected risk of choosing α is the sum of losses over all classes:

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

- We choose the class with minimum expected risk:

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

- It is meaningful to define losses as: $\lambda_{ik} = \begin{cases} 0 \text{ if } & i = k \\ 1 \text{ if } & i \neq k \end{cases}$

Losses and Risks

- When we define losses as:

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

- Then risk becomes:

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

- For minimum risk, choose the most probable class.

Losses and Risks: Reject

- In some applications, the cost of choosing a wrong class is very high. So, we may think about a third option: **rejecting** to classify the sample.
- Assume we have a loss of rejection, such that 0<λ<1

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

Risk of rejection: $$R(\alpha_{K+1} | \mathbf{x}) = \sum_{k=1}^{K} \lambda P(C_k | \mathbf{x}) = \lambda \sum_{k=1}^{K} P(C_k | \mathbf{x}) = \lambda$$

Risk of choosing C_i : $$R(\alpha_i | \mathbf{x}) = \sum_{k \neq i} P(C_k | \mathbf{x}) = 1 - P(C_i | \mathbf{x})$$

Losses and Risks: Reject

- Since we have

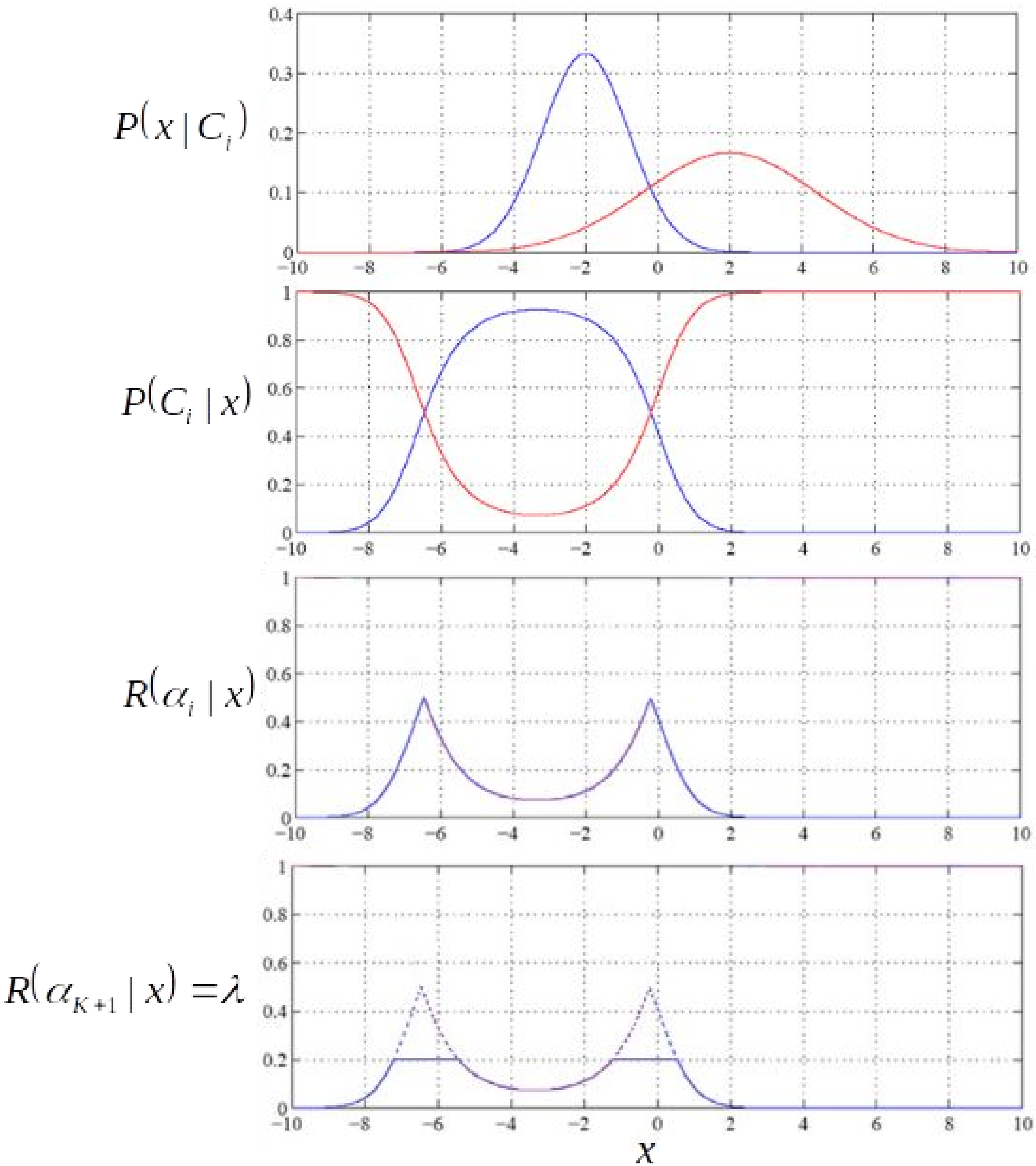
$$R(\alpha_{K+1} \mid \mathbf{x}) = \lambda$$

and

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

- If probability of belonging to class C_i is low or risk is high, we should choose to reject.
- Final decision criterion:

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



Likelihoods

Posteriors

Risk

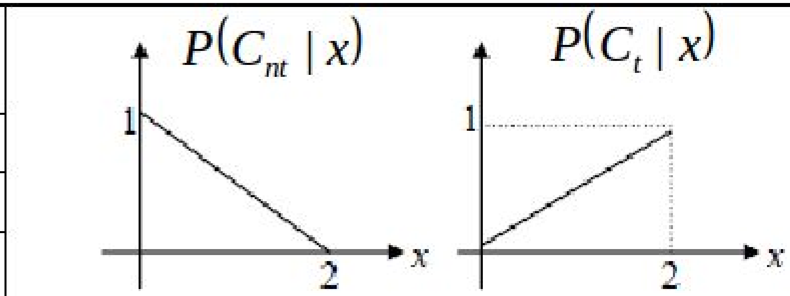
Risk with rejection, rejection loss (λ)=0.2

Figures from Introduction to Machine Learning 2ed., E Alpaydın, 2010.

Losses and Risks: Example

- Assume some sensor data, **x**, is used to detect targets. There are 2 classes: target and no-target.
- If target, system responds, if no-target, system stays idle. A third option is rejection of making a decision.
- The following table is constructed with past experience:

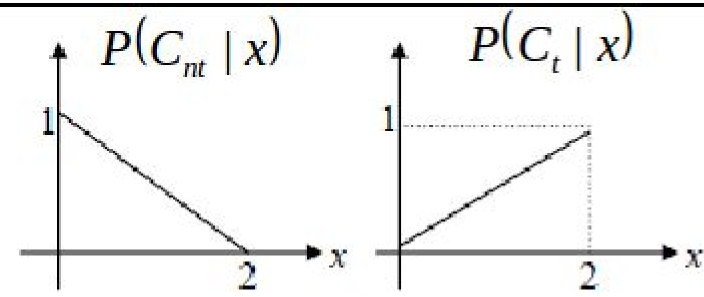
LOSS	*Target*	*No-target*
Respond	0	2ε
Stay Idle	4ε	0
No-decision	ε	ε



- If your sensor data x=1.5, would you respond? Would you stay idle? Would you reject to make a decision?

Losses and Risks: Example

LOSS	*Target*	*No-target*
Respond	0	2ε
Stay Idle	4ε	0
No-decision	ε	ε



$$R(\alpha_t \mid x) = \sum_{k \neq t} \lambda_{t,k} P(C_k \mid x) = \lambda_{t,nt} P(C_{nt} \mid x) \rightarrow R(\alpha_t \mid 1.5) = 2\varepsilon \cdot 0.25 = 0.5\varepsilon$$

$$R(\alpha_{nt} \mid x) = \sum_{k \neq nt} \lambda_{nt,k} P(C_k \mid x) = \lambda_{nt,t} P(C_t \mid x) \rightarrow R(\alpha_{nt} \mid 1.5) = 4\varepsilon \cdot 0.75 = 3\varepsilon$$

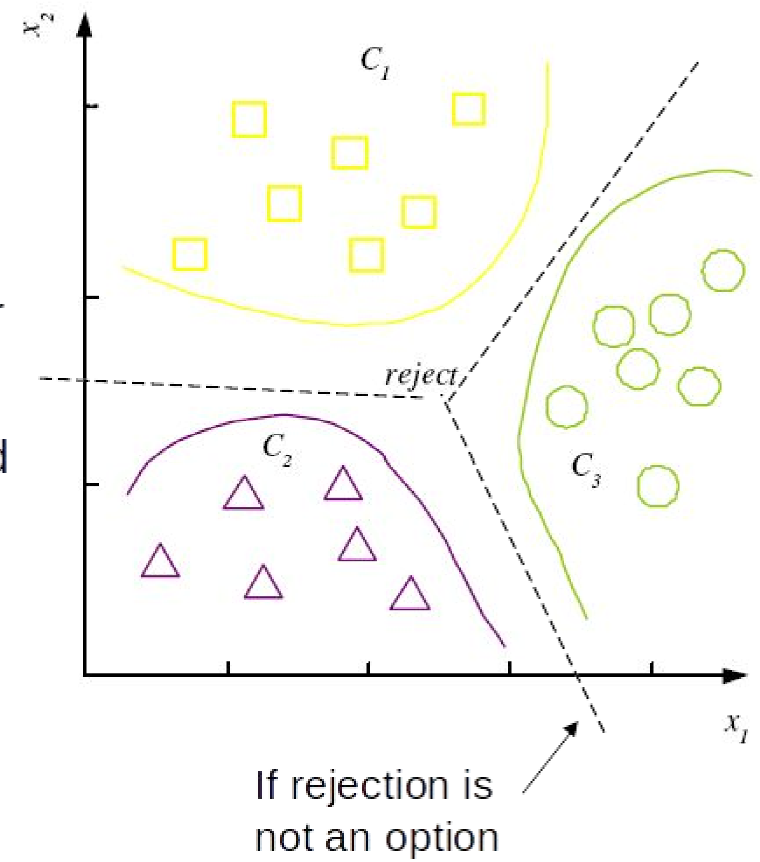
$$R(\alpha_{K+1} \mid x) = \sum_{k=1}^{K} \lambda P(C_k \mid x) = \lambda (P(C_t \mid x) + P(C_{nt} \mid x)) \rightarrow R(\alpha_{K+1} \mid 1.5) = \varepsilon \cdot 1 = \varepsilon$$

Discriminant Functions

- Classification is done by defining discriminant function g(x).

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

$$g_i(\mathbf{x}) = \begin{cases} -R(\alpha_i \mid \mathbf{x}) & \longleftarrow \text{minimum risk} \\ P(C_i \mid \mathbf{x}) & \longleftarrow \text{maximum posterior} \\ p(\mathbf{x} \mid C_i)P(C_i) & \longleftarrow \text{unnormalized posterior} \end{cases}$$



- K decision regions $R_1,\ldots,R_K$

$$R_i = \{\mathbf{x} \mid g_i(\mathbf{x}) = \max_k g_k(\mathbf{x})\}$$

Figures from Introduction to Machine Learning 2ed., E Alpaydın, 2010.

Summary

We have learned about:

- Bayes' Rule
- How to Make a Decision using Bayes' Rule?
- Risk
- Risk with Rejection Option
- What are Discriminant Functions?