Machine Learning 1
Lecture 5.5 - Supervised Learning
Classification - Decision Theory

Erik Bekkers

(Bishop 1.5)
Decision theory

- Dataset: Input vectors \( \mathbf{x} \in \mathbb{R}^D \), ground truth targets \( t \in \{ C_1, \ldots, C_k \} \)
- Divide input space \( \mathbb{R}^D \) into \( K \) decision regions \( \mathcal{R}_k, \ k = 1, \ldots, K \)
- Every observed datapoint

\[
\text{ground truth } b_n = C_i \\
\text{prediction } \hat{t}_n = C_k \quad (\hat{t}_n \notin \mathcal{R}_k)
\]

- Confusion matrix: ground truth classes vs. predicted classes

\[
\begin{pmatrix}
C_1 & \mathcal{R}_1 & 6 & 1 & \ldots & 0 \\
C_2 & \mathcal{R}_2 & 5 & 3 & \ldots & 1 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
C_K & \mathcal{R}_K & 2 & 0 & \ldots & 8
\end{pmatrix}
\]

- Diagonal elements: correctly classified
- Off-diagonal elements: misclassified
Decision theory: Misclassification Rate

- Classification goal: Minimize the misclassification rate

- Assume observations are drawn from joint distribution $p(x, t)$

- Probability of a misclassification:

  $$p(\text{mistake}) = \sum_{i=1}^{K} \sum_{k \neq i} p(x \in R_i, C_k)$$

  $$= 1 - \sum_{k=1}^{K} p(x \in R_k, C_k)$$

Minimizing misclassification rate

- Assign $x$ to class $C_k$ if

  $$p(x, t = C_k) > p(x, t = C_j), j \neq k$$

- Note: $p(x, C_k) = p(C_k|x)p(x)$

  - Check for the largest posterior class prob

    $$p(C_k|x) > p(C_j|x), j \neq k$$
Decision theory: Misclassification Rate

\( \hat{x} \) : Decision boundary

\( x_0 \) : Optimal Decision boundary

\( p(x, C_1) = p(x, C_2) \)

**Figure:** joint probability distributions and decision boundary (Bishop 1.24)
Minimizing the Misclassification Rate: Problems

• Not all errors have the same impact!

**Example: Medical diagnosis of cancer**

• Error 1: Label a healthy person as having cancer.

• Error 2: Label a sick person as healthy. Lack of treatment!

• If cancer only occurs in 1% of all patients, a classifier which labels everyone as healthy has a misclassification rate of 1%!

Class Imbalance
Possible solution: use different weights for different error types

\[ L = \begin{pmatrix}
  0 & 1000 \\
  1 & 0
\end{pmatrix} \]

Expected loss: \( \mathbb{E}[L] = \sum_{k, j} L_{kj} \int_{\mathbb{R}_j} p(x, C_k) dx \)

Minimize expected loss:

Assign \( x \) to \( C_k \) if \( \sum_{j=1}^{K} L_{jk} p(x, C_j) \) is minimal.
Classification Strategies

1. **Discriminant functions**
   Direct mapping of input to target \( t = y(x, w) \)

2. **Probabilistic discriminative models**
   Posterior class probabilities:
   \[
   p(C_k | x) \]

3. **Probabilistic generative models**
   Class-conditional densities:
   \[ p(x | C_k) \]
   Prior class probabilities:
   \[ p(C_k) \]
   \[
   \begin{align*}
   1. & \quad p(x, C_k) = p(x | C_k) p(C_k) \\
   2. & \quad p(C_k | x) = \frac{p(x | C_k) p(C_k)}{p(x)}
   \end{align*}
   \]