Lecture 15.
Pattern Classification (I): Statistical Formulation

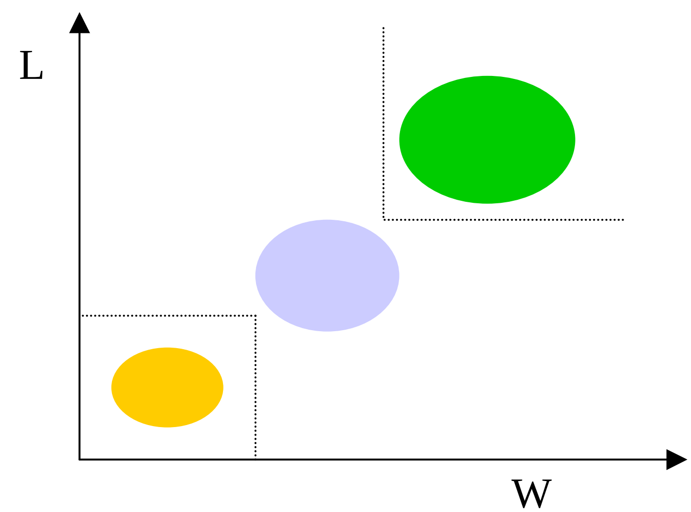
Outline

- Statistical Pattern Recognition
  - Maximum Posterior Probability (MAP) Classifier
  - Maximum Likelihood (ML) Classifier
  - K-Nearest Neighbor Classifier
  - MLP classifier

An Example

- Consider classify eggs into 3 categories with labels: medium, large, or jumbo.
- The classification will be based on the weight and length of each egg.
- Decision rules:
  1. If W < 10 g & L < 3cm, then the egg is medium
  2. If W > 20g & L > 5 cm then the egg is jumbo
  3. Otherwise, the egg is large

- Three components in a pattern classifier:
  – Category (target) label
  – Features
  – Decision rule



Statistical Pattern Classification

- The objective of statistical pattern classification is to draw an optimal decision rule given a set of training samples.
- The decision rule is optimal because it is designed to minimize a cost function, called the expected risk in making classification decision.
- This is a learning problem!

Assumptions

1. Features are given.
   - Feature selection problem needs to be solved separately.
   - Training samples are randomly chosen from a population
2. Target labels are given
   - Assume each sample is assigned to a specific, unique label by the nature.
   - Assume the label of training samples are known.

Pattern Classification Problem

Let X be the feature space, and

$C = \{c(i), 1 \le i \le M\}$ be M class labels.

For each $x \in X$, it is assumed that the nature assigned a class label $t(x) \in C$ according to some probabilistic rule.

Randomly draw a feature vector x from X,

$P(c(i)) = P(x \in c(i))$ is the *a priori* probability that $t(x) = c(i)$ without referring to x.

$P(c(i)|x) = P(x \in c(i)|x)$ is the *posteriori* probability that $t(x) = c(i)$ *given* the value of x

$P(x|c(i)) = P(x | x \in c(i))$ is the conditional probability (a.k.a. *likelihood function*) that x will assume its value *given* that it is drawn from class c(i).

P(x) is the *marginal* probability that x will assume its value without referring to which class it belongs to.

Use Bayes' Rule, we have

$$P(x|c(i))P(c(i)) = P(c(i)|x)P(x)$$

Also,

$$P(c(i)\,|\,x) = \frac{P(x\,|\,c(i))P(c(i))}{\sum_{i=1}^{M} P(x\,|\,c(i))P(c(i))}$$

Decision Function and Prob. Mis-Classification

- Given a sample x $\in$ X, the objective of statistical pattern classification is to design a decision rule g(x) $\in$ C to assign a label to x.
- If g(x) = t(x), the naturally assigned class label, then it is a correct classification. Otherwise, it is a mis-classification.
- Define a 0-1 loss function:

$$\ell(x \mid g(x)) = \begin{cases} 0 & \text{if } g(x) = t(x) \\ 1 & \text{if } g(x) \neq t(x) \end{cases}$$

Given that *g(x)* = *c(i*)*, then

$$P(\ell(x \mid g(x) = c(i^*)) = 0 \mid x)$$

$$= P(t(x) = c(i^*) \mid x) = P(c(i^*) \mid x)$$

Hence the probability of mis-classification for a specific decision g(x) = c(i*) is

$$P(\ell(x \mid g(x) = c(i^*)) = 1 \mid x)$$

$$= 1 - P(c(i^*) \mid x)$$

Clearly, to minimize the Pr. of mis-classification for a given x, the best choice is to choose g(x) = c(i*) if

P(c(i*)|x) > P(c(i)|x) for i $\neq$ i*

MAP: Maximum A Posteriori Classifier

The MAP classifier stipulates that the classifier that minimizes pr. of mis-classification should choose

$g(x) = c(i^*)$ if

$P(c(i^*)|x) > P(c(i)|x), i \neq i^*$.

This is an optimal decision rule.

Unfortunately, in real world applications, it is often difficult to estimate $P(c(i)|x)$.

Fortunately, to derive the optimal MAP decision rule, one can instead estimate a *discriminant function* $G_i(x)$ such that for any $x \in X$, $i \neq i^*$.

$$G_{i^*}(x) > G_i(x) \text{ iff}$$

$$P(c(i^*)|x) > P(c(i)|x)$$

$G_i(x)$ can be an approximation of $P(c(i)|x)$ or any function satisfying above relationship.

Maximum Likelihood Classifier

Use Bayes rule,

p(c(i)|x) = p(x|c(i))p(c(i))/p(x).

Hence the MAP decision rule can be expressed as:

g(x) = c(i*) if

p(c(i*))p(x|c(i*)) > p(c(i))p(x|c(i)), $i \neq i^*$.

Under the assumption that the a priori Pr. is unknown, we may assume p(c(i)) = 1/M. As such, maximizing p(x|c(i)) is equivalent to maximizing p(c(i)|c).

- The *likelihood function* p(x|c(i)) may assume a uni-variate Gaussian model. That is,

  $$p(x|c(i)) \sim N(\mu_i,\Sigma_i)$$

  μ_i,Σ_i can be estimated using samples from {x|t(x) = c(i)}.
- A priori pr. p(c(i)) can be estimated as:

$$P(c(i)) = \frac{\{x; \#\, x \text{ s. t. } t(x) = c(i)\}}{|X|}$$

Nearest-Neighbor Classifier

- Let {y(1), • • •, y(n)} $\in$ X be n samples which has already been classified. Given a new sample x, the NN decision rule chooses g(x) = c(i) if

$$y(i^*) = \underset{1 \le i \le n}{Min.} \| y(i) - x \|$$

  is labeled with c(i).
- As n $\rightarrow\infty$, the prob. Mis-classification using NN classifier is at most twice of the prob. Mis-classification of the optimal (MAP) classifier.
- k-Nearest Neighbor classifier examine the k-nearest, classified samples, and classify x into the majority of them.
- Problem of implementation: require large storage to store ALL the training samples.

MLP Classifier

- Each output of a MLP will be used to approximate the a posteriori probability P(c(i)|x) directly.
- The classification decision then amounts to assign the feature to the class whose corresponding output at the MLP is maximum.
- During training, the classification labels (1 of N encoding) are presented as target values (rather than the true, but unknown, a posteriori probability)
- Denote y(x,W) to be the i^{th} output of MLP, and t(x) to be the corresponding target value (0 or 1) during the training.

$$
\begin{aligned}
e^2(t) &= E\left\{\| t(x) - y(x,W) \|^2\right\} \\
&= E \| t(x) - E[t(x)\,|\,x] + \\
&\qquad E[t(x)\,|\,x] + y(x,W) \|^2 \} \\
&= E\left\{\| t(x) - E[t(x)\,|\,x] \|^2\right\} + \\
&\qquad E\left\{\| E[t(x)\,|\,x] - y(x,W) \|^2\right\} \\
&\geq E\left\{\| E[t(x)\,|\,x] - y(x,W) \|^2\right\}
\end{aligned}
$$

- Hence *y(x,W)* will approximate E(t(x)|x) = P(c(i)|x)