INFO411 Lecture - Classification I

Jeremiah Deng

University of Otago

15 August 2017

Outline

The Problem

- Instances with class labels are given
  - "Training data"
  - Supervised learning
- Task: predict the class label of a new instance
- How?
- Ref.: Alpaydin, Chapters 5,8,10

Approaches

- **Parametric:** uses Bayesian theory and probability estimation techniques to model the data
  - Bayes classifier
- **Non-parametric:** no modeling needed
  - Nearest neighbour
  - Linear discriminant analysis
  - Learning vector quantization
  - Support vector machine
- **Non-metric:** handles nominal data that have no natural notion of similarity
  - Decision tree
  - Random forest

Review - Probability Theory

- Independent events $P(A, B) = P(A)P(B)$
- Joint probability and conditional $P(A, B) = P(A)|P(B|A)$
- The Law of Total Probability (B_i mutually exclusive)

$$P(A) = \sum_{i=1}^{N} P(A|B_i)P(B_i)$$

- Bayes' Theorem (named after Rev Thomas Bayes, later developed by Laplace)

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{P(A)}$$

- "Is to the theory of probability what Pythagoras's theorem is to geometry"

Bayes' Rule - Interpretation

- Another form

$$P(B_i|A) = \frac{P(A|B_i)P(B_i)}{\sum_j P(A|B_j)P(B_j)}$$

- Suppose A is our data instance, $\{B_i\}$ are the classes.
- $P(B_i)$: *priors*
- $P(A|B_i)$: *conditional*
- $P(B_i|A)$: *posterior*.
- From conditional prob. and priors, we get to know posterior
- Or in plain English: posterior=$\frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$
- i.e., classification done!

Example 1: A *Scary* Test

- A patient is tested positive for a rare disease D.
- Panic?
- What is known:
  - Test is *quite* accurate: $P(T^+|D) = 0.95$, $P(T^+|!D) = 0.02$
  - Priors. $P(D) = 0.001$ (being ill), so $P(!D) = 0.999$ (being ok).
- Test positive, how *likely* is the patient ill?
- Let's try with Bayes Law:

$$P(D|T^+) = \frac{P(T^+|D)P(D)}{P(T^+)}$$

$$P(T^+) = P(T^+|D)P(D) + P(T^+|!D)P(!D)$$

A Famous Example: OJ Simpson Case

- In the Simpson case, defendant attorney Alan Dershowitz repeatedly declared that fewer than 1 in 1,000 women who are abused by their mates go on to be killed by them.
- What's the chance that Simpson is guilty of murder?
- Denote:
  - Sample space S - married couples in which the husband beat his wife
  - Event H - cases in S in which the husband has since murdered his wife; $!H$ - wife in S not murdered by husband
  - Event M - all cases in S in which the wife has been murdered
- We have $P(H) = 0.001, P(M|H) = 1, P(M|!H) = 0.0001$ at most in US.
- $P(H|M) = \frac{P(M|H)P(H)}{P(M|H)P(H)+P(M|!H)P(!H)} = \frac{0.001}{0.001+0.0001\times 0.999} = 0.91$

Bayesian Decision Theory

- Foundation for statistical pattern recognition
- Assume
  - A set of classes $\{\omega_1, \omega_2, ..., \omega_c\}$
  - Prior prob. known: $P(\omega_i)$
  - Conditionals known: $p(x|\omega_i)$
  - Posterior prob. ?

$$P(\omega_i|x) = \frac{p(x|\omega_i)P(\omega_i)}{p(x)}$$

- Define the classification error:

$$R_i(x) = \sum_{j \neq i} P(\omega_j|x) = 1 - P(\omega_i|x)$$

- To minimize classification error, we should assign x to the class that maximize the posterior prob.

Bayesian Classifier

- Define and use a "discriminant function" $g_i(\mathbf{x})$
- Principle:

  Assign $\mathbf{x}$ to class ω_i , if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$.
- Usually we can have
  - $g_i(\mathbf{x}) = P(\omega_i|\mathbf{x})$, or
  - $g_i(\mathbf{x}) = \ln p(\mathbf{x}|\omega_i) + \ln P(\omega_i)$.
- Find decision region boundaries:

  let $g_i(\mathbf{x}) = g_j(\mathbf{x})$
- For $p(\mathbf{x}|\omega_i)$ of Gaussian (normal) density:

$$g_i(\mathbf{x}) = -\tfrac{1}{2}(\mathbf{x} - \mu_i)^t \Sigma_i^{-1}(\mathbf{x} - \mu_i) - \tfrac{1}{2}\ln|\Sigma_i| + \ln P(\omega_i)$$

  Σ_i: covariance matrix of class ω_i

A Simple Case

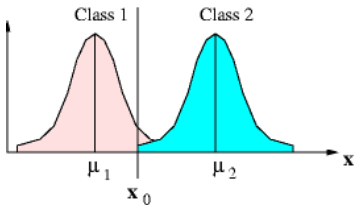
- Two classes:

  Same prior, same covariance: $\Sigma_i = \sigma I$, $P(\omega_1) = P(\omega_2)$.
- Discrimination function

$$g_i(\mathbf{x}) = \frac{1}{\sigma^2}\mu_i^t \mathbf{x} - \frac{1}{2\sigma^2}\mu_i^t \mu_i.$$

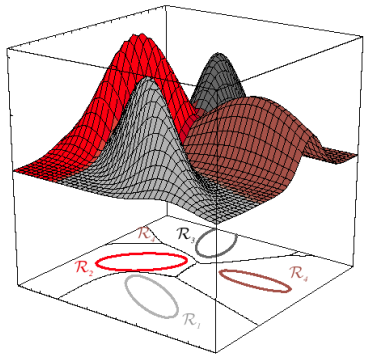
- Reduced to a linear machine.
- Classification boundary at

$$\mathbf{x} = \frac{\mu_1 + \mu_2}{2}$$



Complex Decision Boundaries

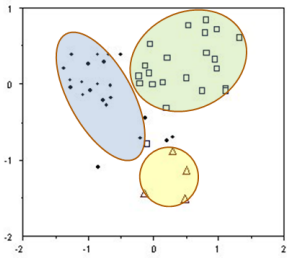
- Decision boundary in quadrature form for arbitrary Gaussian distributions.
- Question: how do we get the Gaussian distributions?



from DHS, 2nd Ed.

How Does It Work - actually

- Collect data items and their class "labels"
- Apply maximum-likelihood estimation of the Gaussian distribution for each class
- Derive the discriminant functions out of the distribution functions
- Conduct classification by assigning class labels according to the biggest discriminant function value



K-Nearest Neighbour (k-NN) Estimation

- We usually need to estimate the a posteriori probability
- Suppose we now place a cell of volume V around $\mathbf{x}$
- There are n labelled samples, of c classes
- The cell captures k samples, of which k_i belong to class i
- Then the estimate for the joint probability is

$$p(\mathbf{x}, \omega_i) = \frac{k_i/n}{V}.$$

- And a reasonable estimate for a posteriori probability is

$$P(\omega_i|\mathbf{x}) = \frac{p(\mathbf{x}, \omega_i)}{\sum_{j=1}^{c} p(\mathbf{x}, \omega_j)} = \frac{k_i}{k}$$

There may be a different way?

- Can we bypass probability estimation?
- Can we work on the discriminant function directly?
- Or, just work with exemplars?

→ Non-parametric methods

k-NN Classification Rules

- 1-NN
  - Given a test point/pattern $\mathbf{x}$, its nearest prototype is $\mathbf{x}'$.
  - Classify $\mathbf{x}$ by assigning it to the label associated with $\mathbf{x}'$.
  - Decision boundary piecewise-linear.
  - Asymptotic error rate no worse than twice of the Bayesian classifier.
- k-NN
  - Labelled prototype set D_n.
  - Given a test pattern $\mathbf{x}$, find the k nearest samples in D_n.
  - Classify by assigning it to the label most frequently represented in the k-nearest neighbours (i.e., by voting)
  - Smoother decision boundary compared with 1-NN
- Drawbacks
  - Large time complexity in classification
  - There may be ties between classes.

k-NN classifier: boost up the efficiency

- Use partial distance
- Calculate the distances using only a subset of the full data dimensions
- Construct a search tree
- Use indexed tree structure to guide the search / distance calculation process so that only nearby prototypes are involved
- Prototype editing (condensing). Here's a simple approach:
  - Initialize D with the entire data set.
  - Iterate through D: if sample x's neighbours are NOT all of the same class as that of x, mark x.
  - Remove all samples that are NOT marked from D.

k-NN Classifier Tie-Breaking

- Use odd k
- Weighted k-NN
  - Contribution by the nearest neighbours weighted by their distances from the input
  - E.g., using distance reciprocal as weights -

    Generate scores for all classes: $s(\mathbf{x}, c) = \sum_j \frac{1}{d(\mathbf{x}, \mathbf{n}_j) + \epsilon}$, with $\text{class}(\mathbf{n}_j) = c$

    Assign $\mathbf{x}$ to the class with the biggest score.

Recap

- Revise probability theory ☺
- Bayes' Law
- Prior, Likelihood, Posterior
- Bayes classifiers
- k-NN
- Alpaydin, Ch. 3, 4, 8
- DHS, Ch. 2 & 4

What's Next?

- Extending unsupervised methods?
- Working on discriminant functions directly?
- Support vector machines
- Decision trees
- ...
- "Mixture of Experts"