Nonparametric Density Estimation – k-nearest neighbor (kNN)

Different Approaches - More Detail

Pattern Classification
- Statistical Approach
  - Basic concepts:
    - Bayes decision rule (NN, LR, LDA)
    - Parametric learning (ML, DL)
  - Non-Parametric learning (KNN)
- Syntactic Approach
  - Basic concepts:
    - Distance
  - Agglomerative method
- Unsupervised
  - Basic concepts:
    - Bayesian decision rule
    - Discriminative
    - Supervised vs. unsupervised
- Unsupervised
  - Distance
  - Agglomerative method
  - k-means
  - Winner-take-all
  - Kohonen maps

Dimensionality Reduction
- Fisher’s linear discriminant (LDA)
- K-L transform (PCA)

Performance Evaluation
- ROC curve
- TP, TN, FN, FP

Stochastic Methods
- Local optimization (GD)
- Global optimization (SA, GA)

Bayes Decision Rule (Recap)

$p(x) = \frac{p(x|\omega_1)p(\omega_1)}{p(x|\omega_1)p(\omega_1) + \sum_{\omega_i=2}^{\omega} p(x|\omega_i)p(\omega_i)}$

$g(x) = -\ln p(x|\omega_1) + \ln p(\omega_1)$

Maximum Posterior Probability
- For a given $x$, if $p(\omega_1 | x) > p(\omega_2 | x)$, then $x$ belongs to class 1, otherwise, 2.

Likelihood Ratio
- If $p(x|\omega_1) > p(x|\omega_2)$ then decide $\omega_1$, that is
  $p(x|\omega_1) > p(x|\omega_2)$
  $p(x|\omega_2) < p(x|\omega_1)$

Discriminant Function
- The classifier will assign a feature vector $x$ to class $\omega_j$ if
  $g_j(x) > g_2(x)$

Three cases

Estimate Gaussian, Two-modal Gaussian
Dimensionality reduction
Performance evaluation and ROC curve
Motivation

• Estimate the density functions without the assumption that the pdf has a particular form

\[ p(x|\omega_j) = \frac{p(x|\omega_j)p(\omega_j)}{p(x)} \]

*Problem with Parzen Windows

• Discontinuous window function -> Gaussian
• The choice of h
• Still another one: fixed volume

\[ p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \phi \left( \frac{x-x_i}{h} \right) \]

kNN (k-Nearest Neighbor)

• To estimate \( p(x) \) from \( n \) samples, we can center a cell at \( x \) and let it grow until it contains \( k_n \) samples, and \( k_n \) can be some function of \( n \)
• Normally, we let

\[ k_n = \sqrt{n} \]
kNN in Classification

Given \( c \) training sets from \( c \) classes, the total number of samples is

\[ n = \sum n_c \]

* Given a point \( x \) at which we wish to determine the statistics, we find the hypersphere of volume \( V \) which just encloses \( k \) points from the combined set. If within that volume, \( k_m \) of those points belong to class \( m \), then we estimate the density for class \( m \) by

\[ p(x | \omega_m) = \frac{k_m}{n V} \]

\[ p(\omega_m | x) = \frac{n}{n V} \cdot \frac{k_m}{k} \cdot \frac{V}{n V} \]

kNN Classification Rule

The decision rule tells us to look in a neighborhood of the unknown feature vector for \( k \) samples. If within that neighborhood, more samples lie in class \( i \) than any other class, we assign the unknown as belonging to class \( i \).

Problems

What kind of distance should be used to measure “nearest”
- Euclidean metric is a reasonable measurement

Computation burden
- Massive storage burden
- Need to compute the distance from the unknown to all the neighbors
Computational Complexity of kNN

- In both space (storage space) and time (search time)
- Algorithms reducing the computational burden
  - Computing partial distances
  - Prestructuring
  - Editing the stored prototypes

Method 1 - Partial Distance

- Calculate the distance using some subset $r$ of the full $d$ dimensions. If this partial distance is too large, we don’t need to compute further
- What we know about the subspace is indicative of the full space

$$D_r(x, h) = \left( \sum (x_i - h_i)^2 \right)^{1/2}$$

Method 2 - Prestructuring

- Prestructure samples in the training set into a search tree where samples are selectively linked
- Distances are calculated between the testing sample and a few stored “root” samples, and then consider only the samples linked to it
- Closest distance no longer guaranteed.
Method 3 – Editing/Pruning/Condensing

- Eliminate “useless” samples during training
  - For example, eliminate samples that are surrounded by training points of the same category label
- Leave the decision boundaries unchanged

Discussion

- Combining three methods
- Other concerns
  - The choice of k
  - The choice of metrics

Distance (Metrics) Used by kNN

- Properties of metrics
  - Nonnegativity (D(a,b) >= 0)
  - Reflexivity (D(a,b) = 0 iff a=b)
  - Symmetry (D(a,b) = D(b,a))
  - Triangle inequality (D(a,b) + D(b,c) >= D(a,c))
- Different metrics
  - Minkowski metric
    - Manhattan distance (city block distance)
    - Euclidean distance
  - When k is inf, maximum of the projected distances onto each of the d coordinate axes
Visualizing the Distances

FIGURE 4.19. Each colored surface consists of points a distance 1.0 from the origin, measuring different values for k in the Minkowski metric; k is printed in red. Thus, the white surfaces correspond to the \( L_0 \) norm (Manhattan distance), the light gray sphere corresponds to the \( L_1 \) norm (Dischord distance), the dark gray ones correspond to the \( L_2 \) norm, and the pink has corresponds to the \( L_\infty \) norm. From Richard O. Duda, Peter E. Hart, and David G. Stork, Pattern Classification. Copyright © 2001 by John Wiley & Sons, Inc.