ECE471-571 – Pattern Recognition

Lecture 9 – Nonparametric Density Estimation – Parzen Windows

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.eecs.utk.edu/faculty/qi
Email: hqi@utk.edu

Pattern Classification

- Statistical Approach
  - Supervised
    - Bayes decision rule
    - Linear discriminant
  - Unsupervised
    - Agglomerative method
    - Mixture models
- Non-parametric learning (SNK)
- NN (Neural Network)
- Support vector machine
- Kohonen maps
- Rule learning (C4.5)

Non-Statistical Approach

- Supervised
  - Decision Tree
  - Syntactic approach
  - NN (BP)
  - Support Vector Machine
  - Deep Learning (DL)
  - Mean shift
- Unsupervised
  - K-means
  - Kohonen maps
  - Winner-takes-all

Performance Evaluation

- ROC curve (TP, TN, FN, FP)
- Cross validation
- Classifier fusion
  - Majority voting
  - NB, BKS

Stochastic Methods

- Local opt (GD)
- Global opt (SA, GA)

Dimensionality Reduction

- FLD, PCA

Review - Bayes Decision Rule

\[ p(y_1|x) \frac{p(x|y_1)}{p(y)} \]

Maximize posterior probability

- Case 1: Minimum Euclidean Distance (Linear Machine), \( \Sigma_1 = \Sigma_2 \)
- Case 2: Minimum Mahalanobis Distance (Linear Machine), \( \Sigma_1 = \Sigma_2 \)
- Case 3: Quadratic classifier, \( \Sigma_1 = \Sigma_2 \)

\[ p(x|y_1) > p(x|y_2) \]

\[ \frac{p(x|y_1)}{p(x|y_2)} > \frac{p(y_1)}{p(y_2)} \]

The classifier will assign a feature vector \( x \) to class \( y_1 \), otherwise, \( y_2 \).

\[ p(y_1|x) \frac{p(x|y_1)}{p(y)} \]

For a given \( x \), if \( p(y_1|y) > p(y_2|y) \)

then \( x \) belongs to class 1, otherwise, 2.

\[ p(x|y) \]

\[ \frac{p(x|y_1)}{p(x|y_2)} > \frac{p(y_1)}{p(y_2)} \]

\[ \frac{p(x|y_1)}{p(x|y_2)} \]

For a given \( x \), if \( p(y_1|y) > p(y_2|y) \)

then \( x \) belongs to class 1, otherwise, 2.

\[ p(y_1|x) \frac{p(x|y_1)}{p(y)} \]

The classifier will assign a feature vector \( x \) to class \( y_1 \), otherwise, \( y_2 \).

\[ p(x|y_1) > p(x|y_2) \]

\[ \frac{p(x|y_1)}{p(x|y_2)} > \frac{p(y_1)}{p(y_2)} \]

\[ \frac{p(x|y_1)}{p(x|y_2)} \]

For a given \( x \), if \( p(y_1|y) > p(y_2|y) \)

then \( x \) belongs to class 1, otherwise, 2.

\[ p(y_1|x) \frac{p(x|y_1)}{p(y)} \]

The classifier will assign a feature vector \( x \) to class \( y_1 \), otherwise, \( y_2 \).

\[ p(x|y_1) > p(x|y_2) \]

\[ \frac{p(x|y_1)}{p(x|y_2)} > \frac{p(y_1)}{p(y_2)} \]

\[ \frac{p(x|y_1)}{p(x|y_2)} \]

For a given \( x \), if \( p(y_1|y) > p(y_2|y) \)

then \( x \) belongs to class 1, otherwise, 2.

\[ p(y_1|x) \frac{p(x|y_1)}{p(y)} \]

The classifier will assign a feature vector \( x \) to class \( y_1 \), otherwise, \( y_2 \).

\[ p(x|y_1) > p(x|y_2) \]

\[ \frac{p(x|y_1)}{p(x|y_2)} > \frac{p(y_1)}{p(y_2)} \]

\[ \frac{p(x|y_1)}{p(x|y_2)} \]

For a given \( x \), if \( p(y_1|y) > p(y_2|y) \)

then \( x \) belongs to class 1, otherwise, 2.

\[ p(y_1|x) \frac{p(x|y_1)}{p(y)} \]

The classifier will assign a feature vector \( x \) to class \( y_1 \), otherwise, \( y_2 \).

\[ p(x|y_1) > p(x|y_2) \]

\[ \frac{p(x|y_1)}{p(x|y_2)} > \frac{p(y_1)}{p(y_2)} \]

\[ \frac{p(x|y_1)}{p(x|y_2)} \]

For a given \( x \), if \( p(y_1|y) > p(y_2|y) \)

then \( x \) belongs to class 1, otherwise, 2.
Motivation

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

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

*Start from Histogram

- In order to generate a reasonable representation for the density, we’d like to first “smooth” the data over cells
- The probability that a vector \( x \) will fall into a region \( R \) is \( P = \int_R p(x) dx \)
- If \( p(x) \) does not vary significantly within \( R \), then \( P = p(x)V \)
- For a training set of \( n \) samples, \( k \) of them fall into the hypervolume \( V \), we can then estimate \( p(x) \) by \( \hat{p}(x) = \frac{k/n}{V} \)

*Parzen Windows

- The density estimation at \( x \) is calculated by counting the number of samples fall within a hypercube of volume \( V_n \) centered at \( x \)
- Let \( R \) be a \( d \)-dimensional hypercube, whose edges are \( h_0 \) units long. Its volume is then \( V_n = h_0^d \)
- The window function
  \[ \psi(a) = \begin{cases} 1 & |a| < 0.5, \\ 0 & \text{otherwise} \end{cases} \]
  \[ k = \sum_{j=1}^{n} \psi \left( \frac{x - \omega_j}{h_0} \right) \]
- Therefore
  \[ \hat{p}(x) = \frac{1}{n} \sum_{j=1}^{n} \frac{\psi \left( \frac{x - \omega_j}{h_0} \right)}{V_n} \]
*Problem

- Hypercube – why should a point just inside the hypercube contribute the same as a point very near to \( x \), while a point just outside the hypercube contributes nothing?
- Use a continuous window function

*Continuous Window Function

- Univariate
  \[ \psi(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \]

- Multi-variate
  \[ p(x) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h_j} \exp\left(-\frac{1}{2} \left( \frac{x - x_j}{h_j} \right)^2 \right) \]

- Making \( \Sigma \) an identity matrix
  \[ p(x) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{h_j} \exp\left(-\frac{1}{2} \left( \frac{x - x_j}{h_j} \right)^2 \right) \]

- \( h_j \) reflects the variance (spread) of the smoothing kernel (window function) in the \( j \)th coordinate direction. If we assume the spread is equal in all directions
  \[ p(x) = \frac{1}{nh^d} \sum_{j=1}^{n} \exp\left(-\frac{1}{2n} \left( \frac{x - x_j}{h} \right)^2 \right) \]

*Comparison
*Another Problem

- How to choose $h$?
- A large $h$ will result in a great deal of smoothing and loss of resolution
- A very small $h$ will tend to degenerate the estimator into a collection of $n$ sharp peaks, each centered at a sampling point
- Solution: $h$ should depend on the number of samples. If only a few samples are available, we require a large $h$ and considerable smoothing, whereas if many points are available, we can use a smaller $h$ without the danger of degenerating into separate peaks.

*The Choice of $h$

- We make $h$ a function of $n$
  \[ h = \frac{1}{\sqrt{n}} \]

*Problem with Parzen Windows

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