PA196: Pattern Recognition

05. Nonparametric techniques

Dr. Vlad Popovici
popovici@recetox.muni.cz

RECETOX
Masaryk University, Brno

Outline

1. Density estimation
   - Histograms
   - Parzen density estimation
2. *k*-nearest neighbor estimation
3. Nearest neighbor classification rule
   - *k*−NN decision rule
   - Refinements
   - Distances

Introduction

- let $\mathbf{X}_1, \ldots, \mathbf{X}_n$ be i.i.d. $d-$dimensional random variables
- let $p(\mathbf{x})$ be their continuous distribution:

$$p(\mathbf{x}) \geq 0, \qquad \int_{\mathbb{R}^d} p(\mathbf{x})\, d\mathbf{x} = 1$$

- the problem is to estimate $p(\mathbf{x})$ i.e. find $\hat{p}(\mathbf{x})$
- Note: *a density estimate does not need to be a density itself!*; it can have negative values or infinite integral...

Desirable properties:

- asymptotical unbiasedness:

$$\mathrm{E}[\hat{p}(\mathbf{x})] \to p(\mathbf{x}) \text{ as } n \to \infty$$

- consistency:
  - mean squared error: $MSE(\hat{p}) = \mathrm{E}[(\hat{p}(\mathbf{x}) - p(\mathbf{x}))^2]$
  - $\leftrightarrow$ $MSE(\hat{p}) = \mathrm{Var}(\hat{p}) + [\mathrm{bias}(\hat{p})]^2$
  - if $MSE \to 0$ for all $\mathbf{x} \in \mathbb{R}^d$ than it is a *pointwise consistent estimator of p in the quadratic mean*
- global measure of accuracy: the mean integrated squared error (average of all possible samples):

$$MISE = \mathrm{E}\left[\int (\hat{p}(\mathbf{x}) - p(\mathbf{x}))^2 \, d\mathbf{x}\right] = \int \mathrm{E}[(\hat{p}(\mathbf{x}) - p(\mathbf{x}))^2] \, d\mathbf{x}$$

Outline

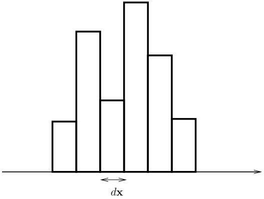
1. Density estimation
   - Histograms
   - Parzen density estimation

2. *k*-nearest neighbor estimation

3. Nearest neighbor classification rule
   - *k*−NN decision rule
   - Refinements
   - Distances

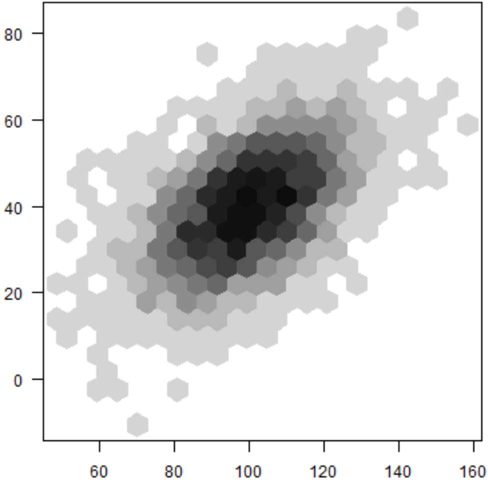
Histograms

- the simplest density estimator: divide the interval of values in N equal intervals (cells)
- $\hat{p}(x) = \frac{n_j}{\sum_{j=1}^{N} n_j dx}$ where n_j is the number of points falling into the j–th interval straddling the point x
- in d dimensions: $\hat{p}(\mathbf{x}) = \frac{n_j}{\sum_{j=1}^{N} n_j dV}$



Problems:

- exponential growth of number of cells (N^d)
- super-exponential growth in sample size needed for a proper estimation
- discontinuity between cells


80
60
40
20
0
60
80
100
120
140
160

Modifications:

- data-adaptive histograms: allow the location, size and shape of the cells to adapt to the available data
- assume variable independence (*naive Bayes*): $p(\mathbf{x}) = \prod_{i=1}^{d} p(x_i)$. For each variable one can use a histogram with N cells, which leads to $Nd \ll N^d$ cells.
- Lancaster models: assume that interactions above a certain order vanish.
- Bayesian networks:

$$p(\mathbf{x}) = p(x_d|x_1,\ldots,x_{d-1})p(x_{d-1}|x_1,\ldots,x_{d-2})p(x_2|x_1)p(x_1)$$

- dependence trees: pairwise conditional probabilities

Outline

1. Density estimation
   - Histograms
   - Parzen density estimation
2. *k*-nearest neighbor estimation
3. Nearest neighbor classification rule
   - *k*−NN decision rule
   - Refinements
   - Distances

Parzen estimator (kernel methods)

- fix the volume of the cell and use the number of point falling within to construct a density estimate
- idea: smooth the histogram with a properly selected kernel function
- the kernels are chosen to have a compact support
- the density estimate is

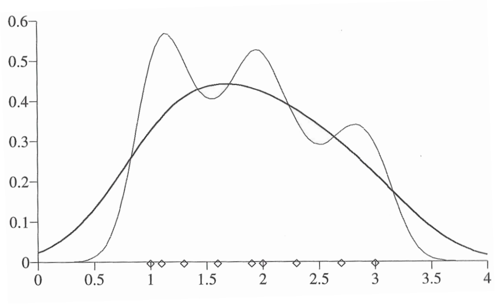
$$\hat{p}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where K is the kernel function and h is a smoothing parameter (spread, bandwidth)

Examples of kernel functions

- rectangular: $K(x) = \begin{cases} 1/2, & \text{for } |x| < 1 \\ 0, & \text{otherwise} \end{cases}$
- triangular: $K(x) = \begin{cases} 1 - |x|, & \text{for } |x| < 1 \\ 0, & \text{otherwise} \end{cases}$
- normal: $K(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$
- Bartlett-Epanechnikov:
  $K(x) = \begin{cases} \frac{3}{4}(1 - x^2/5)/\sqrt{5}, & \text{for } |x| < \sqrt{5} \\ 0, & \text{otherwise} \end{cases}$

Different levels of smoothing:



from *Webb: Statistical pattern recognition*

Outline

1. Density estimation
   - Histograms
   - Parzen density estimation

2. *k*-nearest neighbor estimation

3. Nearest neighbor classification rule
   - *k*–NN decision rule
   - Refinements
   - Distances

k–NN

- the probability that a point $\mathbf{z}$ falls into a volume V centered at $\mathbf{x}$ is

$$\theta = \int_{V(\mathbf{x})} p(\mathbf{x})\, d\mathbf{x}$$

- for a small volume, $\theta \approx p(\mathbf{x})V$
- on the other hand, $\theta \approx \frac{k(\mathbf{x})}{n}$: the fraction of points falling within V
- $\Rightarrow$ k–NN density estimator:

$$\hat{p}(\mathbf{x}) = \frac{k(\mathbf{x})}{nV}$$

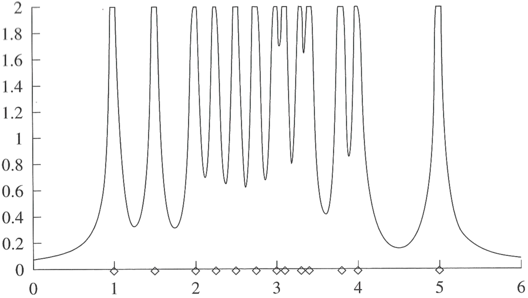
- k–NN: fix $k(\mathbf{x})/n$ or, equivalently (for a given n) fix k and find the volume V centred at containing k points

- example: if $\mathbf{x}_k$ is the k–th closest point to $\mathbf{x}$ then V can be taken as a sphere of radius $||\mathbf{x} - \mathbf{x}_k||$
- the volume of a d–dimensional sphere is

$$\frac{2r^d\pi^{\frac{d}{2}}}{d}\Gamma(d/2)$$

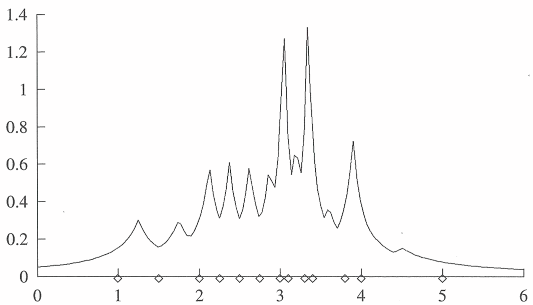
  where $\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}\,dx$ (for $n \in \mathbb{N}$, $\Gamma(n) = (n-1)!$)
- this is in contrast with the histogram, where the volume is fixed and k varies

k−NN density estimation with $k = 1$



from *Webb: Statistical pattern recognition*

k−NN density estimation with $k = 2$



from *Webb: Statistical pattern recognition*

Notes:

- the density estimate produced is not a density itself
- (the estimate varies as $1/|x|$ leading to an infinite integral)
- it is asymptotically unbiased if

$$\lim_{n\to\infty} k(n) = \infty$$

$$\lim_{n\to\infty} \frac{k(n)}{n} = 0$$

Outline

1. Density estimation
   - Histograms
   - Parzen density estimation
2. k-nearest neighbor estimation
3. Nearest neighbor classification rule
   - $k-$NN decision rule
   - Refinements
   - Distances

Outline

1. Density estimation
   - Histograms
   - Parzen density estimation

2. *k*-nearest neighbor estimation

3. Nearest neighbor classification rule
   - *k*–NN decision rule
   - Refinements
   - Distances

- k−NN can be used to estimate the density → apply MAP rule to get a classification rule
- let there be k_i samples of class g_i among the closest k samples to $\mathbf{x}$; $\sum_{i=1}^{m} k_i = k$ (m is the total number of classes)
- let n_i be the total number of samples from class g_i: $\sum_{i=1}^{m} n_i = n$
- then the estimate of the class-conditional probability is

$$\hat{p}(\mathbf{x}|g_i) = \frac{k_i}{n_i V}$$

- the estimated prior is $\hat{p}(g_i) = \frac{n_i}{n}$

k–NN decision rule

- MAP rule: assign $\mathbf{x}$ to g_i if $\hat{p}(g_i|\mathbf{x}) \geq \hat{p}(g_j|\mathbf{x})$ for all j
- from Bayes' theorem: assign $\mathbf{x}$ to g_i if

$$\frac{k_i}{n_i V}\frac{n_i}{n} \geq \frac{k_j}{n_j V}\frac{n_j}{n}$$

for all $j \neq i$

k–NN decision rule

Assign $\mathbf{x}$ to g_i if

$$k_i \geq k_j, \quad \forall j \neq i$$

What about the ties? Breaking the ties

- random assignment among classes with the same number of neighbors
- assign to the class with the closest mean vector
- assign to the most compact class
- weighted distance
- etc. etc.

Error rate for k–NN

(Cover, Hart, 1967)

$$e^* \leq e \leq e^* \left(2 - \frac{me^*}{m-1}\right)$$

where e^* is the Bayes error rate, m is the number of classes and e is the k–NN error rate
As $n \to \infty$, $e^* \leq e \leq 2e^*$.

Note on implementing k−NN:

- as n becomes large, finding the k NN incurs more computation
- various approximating algorithms, e.g. LAESA: linear approximating and eliminating search algorithm
- idea: use the properties of the metric space and reduce the number of comparisons to a set of identify "prototypes"

Outline

1. Density estimation
   - Histograms
   - Parzen density estimation

2. k-nearest neighbor estimation

3. Nearest neighbor classification rule
   - $k-$NN decision rule
   - Refinements
   - Distances

Refinements: editing techniques

Idea: remove misclassified samples to obtain homogeneous regions.
Procedure: given a set *R* and a classification rule η, let *S* be the set of misclassified samples from *R* by η. Remove these and re-train η on $R' = R \setminus S$, etc. etc

Possible implementation:

1. consider a partition of the full set into N subsets $R_1, \ldots, R_N$
2. classify samples in R_i using $k-$NN trained on the union of M "next" sets: $R_{(i+1) \mod N} \cup \cdots \cup R_{(i+M-1) \mod N}$ for $1 \leq M \leq N - 1$
3. remove the samples misclassified and repartition
4. repeat until a predefined number of iterations do not remove any more samples

Notes:

- $M = N - 1$ is similar to cross-validation
- if N is equal to number of samples, the procedure becomes leave-one-out
- the result is a set of homogeneous "clusters" of samples

Refinements: condensation

- after editing, the clusters can be "condensed"
- idea: remove samples in the center of the clusters, that do not contribute to the decision

Outline

1. Density estimation
   - Histograms
   - Parzen density estimation
2. *k*-nearest neighbor estimation
3. Nearest neighbor classification rule
   - *k*−NN decision rule
   - Refinements
   - Distances

Distance

- choice of distance depends on the (knowledge of the) domain
- is the space isotrop? are some variables "more important"? etc etc
- general Euclidean distance:

$$d(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^t \mathbf{A} (\mathbf{x} - \mathbf{z})}$$

- alternative (van der Heiden, Groen - 1997 - radar applications):

$$d(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x}^{(p)} - \mathbf{z}^{(p)})^t (\mathbf{x}^{(p)} - \mathbf{z}^{(p)})}$$

where

$$x_i^{(p)} = \begin{cases} (x_i^p - 1)/p, & \text{if } 0 < p \leq 1 \\ \log x_i, & \text{if } p = 0 \end{cases}$$

What about k?

- the larger k the more robust is the procedure; however
- k must be less than the smallest of n_i
- k can be optimized in a cross-validation approach
- Enas, Choi (1986) suggest: $k \approx n^{2/8}$ or $k \approx n^{3/8}$ where n is the sample size