Alpaydin Chp 8.

Note Title 3/30/2008

Non-Parametric Methods

We do not assume a specific model $g(x|\theta)$
Instead we assume that similar inputs have similar outputs.

Instance-based or Memory-based learning.
All (most) instances should be stored - requires $O(N)$ storage.

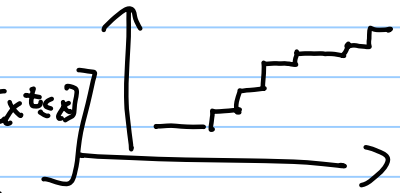
Nonparametric Density Estimation

Data $X = \{x^t\}_{t=1}^N$ i.i.d. from some unknown source $p(x)$

Cumulative distribution function (1-D)

Nonparametric estimate $\hat{F}(x) = \frac{\#[x^t \le x]}{N}$

→ Probability $\hat{p}(x) = \frac{1}{h}\left[\frac{\#[x^t \le x+h] - \#[x^t \le x]}{N}\right]$



h - length of interval (close enough)

Histogram Estimator

Input space divided into equal sized intervals named bins. Origin x_0 and bin width h.

$[x_0 + mh, x_0 + (m+1)h]$

$\hat{p}(x) = \frac{\#[x^t \text{ in same bin as } x]}{Nh}$.

Choice of bin size affects distribution.

Histogram Estimator (cont)

Naive estimator (no origin)

$$\hat{p}(x) = \frac{\#\{x-h < x^t \le x+h\}}{2Nh}$$

This can be written as:

$$\hat{p}(x) = \frac{1}{Nh}\sum_{t=1}^{N} w\left(\frac{x-x^t}{h}\right)$$

weights defined by $w(u) = \frac{1}{2}$, $|u| < 1$, $= 0$, otherwise
[jumps at $x^t \pm h$]

Kernel Estimator

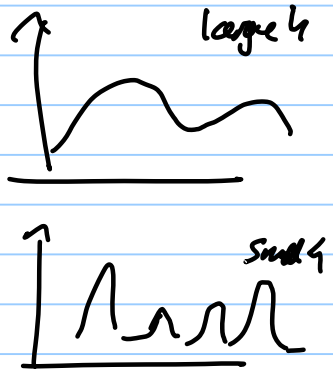
To get a smooth estimate, we use a smooth weight function - kernel function.

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$$

Kernel estimator / Parzen window

$$\hat{p}(x) = \frac{1}{Nh}\sum_{t=1}^{N} K\left(\frac{x-x^t}{h}\right)$$

(Can truncate $K(u)$ for large $|u|$)
Again, h acts as scale.



(3) k-Nearest neighbour estimator

Adapts the amount of smoothing to the local density of data.

Degree of smoothing is controlled by k, the number of neighbours

Distance $|a-b|$ between examples a, b

For each x, define

$$d_1(x) \leq d_2(x) \leq \ldots \leq d_N(x)$$

to be the distances from x to other sample points.

k-nearest neighbour is $\hat{p}(x) = \frac{k}{2Nd_k(x)}$

(Compared to Parzen – $d_h(x) = h$ – but instead of fixing h and counting samples, we fix the no. of observations!)

To get a smoother estimate, we use kernel

$$\hat{p}(x) = \frac{1}{Nd_h(x)} \sum_{t=1}^{N} K\left(\frac{x - x^t}{d_h(x)}\right)$$

like kernel with adaptive smoothing $h = d_h(x)$

(4) Generalization to Multivariate Data

$\mathcal{X} = \{\underline{x}^{(i)}\}_{i=1}^{N}$, $\hat{p}(\underline{x}) = \frac{1}{N h^d} \sum_{i=1}^{N} K\left(\frac{\underline{x} - \underline{x}^{(i)}}{h}\right)$

Typical Candidate — multivariate Gaussian. with $\int_{\mathbb{R}^d} K(\underline{x})\, d\underline{x} = 1$.

$$K(\underline{u}) = \left(\frac{1}{\sqrt{2\pi}}\right)^d e^{-\frac{1}{2}|\underline{u}|^2}$$

But non-parametric methods are dangerous in high-dimensional spaces because of the curse of dimensionality.

If $\underline{x}$ is an 8-dim
10 bins per dimension
Then require 10^8 bins

For kernels, the data will typically have local preferred directions. So the Gaussian kernel should be of form $K(\underline{u}) = \frac{1}{(2\pi)^{d/2} |\underline{\Sigma}|^{1/2}} e^{-\frac{1}{2} \underline{u}^T \underline{\Sigma}^{-1} \underline{u}}$

(5) # Nonparametric Classification

$$\hat{p}(\underline{x}|C_i) = \frac{1}{N_i h^d} \sum_{t=1}^{N} K\left(\frac{\underline{x}-\underline{x}^t}{h}\right) r_i^t$$

$r_i^t = 1$, if $\underline{x}^t \in C_i$
0, otherwise.

MLE $\hat{P}(C_i) = N_i/N$. $\quad N_i = \sum_t r_i^t$.

Discriminant is $g_i(\underline{x}) = \hat{p}(\underline{x}|C_i)\hat{P}(C_i)$

$$= \frac{1}{N h^d}\left\{\sum_{t=1}^{N} K\left(\frac{\underline{x}-\underline{x}^t}{h}\right) r_i^t\right\}$$

For special case of k-nn estimator

$$\hat{p}(\underline{x}|C_i) = \frac{k_i}{N_i V^k(\underline{x})}$$

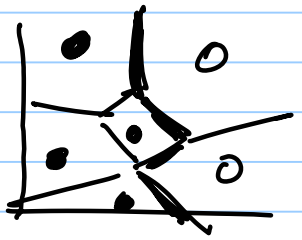
$V^k(\underline{x})$ volume containing the k-nearest neighbor

$$\hat{P}(C_i|\underline{x}) = \frac{\hat{p}(\underline{x}|C_i)\hat{P}(C_i)}{\hat{p}(\underline{x})} = \frac{k_i}{k}.$$

k-nearest neighbor classifier assigns the input to the class having most examples among the k-nearest neighbors.

(6) Condensed Nearest Neighbor

decrease the number of stored instances without degrading performance



e.g. can remove those instances ● & ○ which are not at the boundaries

Nonparametric Regression: Smoothing Models

regression $\mathcal{X} = \{x^t, r^t\}$ $\quad r^t \in \Re$.

$$r^t = g(x^t) + \epsilon.$$

Fuzzy neighborhood of x and average the r values to calculate $\hat{g}(x)$.

running mean smoother

$$\hat{g}(x) = \frac{\sum_{t=1}^{N} b(x, x^t) r^t}{\sum_{t=1}^{N} b(x, x^t)}$$

with $b(x, x^t) = 1$, if x^t in same bin as x

0, otherwise.

weighted mean smoother

$$\hat{g}(x) = \frac{\sum_{t=1}^{N} w\left(\frac{x - x^t}{h}\right) r^t}{\sum_{t=1}^{N} w\left(\frac{x - x^t}{h}\right)}$$

$w(u) = 1$ if $|u| < 1$

0 otherwise.

Kernel smoother.

$$\hat{g}(x) = \frac{\sum_t K\left(\frac{x - x^t}{h}\right) r^t}{\sum_t K\left(\frac{x - x^t}{h}\right)}$$

(7)

Choosing the Smoothing Parameter.

The choice of h (scale) or k (no. neighbours). Large h or k decreases variance but increases bias.

For smoothing splines

$$\sum_t [r^t - \hat{g}(x^t)]^2 + \lambda \int_a^b [\hat{g}''(x)]^2 dx$$

λ curvature.

Cross-validation is used to tune h, k or λ.