Nonparametric Methods

Ziping Zhao

School of Information Science and Technology
ShanghaiTech University, Shanghai, China

CS182: Introduction to Machine Learning (Spring 2023)
http://cs182.sist.shanghaitech.edu.cn

Ch. 8 of I2ML (Secs. 8.6 – 8.7 excluded)

Outline

Introduction

Nonparametric Density Estimation

Nonparametric Classification

Nonparametric Regression

Nonparametric Classification – I

- Classification based on density estimation:
  - **Step 1**: estimate the class-conditional densities $p(\mathbf{x} \mid C_i)$ (parametric or nonparametric approach).
  - **Step 2**: use Bayes' rule to compute the posterior class probabilities and make optimal decision.
- Kernel estimator of class-conditional densities:

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

where

$$r_i^t = \begin{cases} 1 & \text{if } \mathbf{x}^t \text{ is in } C_i \\ 0 & \text{otherwise} \end{cases}$$

and $N_i = \sum_t r_i^t$.

Nonparametric Classification – II

- MLE of prior probabilities:

$$\hat{p}(C_i) = \frac{N_i}{N}$$

- Discriminant functions:

$$g_i(\mathbf{x}) = \hat{p}(\mathbf{x} \mid C_i)\hat{P}(C_i) = \frac{1}{Nh^d}\sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right) r_i^t$$

  where the common factor $1/(Nh^d)$ can be ignored.

- So each training instance votes for its class and has no effect on other classes; the weight of vote is given by the kernel function $K(\cdot)$, typically giving more weight to closer instances.

k-NN Classifier – I

- *k-NN estimator*:

$$\hat{p}(\mathbf{x} \mid C_i) = \frac{k_i}{N_i V_k(\mathbf{x})}$$

where

  - k_i is the number of neighbors that belong to C_i
  - $V_k(\mathbf{x})$ is the volume of the d-dimensional hypersphere centered at $\mathbf{x}$ with radius $r_k = \|\mathbf{x} - \mathbf{x}^{(k)}\|$ where $\mathbf{x}^{(k)}$ is the k-th nearest observation to $\mathbf{x}$ (among all neighbors from all classes of $\mathbf{x}$). $V_k(\mathbf{x}) = r_k^d c_d$ with c_d is the volume of the unit sphere in d dimensions, for example, $c_1 = 2$, $c_2 = \pi$, $c_3 = 4\pi/3$, and so forth.

k-NN Classifier – II

- Posterior class probabilities:

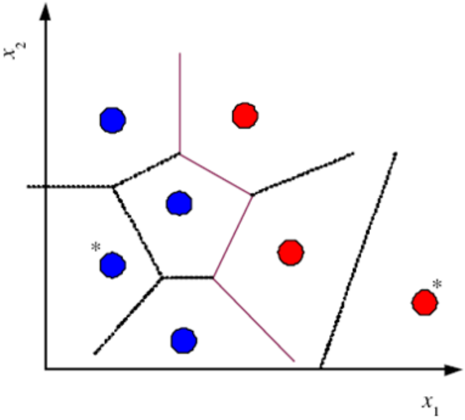
$$\hat{P}(C_i \mid \mathbf{x}) = \frac{\hat{p}(\mathbf{x} \mid C_i)\hat{P}(C_i)}{\sum_j \hat{p}(\mathbf{x} \mid C_j)\hat{P}(C_j)} = \frac{k_i/NV_k(\mathbf{x})}{\sum_j k_j/NV_k(\mathbf{x})} = \frac{k_i}{k}$$

- k-NN classifier: assigns the input $\mathbf{x}$ to the class C_i having most examples among the k neighbors of $\mathbf{x}$, i.e.,

$$i = \arg\max_j \hat{P}(C_j \mid \mathbf{x}) = \arg\max_j k_j$$

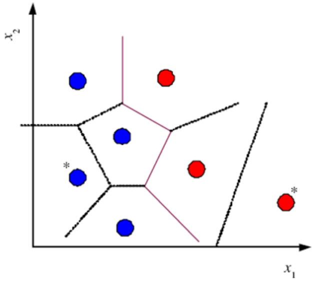
Nearest Neighbor Classifier

- Nearest neighbor classifier: special case of k-NN classier with $k = 1$.
- Voronoi tessellation formed in input space:



Condensed Nearest Neighbor

- Time/space complexity of nonparametric methods (e.g., k-NN): $O(N)$
- Condensing methods: find a small (hopefully smallest) subset $\mathcal{Z}$ of $\mathcal{X}$ such that the error does not increase when $\mathcal{Z}$ is used in place of $\mathcal{X}$.
- Condensed nearest neighbor classier: only the instances that define the discriminant need to be kept but those inside the class regions can be removed (cf. support vector machines).



Outline

Nonparametric Regression

- Nonparametric regression is a.k.a. smoothing models.
- Regression problem:

$$r^t = g(\mathbf{x}^t) + \epsilon$$

  where $r^t \in \mathbb{R}$.
- Nonparametric regression is needed when we cannot find an appropriate parametric model (e.g., polynomial) for $g(\cdot)$.
- Nonparametric regression estimators (a.k.a. smoothers):
  - Running mean smoother
  - Kernel smoother
  - Running line smoother
- Here we consider the univariate case, which can be extended easily to the multivariate case.

Regressogram

- Regressogram:

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

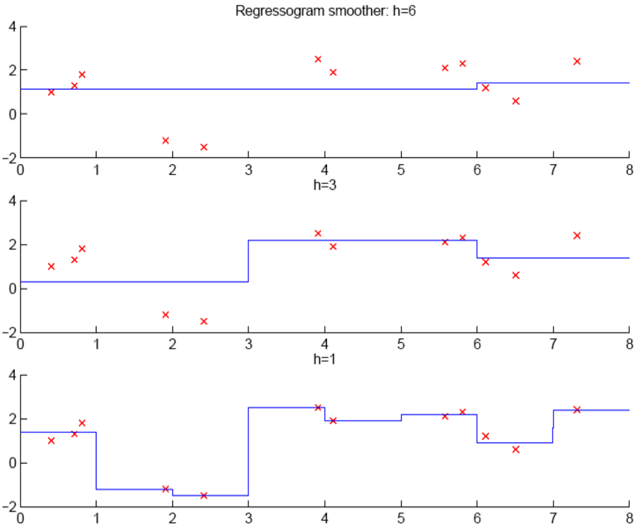
where

$$b(x, x^t) = \begin{cases} 1 & \text{if } x^t \text{ is in the same bin with } x \\ 0 & \text{otherwise} \end{cases}$$

- It can be written as

$$\underset{g(x)}{\text{minimize}} \quad \sum_{t=1}^{N} b(x, x^t) \| r^t - g(x) \|_2^2$$

Regressogram with Different Bin Lengths



Running Mean Smoother

- To avoid the need to fix an origin, the running mean smoother (or bin smoother) defines a bin symmetric around x:

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

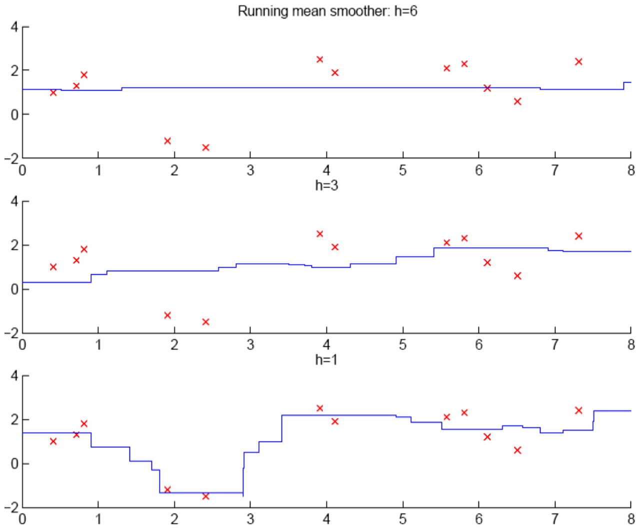
where

$$w(u) = \begin{cases} 1 & \text{if } |u| < 1 \\ 0 & \text{otherwise} \end{cases}$$

- It can be written as

$$\underset{g(x)}{\text{minimize}} \quad \sum_{t=1}^{N} w(\frac{x-x^t}{h}) \|r^t - g(x)\|_2^2$$

Running Mean Smoother with Different Bin Lengths



Kernel Smoother

- Kernel smoother:

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

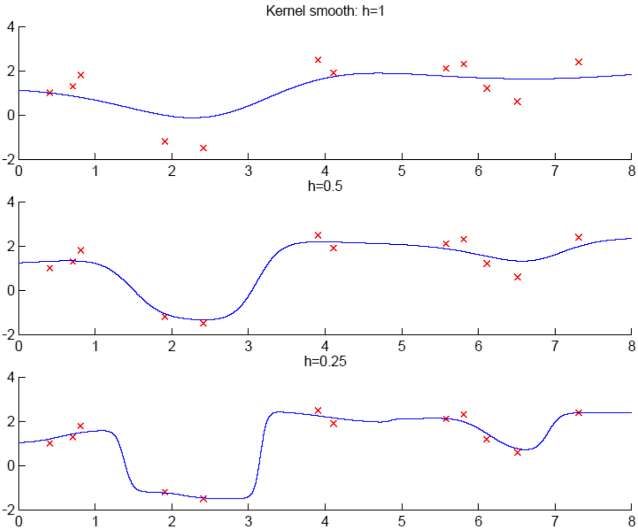
  where $K(\cdot)$ is a kernel, such as Gaussian kernel, that gives less weight to further points.

- It can be written as

$$\underset{g(x)}{\text{minimize}} \quad \sum_{t=1}^{N} K(\frac{x-x^t}{h}) \| r^t - g(x) \|_2^2$$

- k-NN smoother: Instead of fixing h, the number of neighbors k is fixed to adapt to the density around x.

Kernel Smoother with Different Bin Lengths



Running Line Smoother

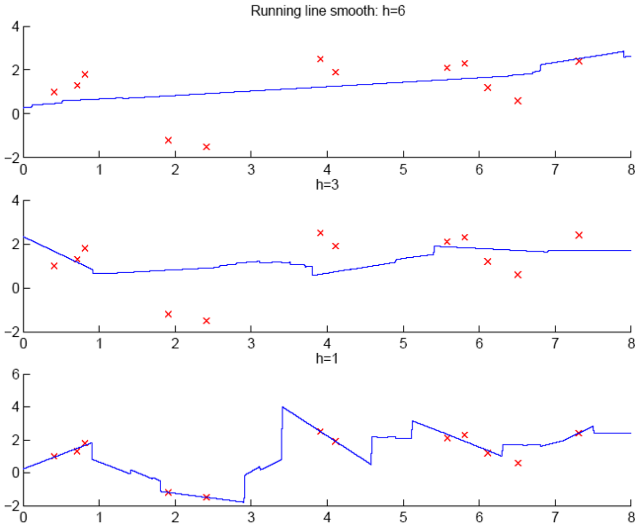
- Unlike the running mean smoother which has discontinuities, the running line smoother uses continuous piecewise linear fit.
- We can use larger bins than running mean smoother because fitting lines provide slightly more flexibility.
- It can be written as

$$\underset{g(x)=a_x x+b_x}{\text{minimize}} \quad \sum_{t=1}^{N} w\left(\frac{x-x^t}{h}\right)\|r^t-(a_x x^t+b_x)\|_2^2$$

  which is a weighted least squares (or weighted linear regression).
- Alternatively, kernel weighting $K(x, x^t)$ may also be used to give the locally weighted running line smoother, a.k.a. locally estimated scatterplot smoothing (loess), which is given by

$$\underset{g(x)=a_x x+b_x}{\text{minimize}} \quad \sum_{t=1}^{N} K\left(\frac{x-x^t}{h}\right)\|r^t-(a_x x^t+b_x)\|_2^2$$

Running Line Smoother with Different Bin Lengths



How to Choose h or k?

- Small h or k (undersmoothing): small bias but large variance.
- Large h or k (oversmoothing): large bias but small variance.
- Regularized cost function for smoothing splines:

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

  - First term: error of fit
  - Second term: penalty for high variability, where $\hat{g}''(x)$ is the curvature of $\hat{g}(\cdot)$ and $[a, b]$ is the input range
  - λ: trades off error and variability and can also be determined by cross-validation.
- Cross-validation may be used to determine the best h or k.