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

Outline

Parametric, Semiparametric, and Nonparametric Methods

- Parametric:
  - $p(\mathbf{x} \mid C_i)$ is represented by a single global parametric model.
  - Topic 3 (Parameter Estimation for Generative Models)
- Semiparametric:
  - $p(\mathbf{x} \mid C_i)$ is represented by a small number of local parametric models.
  - Topic 10 (Clustering and Mixture Models)
- Nonparametric:
  - $p(\mathbf{x} \mid C_i)$ cannot be represented by a single parametric model or a mixture model; the data speaks for itself.
  - Assumption: similar inputs have similar outputs, i.e., smooth functions (e.g., probability density functions, discriminant functions, regression functions).
  - Given a test instance, find a small number of nearest (or most similar) training instances and interpolate from them.
  - A.k.a. instance-based, memory-based, case-based, or lazy learning algorithms.

Outline

Why We Need Nonparametric Density Estimation?

- Common parametric forms rarely fit the densities encountered in practice.
- Classical parametric densities are unimodal, whereas many practical problems involve multimodal densities.
- Non-parametric procedures can be used with arbitrary distributions and without the assumption that the form of the underlying densities are known.

Nonparametric Density Estimation: Univariate Case

- Sample $\mathcal{X} = \{x^t\}_{t=1}^N$, drawn i.i.d. from some unknown probability density $p(x)$, with cumulative distribution function $F(x)$.
- Estimator $\hat{F}(x)$ for $F(x)$:

$$\hat{F}(x) = \frac{\#\{x^t \leq x\}}{N}$$

- Estimator $\hat{p}(x)$ for $p(x)$:

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

  where h is the length of the interval and instances x^t that fall in this interval are assumed to be "close enough".
- The techniques given in this lecture are variants where different heuristics are used to determine the instances that are close and their effects on the estimate.

Histogram Estimator

- The input space is divided into equal-sized intervals called bins or boxes:

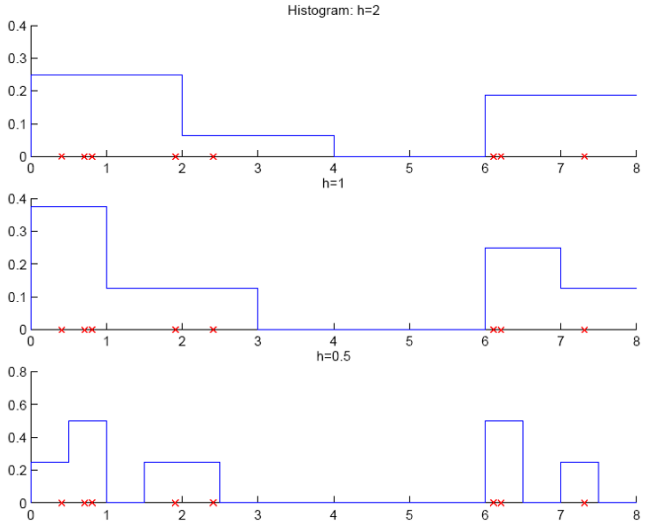
$$\left[x_0 + mh, x_0 + (m+1)h\right)$$

  where x_0 is the origin, h is the bin width or volume, and m is an integer.
- Histogram estimator:

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

- Once the bin estimates are calculated and stored, we do not need to retain the training set.
- The bin width and the starting position are "parameters". The choice of bin width is critical since it plays the role of a smoothing parameter.

Histogram Estimators with Different Bin Sizes



Naive Estimator

- Unlike the histogram estimator, this estimator frees us from setting an origin.
- Naive estimator:

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

  The bin is of size h and x is always at its center.
- Alternative form of estimator:

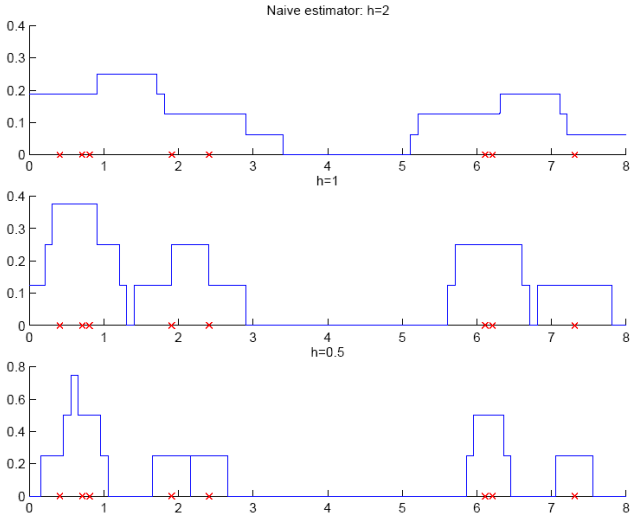
$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^{N} w\Big(\frac{x - x^t}{h}\Big)$$

  with weight function:

$$w(u) = \begin{cases} 1 & \text{if } -1/2 < u \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

- Each x^t has a symmetric region of influence of size h around it and contributes 1 for an x falling in its region. The nonparametric estimate is the sum of influences of x^t whose regions include x, i.e., sum of "boxes."

Naive Estimators with Different Bin Sizes



Kernel Estimator

- Histogram estimator and naive estimator are not continuous at bin boundaries.
- To get a smooth estimator, a smooth weight function called kernel function is used, e.g., Gaussian kernel (typically used for its continuity and differentiability):

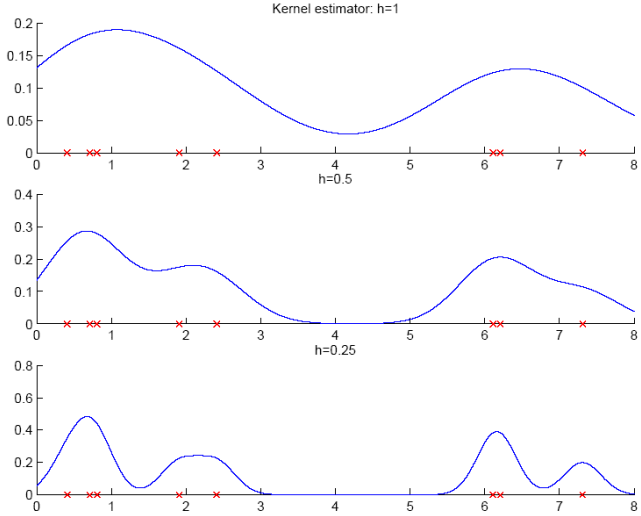
$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$$

- Kernel estimator (a.k.a. Parzen windows):

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

  where $K(\cdot)$ determines the shape of the influences and h determines the width or spread.
- For $\hat{p}(x)$ to be a density, $K(\cdot)$ should be nonnegative and integrates to 1.
- It is a sum of N smooth local functions. The choice of bin width is critical.

Kernel Estimators with Different Window Widths



Properties of Kernel Estimator

- All the x^t have an effect on the estimate at x and this effect decreases smoothly as $|x - x^t|$ increases.
- When h is small, each training instance has a large effect in a small region.
- When h is large, there is more overlap of the kernels and the estimator is smoother.
- One problem with this estimator is that the window width h is fixed across the entire input space.

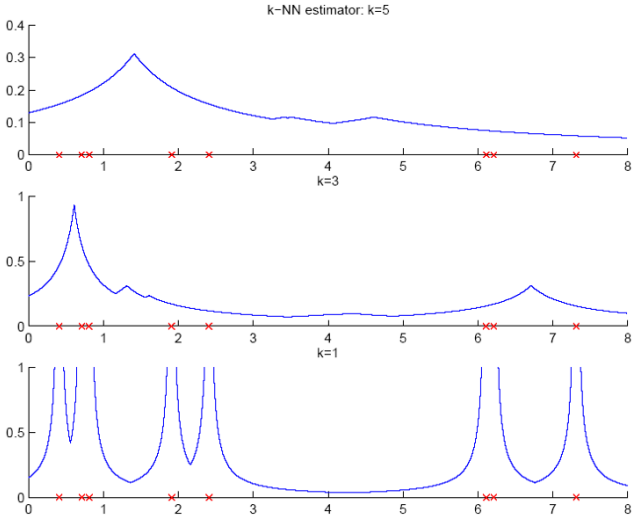
k-Nearest Neighbor Estimator

- While kernel estimator uses the same window width everywhere, the nearest neighbor class of estimators adapts the amount of smoothing to the local density of data.
- The degree of smoothing is controlled by $k(\ll N)$, the number of neighbors taken into account. k plays a similar role as the parameter h in kernel estimators.
- k-nearest neighbor (k-NN) estimator:

$$\hat{p}(x) = \frac{k}{2Nd_k(x)}$$

  where $d_k(x)$ is the distance from x to the kth nearest instance.
- This is like a naive estimator with $h = 2d_k(x)$, the difference being that instead of fixing h and checking how many samples fall in the bin, we fix k, the number of observations to fall in the bin, and compute the bin size.
- When the data density is high, the bins are small; when it is low, the bins are larger.

k-Nearest Neighbor Estimators with Different k Values



k-Nearest Neighbor Estimator with a Kernel Function

- The k-NN estimator is not continuous and hence is not a probability density function since it integrates to ∞, not 1.
- k-nearest neighbor (k-NN) estimator with a kernel function:

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

  where $K(\cdot)$ is typically chosen to be the Gaussian kernel.
- This estimator is like a kernel estimator with adaptive smoothing parameter $h = d_k(x)$.

Multivariate Kernel Estimator – I

- A sample of d-dimensional observations $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^N$
- Multivariate kernel density estimator:

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

with the requirement that

$$\int_{\mathbb{R}^d} K(\mathbf{x}) d\mathbf{x} = 1$$

- Multivariate Gaussian kernel:

$$K(\mathbf{u}) = \Big(\frac{1}{\sqrt{2\pi}}\Big)^d \exp\Big(-\frac{\|\mathbf{u}\|^2}{2}\Big)$$

Multivariate Kernel Estimator – II

- Multivariate kernel as product of the same univariate kernel function with different bandwidths for each dimension:

$$\hat{p}(\mathbf{x}) = \frac{1}{N}\sum_{t=1}^{N}\prod_{i=1}^{d}\frac{1}{h_i}K\Big(\frac{x_i - x_i^t}{h_i}\Big)$$

- Instead of using a single smoothing parameter h for all dimensions which corresponds to using the Euclidean distance, generalization to Mahalanobis distance gives the multivariate ellipsoidal Gaussian kernel:

$$K(\mathbf{u}) = \frac{1}{(2\pi)^{d/2}|\mathbf{S}|^{1/2}}\exp\Big(-\frac{1}{2}\mathbf{u}^T\mathbf{S}^{-1}\mathbf{u}\Big)$$

  where $\mathbf{S}$ is the (general) sample covariance matrix.
- Curse of dimensionality: nonparametric estimation in high-dimensional spaces may require many bins (grows exponentially with data dimensionality d), most of which end up being empty.

Nonparametric Density Estimation Comparisons

- Histogram estimators do not require storage for all the observations, they require storage for the description of the bins. But for simple histograms the number of the bins grows exponentially with the dimension of the observation space.
- Kernal estimators require storage of all observations and N evaluations of the kernel function for each estimate, which is computationally expensive!
- k-NN estimators require the storage of all the observations.

Nonparametric Density Estimation Summary

- Advantages
  - Generality: same procedure for unimodal, normal and bimodal mixture.
  - No assumption about the distribution required ahead of time.
  - With enough samples we can converge to an arbitrarily complicated target density.
- Disadvantages
  - Number of required samples may be very large (much larger than would be required if we knew the form of the unknown density) .
  - Curse of dimensionality.
  - In case of kernal and k-NN, computationally expensive (storage & processing).
  - Sensitivity to choice of bin size, bandwidth,...

Outline