Nonparametric Methods

CAI 5107: Machine Learning

Instructor: Anowarul Kabir

Email: akabir@usf.edu

Fall 2025

Parametric vs. Nonparametric methods

- **Parametric**: Data is drawn from one or mixture of probability distributions
  - No such assumption for nonparametric
  - We assume a single model that might be valid over the whole input space
    - However, this assumption might not true.
  - we are interested in one global solution.

- **Nonparametric**: Assume similar inputs have similar outputs
  - Similar instances mean similar things
  - In nonparametric, there is no single global model. Local models are estimated as they needed, affected only by the nearby training instances.

Nonparametric methods

- **Algorithmic framework**: Find similar instances from the training set using a suitable distance measurement and interpolating from them to find the right output.

- Also known as instance-based or memory-based learning algorithms
  - Might need to store instances into lookup table and interpolate from them
  - Memory requirements $O(N)$ where N is the number of instances in the training set.

– High density at x: a lot of data points sit near x

Nonparametric density estimation

- The nonparametric estimator for cumulative density function (CDF), $F(x)$, at point x is

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

look at all data points x^t
count how many of them $\leq x$
Divided by total # of sample

- The nonparametric estimator for density function (derivative of the cumulative distribution) → PDF

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

- h is the length of the interval and instances x^t that fall in this interval are assumed to be "close enough"

points that fall within the interval $[x, x+h]$ are treated as "close enough" to estimate the density at x

Dataset $x^1, x^2, x^3, \ldots, x^N$

→ Pick a value x

- what fraction of sample are $\leq x$ → CDF
- How many samples fall near x ? → Density

Histogram estimator

non parametric estimator of the probability density function (PDF)

N: total # samples

h: width of the bin

m: (0, 1, 2) bin number

- Given an origin x_0 and a bin width h, the histogram estimator can be defined as

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

  - where the bins are the intervals $[x_o + mh, x_o + (m+1)h)$

Bin 0 $[x_0, x_0 + h]$

Bin 1 $[x_0 + h, x_0 + 2h]$

...

  - Bin width has the most impact on this estimate

h ← large h → bins are wide → density too smooth → underfitting

- **Advantage**: Once the bins are computed, histogram estimate does not require to save the training set.

↳ memory efficient: can throw away the training set once we've counted how many points fall in each bin

Kernel estimator

- The kernel estimator is defined as

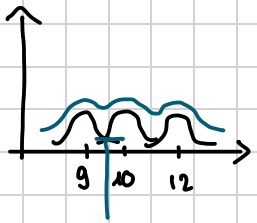
$$\hat{p}(x) = \frac{1}{Nh} \sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right)$$

  - where $K(\cdot)$ can be defined as the Gaussian kernel

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

  - h is the window width and $K(\cdot)$ determines the shape of the influences
- Note: All the training instances have an effect on the estimate of x. And this effect decreases smoothly as $|x - x^t|$ increases.

Idea : each data point lends some local density



← normal distribution for each data point

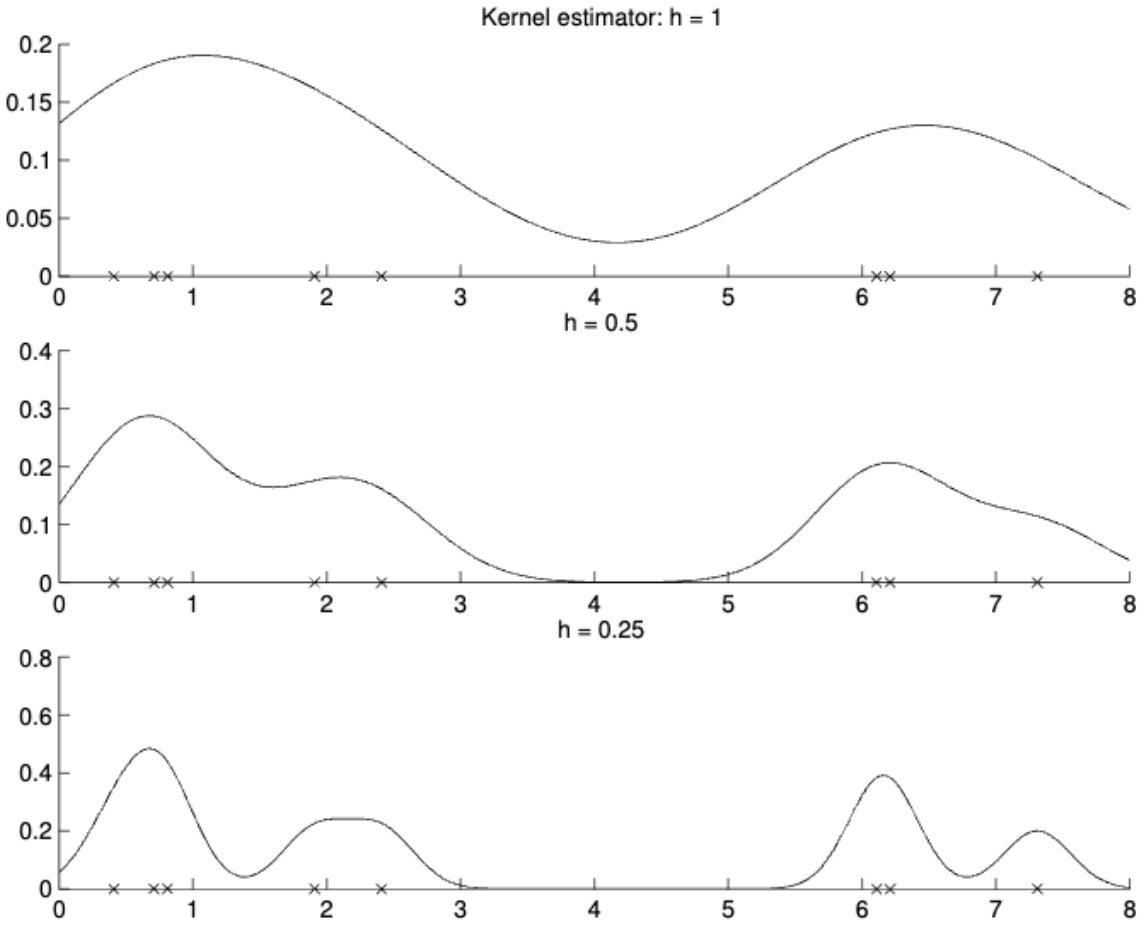
for ex, we want to find the probability density of a 9.5 lbs fish, it will be the sum of 3 probability density functions

- So we're averaging these three probability densities together to get the overall kernal density estimate (overall probability density estimate) for all these different values in between that we did not explicitly observe.
- For example : What is the probability density of seeing a 9.5 lbs fish ?

$$f_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - x_i)$$

- $f_h(x)$: x = 9.5
- $\sum_{i=1}^{n}$: iterating over points; sum all the individual kernel
- $K_h(x - x_i)$: h = kernel centered at x; x_i = each data point in the sample ↳ each of the distribution of the data point

Assume:
- K = normal
- h : band width (narrowness / wideness of each of these individual normal distributions)



over smooth

Smooth

over smooth

Figure 8.3 Kernel estimate for various bin lengths.

K-nearest neighbor (KNN) estimator

- K-nn density estimate is defined as

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

  - where $d_k(x)$ is the distance of the $k-th$ closest sample from x

- k: # of nearest neighbors
- $d_k(x)$: distance from x to its k-th closest data point
- for example, k = 3 → we will sort the distance then choose the 3rd one.

- Note: K-nn is not a probability density function since it integrates to inf, not 1. (does not sum to 1)
- To get a smoother estimate, one can use a kernel function whose effect decreases with increasing distance

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

→ give closer points more influence and farther point less

↘ It's KDE with an adaptive bandwidth ↓ $d_k(x)$

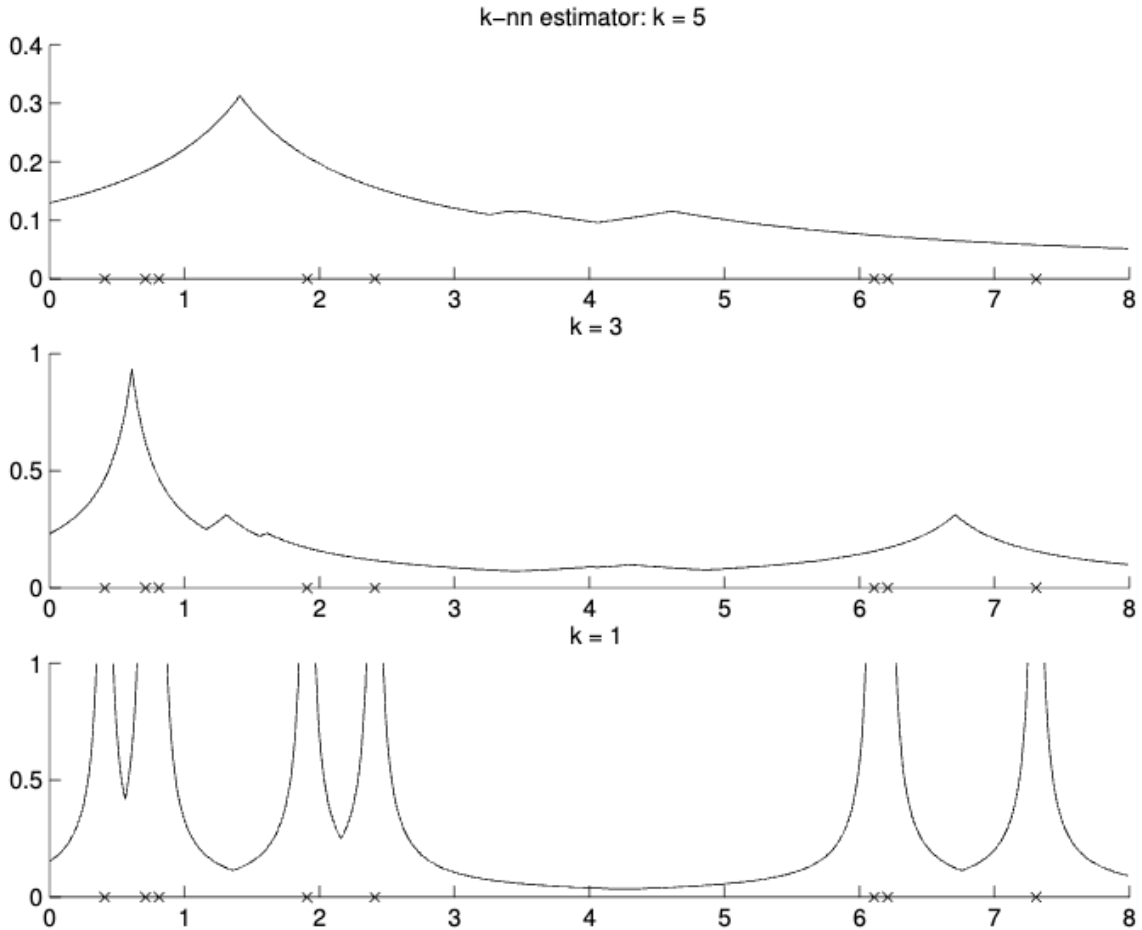


Figure 8.4 *k*-nearest neighbor estimate for various *k* values.

Nonparametric classification

- The estimate of class conditional densities, $p(x|C_i)$, can be defined using the kernel estimator

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

r_i^t : label $= \begin{cases} 1, \text{ if } x^t \in C_i \\ 0, \text{ otherwise} \end{cases}$

N_i : # of labeled instances belonging to C_i

  - Where r_i is an indicator function with 1 and 0, N_i is the number of instances belonging to C_i

- The MLE of prior is: $\hat{P}(C_i) = N_i / N$

- So, the discriminant is:

Bayes' rule

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

- x is assigned to the $i-th$ class with $max\ g_i\ (x)$

K-nn classifier

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

+ k_i : # of neighbors out of k-nearest neighbors that belong to class C_i
+ $V^k(x)$: volume of the d- dimension hypotheses centered at x and of radius $r = \|x - x_k\|$
x_k - k-th neareast neighbor.

- It assigns the input to the class having most examples among the k-neighbors of the input.

- All neighbors have equal vote, and the class having the maximum number of voters among the k neighbors is chosen.

$$\hat{p}(C_i|x) = \frac{\hat{p}(x|C_i) \cdot \hat{p}(C_i)}{\hat{p}(x)} = \frac{k_i}{k}$$

→ assign input to the class that has most example among the k neighbors of that input.

Nonparametric outlier detection

- Outlier, novelty or anomaly is an instance that is very much different from other instances in the sample.

- Outliers may indicate abnormal behavior of the system.

- **One-class classification**: Since the number of outlier examples are very small in the training set, it can be modeled as one-class or other-class classification.

non parametric case: find instances far away from other instances.

Nonparametric outlier detection

- In nonparametric density estimation, the estimated probability is high where there are many training instances nearby, and the probability decreases as the neighborhood becomes more sparse. + low probability -> outliers

- **Local outlier factor**: It compares the denseness of the neighborhood of an instance with the average denseness of the neighborhoods of its neighbors.

  ↳ a point is more isolated, further away from its neighbors

Nonparametric regression: Smoothing models

- Regression is defined as: $r^t = g(x^t) + \epsilon$

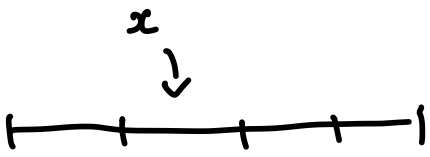
$X = \{x^t, r^t\}_{t=1}^N$

$r^t \in \mathbb{R}$; $r^t = g(x^t) + \epsilon$

- **In parametric regression**, we assume that certain order polynomials (with coefficients) will minimize the error on training set. → we have a polynomial of a certain order and we learn the coefficients in a way that we minimize sum of the squared errors
- **In nonparametric**, we assume that similar x has similar g(x) values

- nonparametric regression is also called smoother

Regressorgram new data x. We have bins



- we check which bin x is in, then average r values of all the data points in that bin

- **Generic formulation**: For given x, our approach is to find the neighborhood of x and average their r values in the neighborhood as the estimation.

- **Regressorgram**: $\hat{g}(x) = \frac{\sum_{t=1}^{N} b(x, x^t) r^t}{\sum_{t=1}^{N} b(x, x^t)}$ → sum of values in the bin → how many data points in the bin

where

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

rather than splitting into bins from the very beginning. We're gonna take a neighborhood around of the point x, neighborhood of width h

- **Kernel smoother**: $\hat{g}(x) = \frac{\sum_t K\left(\frac{x-x^t}{h}\right) r^t}{\sum_t K\left(\frac{x-x^t}{h}\right)}$ $K(\ )$ is Gaussian
  - Note: Kernel smoother gives less weights to the further points.

- **K-nn smoother**: Instead of fixing h, we can fix k, the number of neighbors, adapting the estimate to the density around x.

- Reading materials:
  - Chapter 8.1, 8.2, 8.4, 8.7, 8.8
  - Optional: 8.6