COSC 522 – Machine Learning

Baysian Decision Theory – Nonparametric Learning

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.eecs.utk.edu/faculty/qi
Email: hqi@utk.edu

Questions

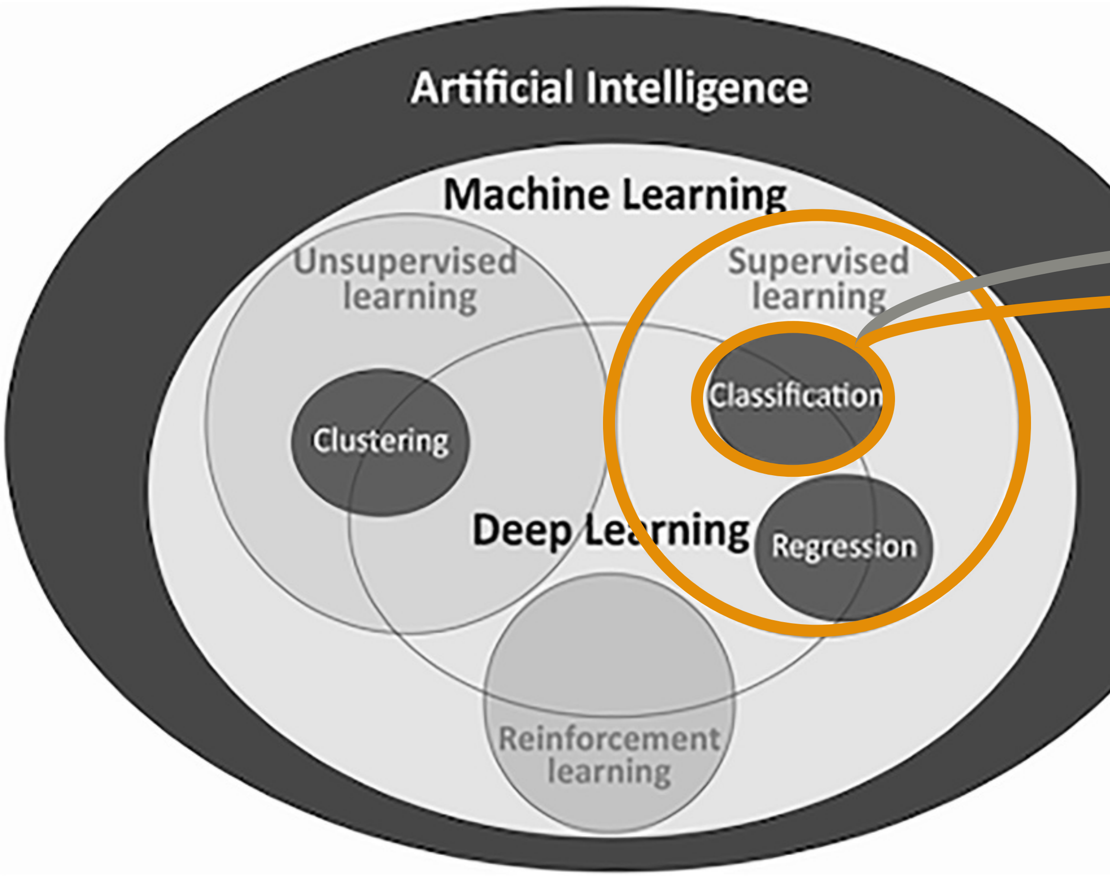
- In general, what is non-parametric learning?
- Under what conditions that non-parametric learning would be preferred?

- What is parzen window and what are the potential issues?

- What is kNN intuitively?
- Does kNN also follow the MPP decision rule?
- What is the decision boundary of kNN?
- When k is fixed, is the radius of neighborhood fixed?
- Is 1NN the same as minimum distance classifier?

- What is the cost function of kNN? What is the optimization approach used? Is kNN optimal in Baysian sense?
- What are the potential issues with kNN?

Where Are We?



	Part 1: Statistical Methods
	Baysian Learning
08/20 (T)	Introduction
08/22 (R)	Baysian Decision Theory and Parametric Learning
08/27 (T)	Non-Parametric Learning
08/29 (R)	ML with Python (taught by TA)
09/03 (T)	Recap
09/05 (R)	Homework and Project Discussion (taught by TA)
	Neural Networks
09/10 (T)	Biological Neuron and Perceptron
09/12 (R)	Back Propagation and Gradient Descent
09/17 (T)	Kernel Methods
09/19 (R)	Support Vector Machine
09/24 (T)	SVM

M. Mafu, "Advances in artificial intelligence and machine learning for quantum communication applications," IET Quantum Communication, 2024, DOI: 10.1049/qtc2.12094

Motivation

- Estimate the density functions without the assumption that the pdf has a particular form

$$P(w_j|x) = \frac{p(x|w_j)P(w_j)}{p(x)}$$

Revisit pdf Estimation

- Probability and pdf (the probability that a vector *x* fall within region *R*)

$$P = \int_R p(x')\,dx'$$

- If *p(x)* does NOT vary significantly within *R*, then

$$P = p(x)\,V$$

- For a training set of *n* samples, *k* of them fall within the hypervolume *V*, we can then estimate *p(x)* by

$$p(x) = \frac{P}{V} \approx p_n(x) = \frac{k_n/n}{V}$$

THE UNIVERSITY OF TENNESSEE KNOXVILLE

PART I: PARZEN WINDOWS

Parzen Windows

$$p_n(x) = \frac{k_n / n}{V}$$

- The density estimation at *x* is calculated by counting the number of samples fall within a hypercube of volume *V* centered at *x*
- Let *R* be a *d*-dimensional hypercube, whose edges are *h* units long. Its volume is then $V=h^d$
- Introducing the "window" function

$$\varphi(u) = \begin{cases} 1 & |u_j| \leq 0.5 \quad j=1,\ldots,d \\ 0 & otherwise \end{cases}$$

- Calculate k_n

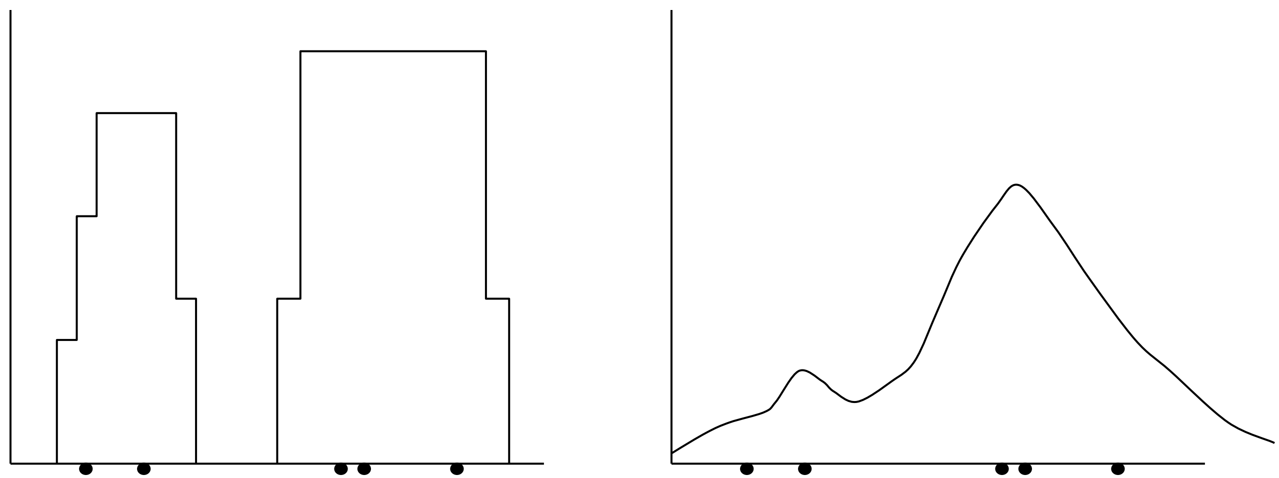
$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{x - x_i}{h}\right)$$

- Hence

$$\boxed{p_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{\varphi\left(\frac{x - x_i}{h}\right)}{V}}$$

Problems

- Hypercube – why should a point just inside the hypercube contribute the same as a point very near to **x**, while a point just outside the hypercube contributes nothing?
- Use a continuous window function



Another Problem

AICIP RESEARCH

- How to choose *h*?
- A large *h* will result in a great deal of smoothing and loss of resolution
- A very small *h* will tend to degenerate the estimator into a collection of *n* sharp peaks, each centered at a sampling point
- Solution: *h* should depend on the number of samples. If only a few samples are available, we require a large *h* and considerable smoothing, whereas if many points are available, we can use a smaller *h* without the danger of degenerating into separate peaks.

$$h = \frac{1}{\sqrt{n}}$$

Problem with Parzen Windows

- Discontinuous window function → Continuous (i.e., Gaussian)
- The choice of *h*

$$h = \frac{1}{\sqrt{n}}$$

- Still another one: Fixed volume

PART II: K-NEAREST NEIGHBOR

The k-nearest neighbor (kNN) Decision Rule - Intuitively

- The decision rule tells us to look in a neighborhood of the unknown test sample for *k* samples. If within that neighborhood, more training samples lie in class *i* than any other class, we assign the unknown as belonging to class *i*.

kNN in Classification

$$p_n(x) = \frac{k_n/n}{V}$$

- Given *c* training sets from *c* classes, the total number of samples is

$$n = \sum_{m=1}^{c} n_m$$

- Given a point **x** at which we wish to determine the statistics, we find the hypersphere of volume ***V*** which just encloses *k* points from the combined set. If within that volume, k_m of those points belong to class *m*, then we estimate the density for class *m* by

$$p\left(x|w_m\right) = \frac{k_m/n_m}{V} \qquad P\left(w_m\right) = \frac{n_m}{n} \qquad p(x) = \frac{k/n}{V}$$

kNN Classification Rule

$$P(\omega_m \mid x) = \frac{p(x \mid \omega_m) P(\omega_m)}{p(x)} = \frac{\dfrac{k_m}{n_m V} \dfrac{n_m}{n}}{\dfrac{k}{nV}} = \frac{k_m}{k}$$

- The decision rule tells us to look in a neighborhood of the unknown feature vector for *k* samples. If within that neighborhood, more samples lie in class *i* than any other class, we assign the unknown as belonging to class *i*.

kNN Decision Boundary

AICIP RESEARCH

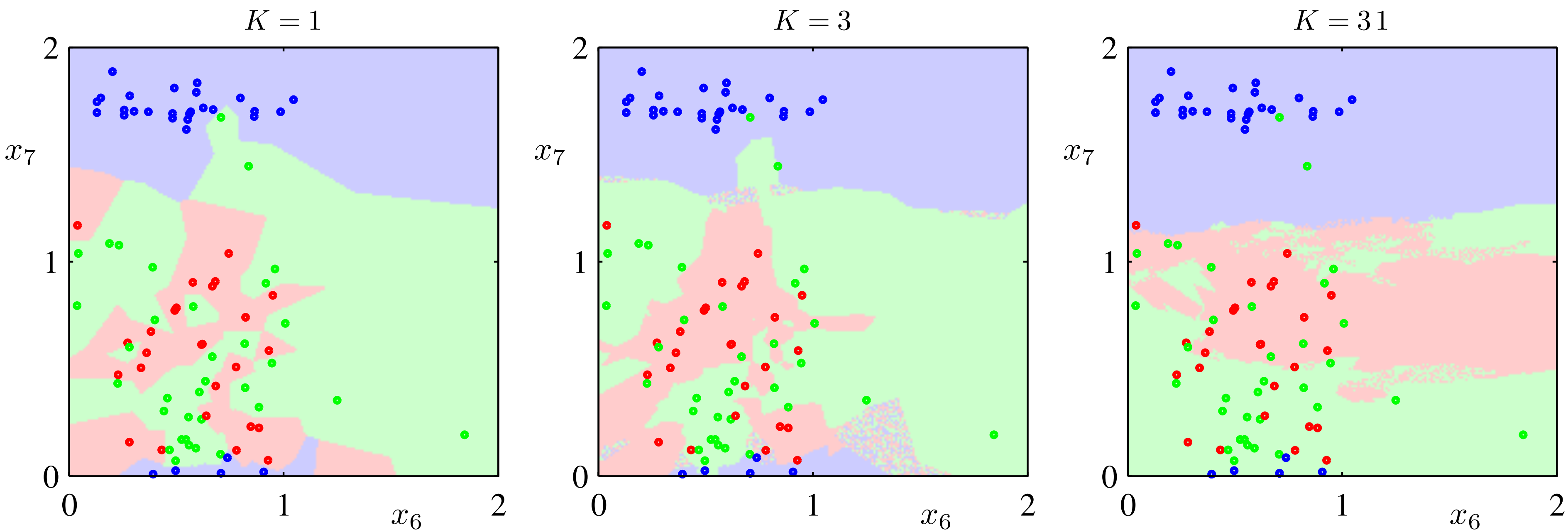


Figure 2.28 Plot of 200 data points from the oil data set showing values of x_6 plotted against x_7, where the red, green, and blue points correspond to the 'laminar', 'annular', and 'homogeneous' classes, respectively. Also shown are the classifications of the input space given by the K-nearest-neighbour algorithm for various values of K.

From [Bishop 2006]

Potential Issues

- What is a good value of “k”? $k_n = \sqrt{n}$
- What kind of distance should be used to measure “nearest”
  – Euclidean metric is a reasonable measurement
- Computation burden
  – Massive storage burden
  – Need to compute the distance from the unknown to all the neighbors

Questions

- In general, what is non-parametric learning?
- Under what conditions that non-parametric learning would be preferred?

- What is parzen window and what are the potential issues?

- What is kNN intuitively?
- Does kNN also follow the MPP decision rule?
- What is the decision boundary of kNN?
- When k is fixed, is the radius of neighborhood fixed?
- Is 1NN the same as minimum distance classifier?

- What is the cost function of kNN? What is the optimization approach used? Is kNN optimal in Baysian sense?
- What are the potential issues with kNN?