CS 559 Machine Learning

Lecture 8: Nonparametric Methods

Ping Wang
Department of Computer Science
Stevens Institute of Technology

STEVENS
INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

Today's Lecture

- Nonparametric Methods
- Kernel Density Estimator
- Nearest Neighbor Algorithm

Nonparametric Methods

- **Parametric approaches:**
  - Use probability distributions that have specific functional forms represented by a set of parameters (e.g. w).
  - w is learned from the training data.
- **Nonparametric approaches:**
  - Make few assumptions about the form of the distribution.

Density Estimation

- In practice, the underlying models can be very complicated and extremely hard to develop specific parametric functional form.
  - The model can be multi-modality.
  - Difficult to collect sufficient amount of data.
  - The selected density in parametric methods might be poor model of the true distribution that generates the data, and further leads to poor predictive performance.
- Therefore, in this lecture, we will consider the nonparametric approaches for density estimation that make few assumptions about the form of the distribution.
- Mainly focus on the frequentist methods.

Density Estimation Using Frequentist Methods

- Given: N observations $\{x_1, \dots, x_N\}$ of a continuous variable x
- Goal: estimate the density model $p(x)$.
- Use histogram: simply partition x into distinct bins
  - N: total number of observations
  - Δ_i: the width of the bins to obtain probability values ($\Delta_i = \Delta$)
  - n_i: the number of observations of x falling in bin i
  - The probability for each bin is given by $p_i = \frac{n_i}{N\Delta_i}$, which is a constant over the width of each bin
  - Easy to show: $\int p(x)dx = 1$

Effect of Bin Width on Histogram

- The data is drawn from the distribution, corresponding to the green curve.
- Choose of bin width:
  - Too small Δ: spiky; with a lot of structure that is not present in the underlying distribution that generated the data set.
  - Too large Δ: too smoothy; fails to capture the bimodal property of the green curve.

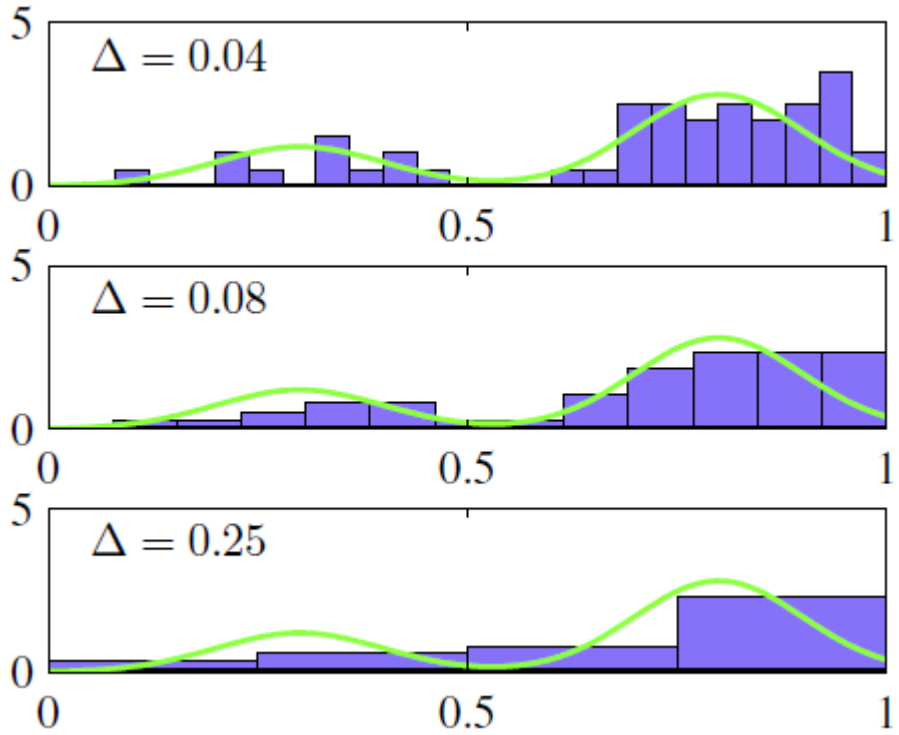
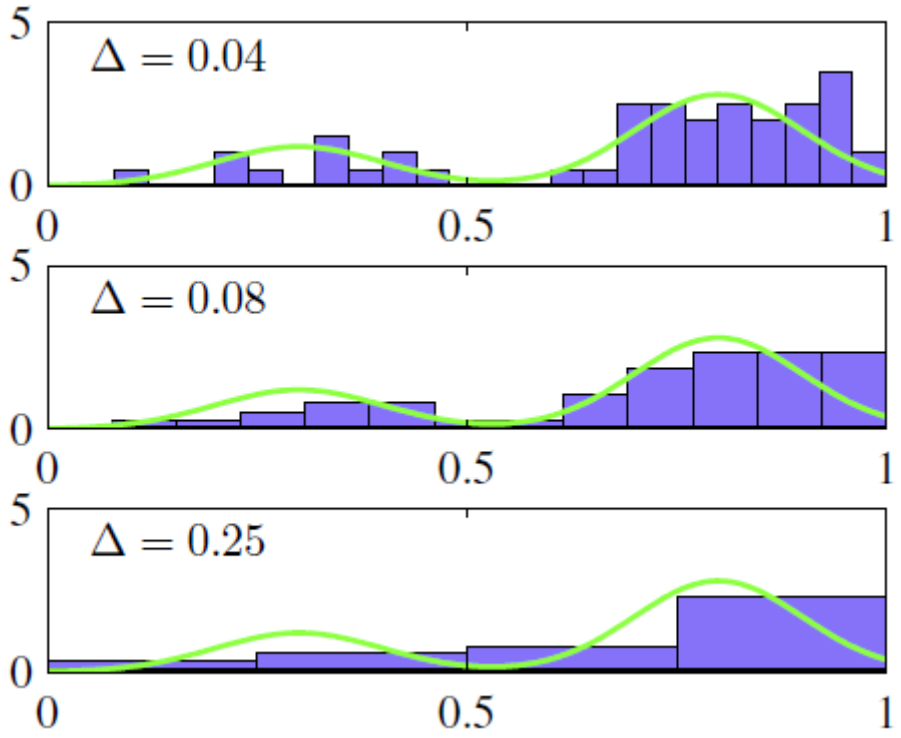


Figure: [C. Bishop, PRML]

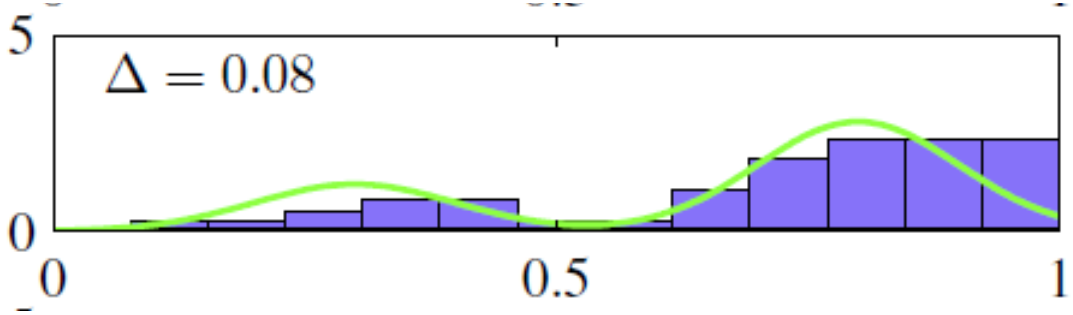
Property of Histogram Method

- **Advantages:**
  - Once the histogram is computed, data can be discarded; good when data is large.
  - Easily applied if data arriving sequentially.
- **Limitations:**
  - Bin edges introduce the discontinuities of estimated density.
  - Exponential scaling with dimensionality: if divide each variable in a D-dimensional space into M bins, the total number of bins will be M^D.



Two Lessons from Histogram Method

1. To estimate the probability density at a particular location, we should consider the data points that lie within some local neighborhood (defined by bins) of that point.
2. The value of the smoothing parameters (e.g., bin width), which describing the spatial extent of the local region, should be neither too large nor too small.



Nonparametric Method for Density Estimation

- Suppose we have collected a data set comprising N observations drawn from unknown probability density $p(x)$
- Goal: estimate the value of $p(x)$
- Consider the small region $\mathcal{R}$ containing x. The probability mass associated with this region is:

$$P = \int_{\mathcal{R}} p(x)dx$$

- Each data point has a probability P of falling within $\mathcal{R}$. In total, K points lie inside region $\mathcal{R}$ follows binomial distribution:

$$Bin(K|N,P) = \frac{N!}{K!\,(N-K)!} P^K (1-P)^{N-K}$$

Nonparametric Method for Density Estimation

- For large N, the distribution will be sharply peaked around the mean:

$$K \approx NP$$

- If the region $\mathcal{R}$ is sufficiently small, $p(x)$ is roughly constant over the region: (V is the volume of the region $\mathcal{R}$)

$$P \approx p(x)V$$

- Our density estimate:

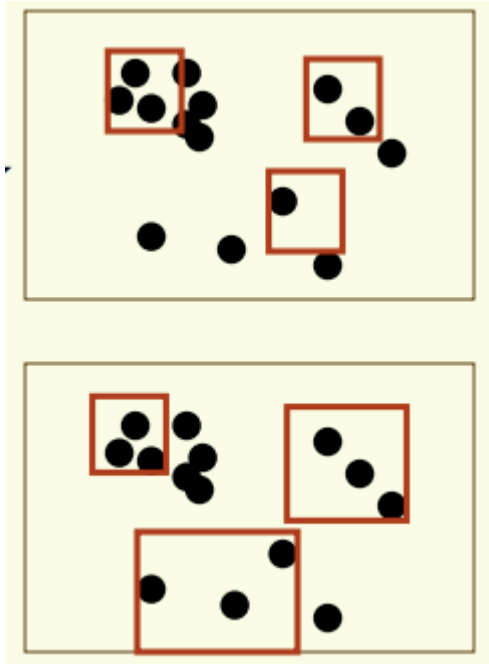
$$p(x) = \frac{K}{NV}$$

Nonparametric Method for Density Estimation

Our density estimate:

$$p(x) = \frac{K}{NV}$$

- We can exploit in **two ways**:
  - If we fix V and determine K from the data, this gives us the kernel approach.
  - If we fix K and determine the value of V from the data, this gives us the K-Nearest Neighbors (KNN).
- Under appropriate conditions and as number of samples goes to infinity, both methods will converge to the true $p(x)$.



Kernel Density Estimators

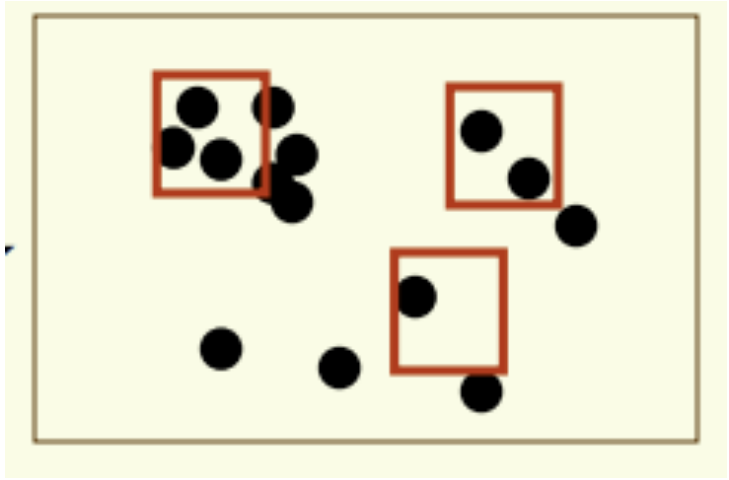
Part of the slides are adapted from https://www.csd.uwo.ca/~oveksler/Courses/CS434a_541a/Lecture6.pdf

Kernel Density Estimators

Our density estimate:

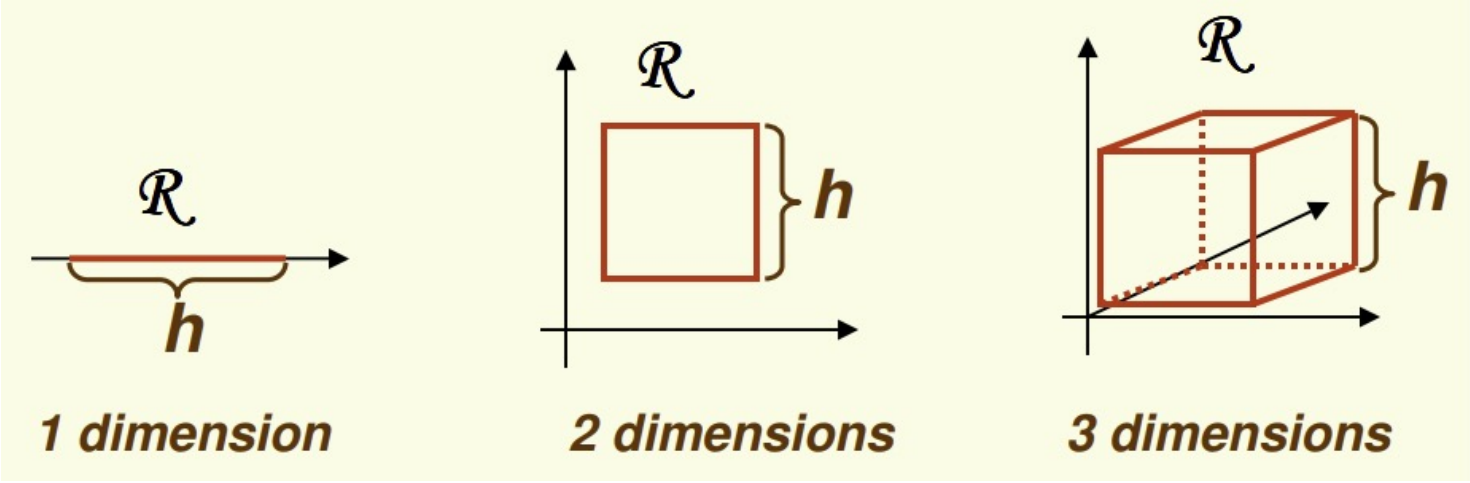
$$p(x) = \frac{K}{NV}$$

- We can exploit in the following way:
  - We fix V
  - Determine K from the data
  - This gives us the kernel approach



Kernel Density Estimators

- In this approach, to estimate densities, we fix the size and shape of region $\mathcal{R}$.
- If we take the region $\mathcal{R}$ to be a small hypercube (with side h, and thus volume is h^D in D-dimensional) centered on the point x, at which we wish to estimate the probability density.
- To estimate the density at point x, simply center the region at x, count the number of points in the region, and substitute everything in the formula.

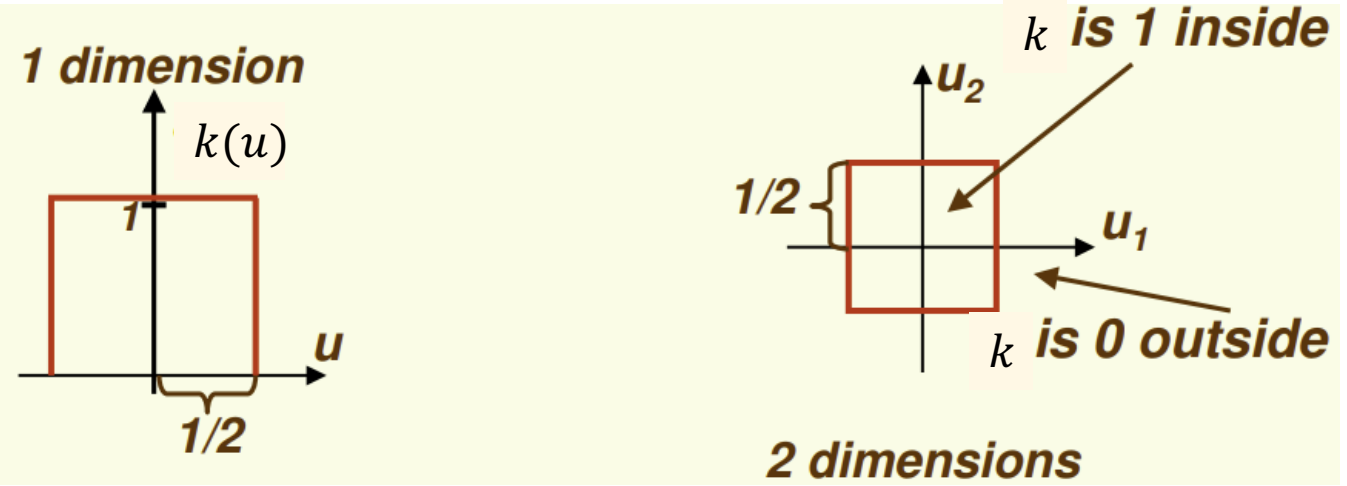


Kernel Density Estimators

- To determine an analytic expression of the probability density.
- We first define a kernel function (parzen window):

$$k(u) = \begin{cases} 1, & |u_i| \le \frac{1}{2}, i = 1, \dots, D \\ 0, & otherwise \end{cases}$$

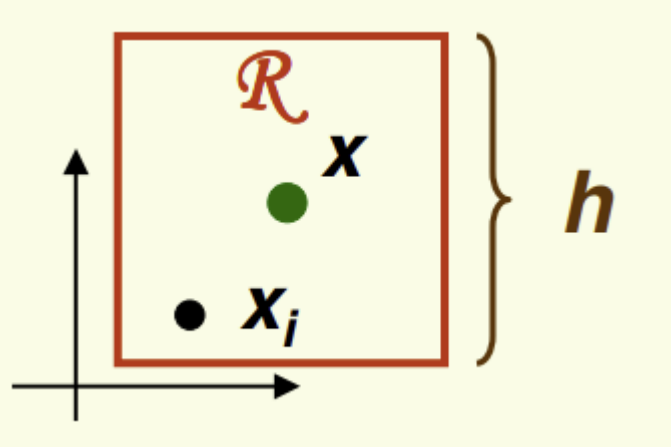
- $k(u)$ defines an unit hypercube centered at the origin.



Kernel Density Estimators

$$k(u) = \begin{cases} 1, & |u_i| \leq \frac{1}{2}, i = 1, \ldots, D \\ 0, & otherwise \end{cases}$$

- $k\left(\frac{x-x_n}{h}\right)$ will be 1 if the data point x_n lies inside a cube of side h centered on x and zero otherwise.



Kernel Density Estimators

- The total number of data points lying inside this cube will therefore be:

$$K = \sum_{n=1}^{N} k\left(\frac{x - x_n}{h}\right)$$

- Recall that our density estimate is:

$$p(x) = \frac{K}{NV}$$

- Substituting K into $p(x)$, we get the analytical expression for the estimated density at x ($V = h^D$ is the volume of the cube in D dimensions):

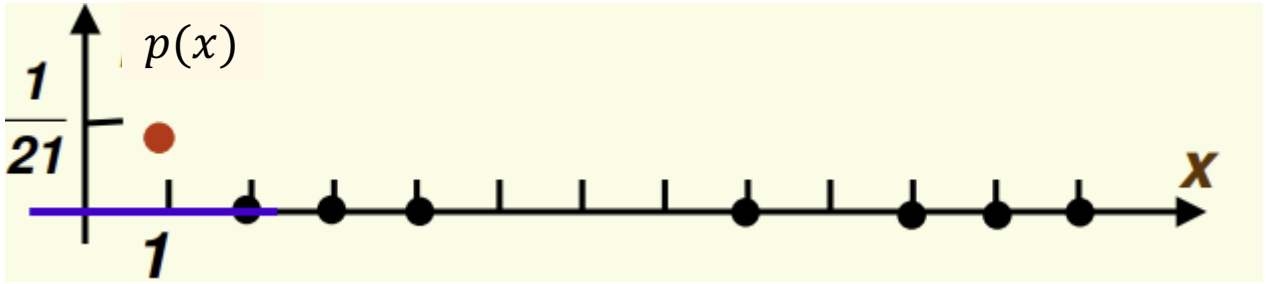
$$p(x) = \frac{K}{NV} = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{h^D} k\left(\frac{x - x_n}{h}\right)$$

Verify $p(x)$ is a Density

- $p(x) \geq 0, for\ any\ x$
- You can also verify that $\int p(x)dx = 1.$

Example in 1D

- $p(x) = \frac{K}{NV} = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{h^D} k\left(\frac{x-x_n}{h}\right)$
- Suppose we have 7 samples $D = \{2, 3, 4, 8, 10, 11, 12\}$, window width $h = 3$, estimate the density at $x = 1$.



- $p(x = 1) = \frac{1}{7}\sum_{n=1}^{7} \frac{1}{3} k\left(\frac{1-x_n}{3}\right) = \frac{1}{21}\left[k\left(\frac{1-2}{3}\right) + k\left(\frac{1-3}{3}\right) + \cdots + k\left(\frac{1-12}{3}\right)\right] = \frac{1}{21}$

Kernel Density Estimators

- $p(x) = \frac{K}{NV} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{h^D}k\left(\frac{x-x_n}{h}\right)$
- Problems: discontinuities at the boundaries of the cubes that the histogram method suffered from.
- We choose a smoother kernel function(Gaussian)

$$k(u) = \frac{1}{(2\pi)^{D/2}}\exp(-\frac{u^2}{2})$$

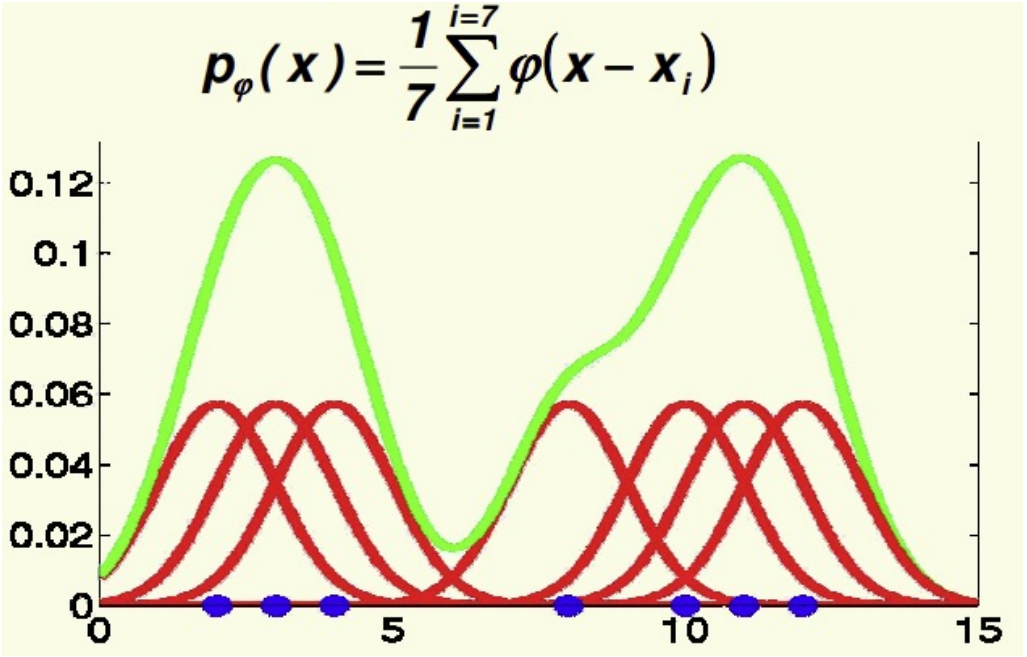
and get the kernel density model:

$$p(x) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{D/2}}\exp\{-\frac{\|x-x_n\|^2}{2h^2}\}$$

where h represents the standard deviation of the Gaussian components.

Kernel Density Estimators

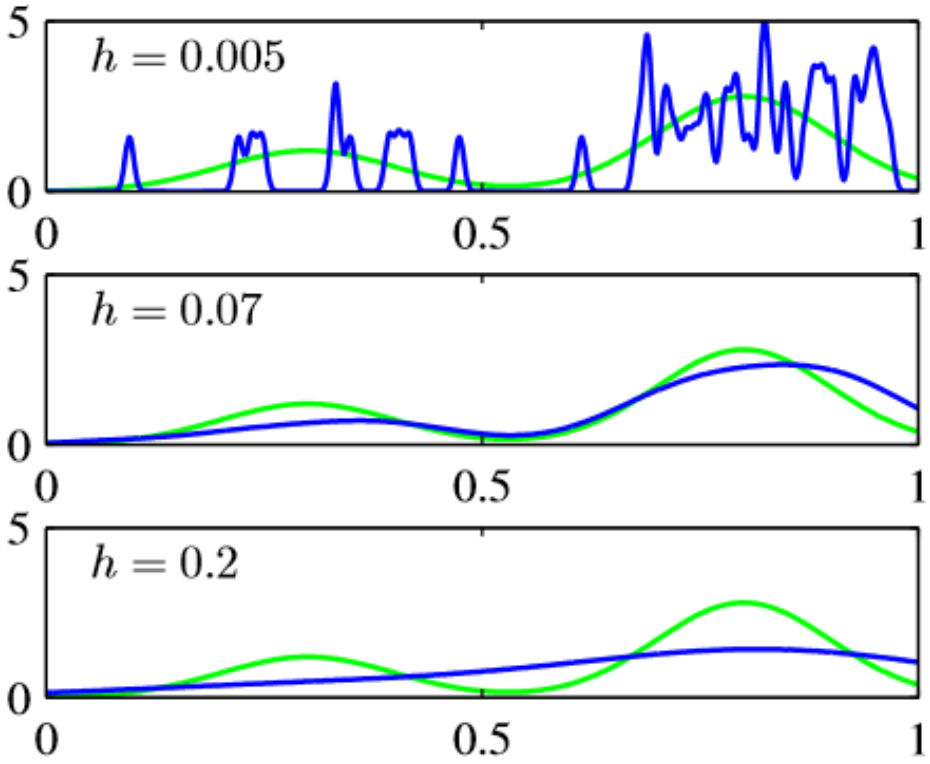
- Essentially, take a Gaussian centered at each data point and represent the unknown density as a mixture of these Gaussians.
- Counting the weighted average of every single sample point.
- 7 samples $D = \{2,3,4,8,10,11,12\}, h = 1$
- The density is estimated by summation of 7 Gaussians, each centered at one of the sample point, and scaled by $\frac{1}{7}$.



Effect of h on $p(x)$

- Green: underlying distribution.
- Blue: Parzen window estimated density using Gaussian.

- If h is too large, the density estimate $p(x)$ is a superposition of N broad, slowly changing functions, thus will be very smooth and "out-of-focus".
- If h is too small, the estimate $p(x)$ will be just superposition of N sharp pulses centered at training samples.



From [C. Bishop PRML]

Problem with Kernel Density Estimators

- h determines the kernel width for all kernels.
- Large value of h may lead to over-smoothing
- Reducing h may lead to noisy estimates
- Thus, the optimal choice for h may be dependent on location within the data space.

Parzen Window Classification

- Estimate the density for each category $p(x|C_i)$ using Parzen window (kernel density estimation), and class prior $p(C_i)$.
- Classify the test point by the class label that has maximum posterior $p(C_i|x)$ through Bayes formula.
- The decision region for a Parzen window classifier depends upon the choice of window function and window width.

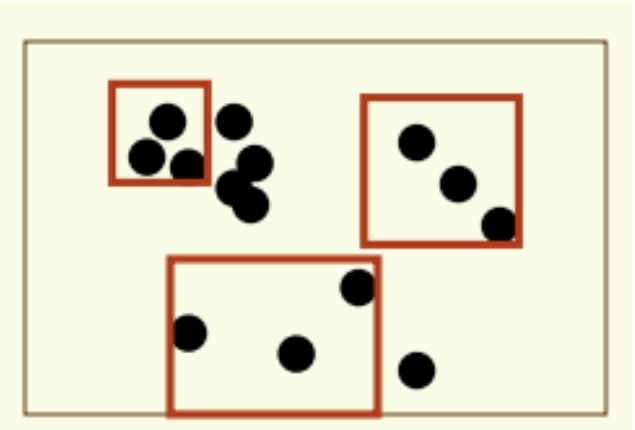
Nearest Neighbor Algorithm

Nearest Neighbor Algorithm

Our density estimate:

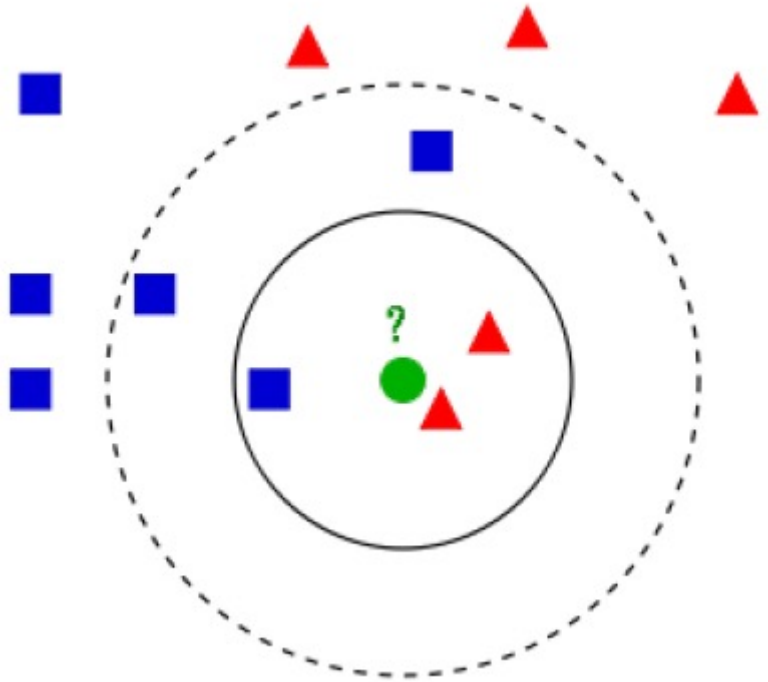
$$p(x) = \frac{K}{NV}$$

- We can exploit in **the following way**:
  - We fix K
  - Determine the value of V from the data
  - This gives us the K-Nearest Neighbors (KNN).



K-Nearest Neighbor Methods

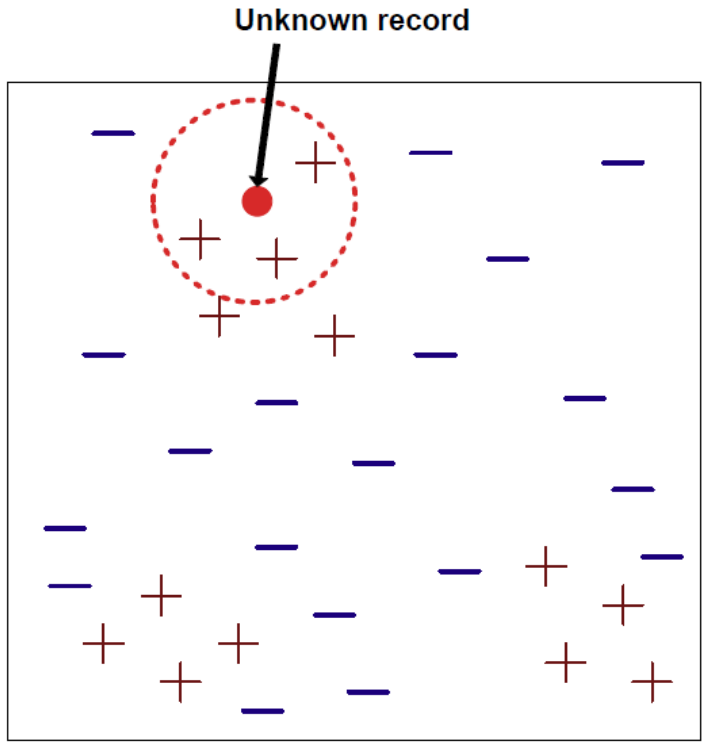
- To classify a new input vector x, examine the K-closest training data points to x and assign the object to the most frequently occurring class.
- Features should be on the same scale (normalize).



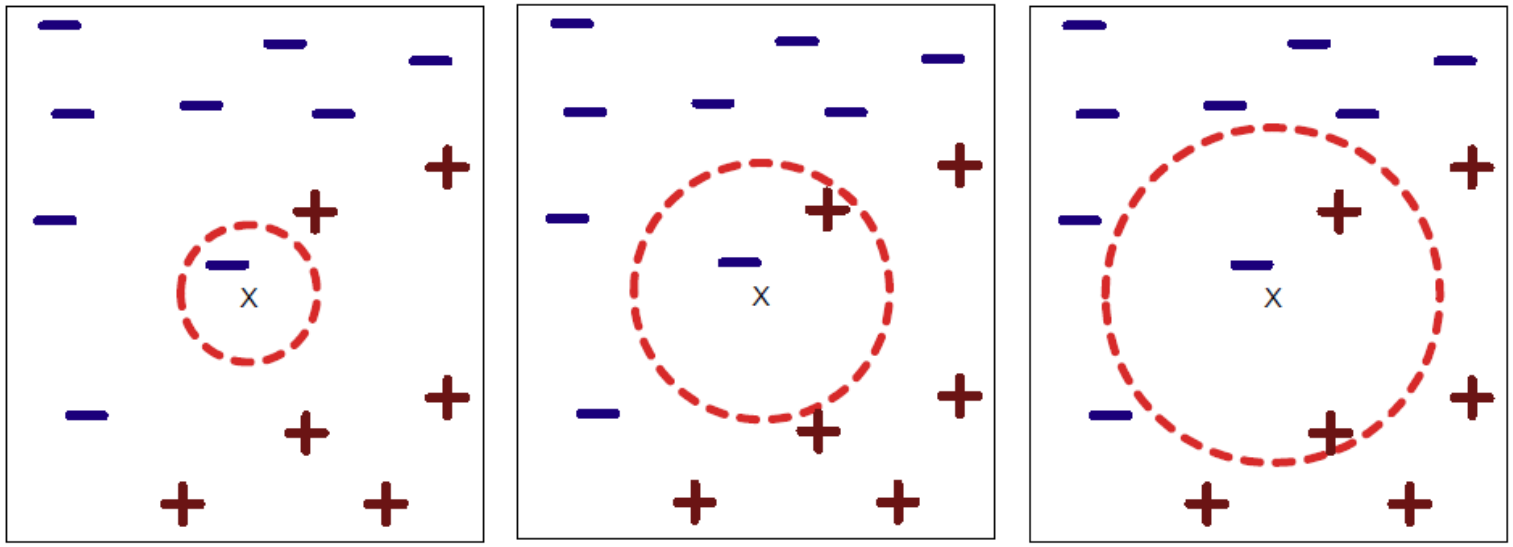
Common values for K: 3, 5

K-Nearest Neighbor Methods

- **Requires three things**
  - The set of stored data points
  - Distance metric to compute distance between records
  - The value of K, the number of nearest neighbors to retrieve
- **To classify an unknown record:**
  - Compute distance to other training records
  - Identify K nearest neighbors
  - Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)



Definition of Nearest Neighbor

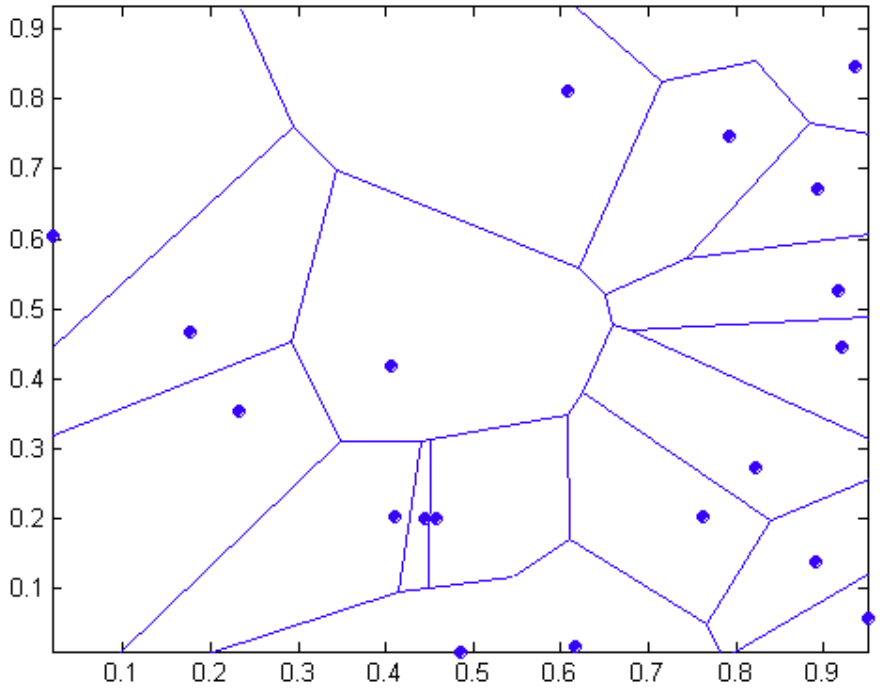


(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a data sample x are data points that have the K smallest distance to x

Decision Boundaries

- The nearest neighbor algorithm does not explicitly compute decision boundaries.
- However, the decision boundaries form a subset of the Voronoi diagram for the training data.
- A partition of the plane into Voronoi cells close to each data point. Each Voronoi cell contains all points of the plane that are closer to the corresponding data point than to any others.
- The more examples that are stored, the more complex the decision boundaries can become.



Distance Measures

- Distance measures
  - Euclidean distance:

    $$d(x_i, x_j) = \sqrt{\sum_{m=1}^{D} (x_{im} - x_{jm})^2}$$

  - Manhattan distance: The distance between two points measured along axes at right angles.

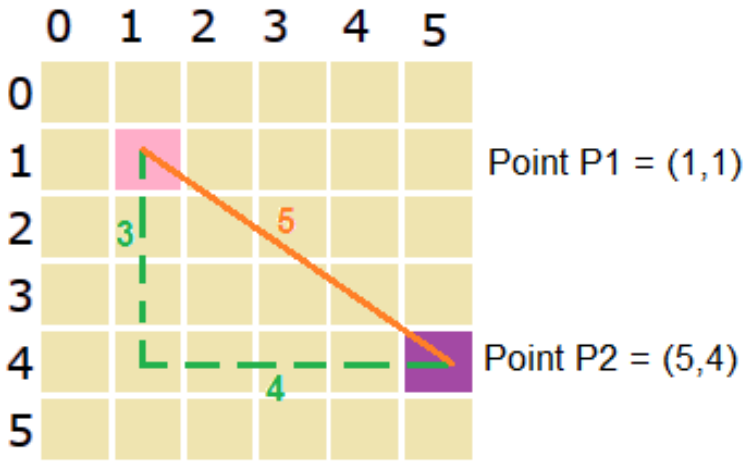
    $$d(x_i, x_j) = \sum_{m=1}^{D} |x_{im} - x_{jm}|$$

  - Similarity: dot product
  - Hamming distance: for binary valued features

    $$d(x_i, x_j) = \sum_{m=1}^{D} \mathbb{I}(x_{im} \neq x_{jm})$$

- Can assign weights to features:

  $$d(x_i, x_j) = \sum_{m=1}^{D} w_m d(x_{im}, x_{jm})$$

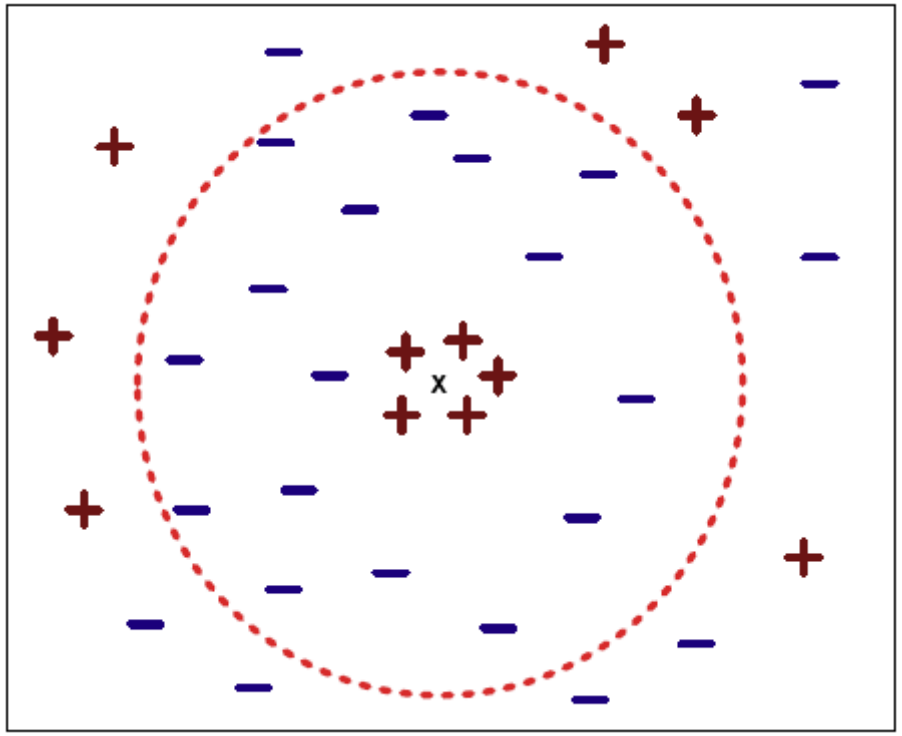


Euclidean distance = $\sqrt{(5-1)^2 + (4-1)^2} = 5$

Manhattan distance = $|5-1| + |4-1| = 7$

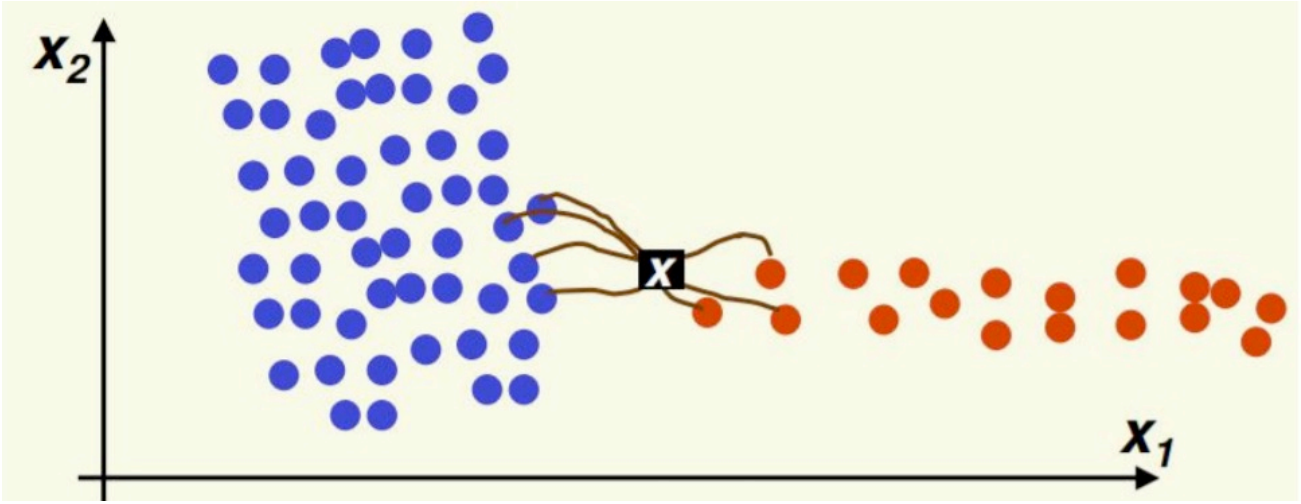
K-Nearest Neighbor Methods

- Determine the class from nearest neighbor list.
  1. Take the majority vote of class labels among the K-nearest neighbors
  2. Weigh the vote according to distance weight factor. For example, $w = \frac{1}{d^2}$.
- Choosing the value of k:
  - If K is too small, sensitive to noise points
  - If K is too large, neighborhood may include points from other classes



Choose K for classification

- K should be large so that error rate is minimized, better classification. (too small K leads to noisy decision boundaries)
- K should be small enough so that only nearby samples are included. (too large K will lead to over smoothed boundaries)



When K is small, x is correctly classified. When K is large, x is misclassified.
http://www.chengjianglong.com/slides/Lecture_7.pdf

Multi-Class Classification - Posterior Probability

- Assuming: N_j points in class C_j with N points in total, so that $\sum_j N_j = N$. To classify a new point x, we draw a sphere centered on x containing K points irrespective of their class. Suppose this sphere has volume V and contains K_j points from class C_j.
- An estimate of the density associated with each class

$$p(x|C_j) = \frac{K_j}{N_j V}$$

- The unconditional density is given by:

$$p(x) = \frac{K}{NV}$$

- The class priors are: $p(C_j) = \frac{N_j}{N}$
- Using Bayes' theorem, the posterior probability of class membership:

$$p(C_j|x) = \frac{p(x|C_j)p(C_j)}{p(x)} = \frac{K_j}{K}$$

Summary

- Non-parametric, no training.
- Prediction algorithm: Look at the K most similar training examples (nearest neighbors).
  - Classification: assign the majority class label (majority voting) of these K neighbors
  - Regression: assign average response of these K neighbors
- Limitations:
  - Require the entire training data set to be stored.
  - Expensive computation if the data set is large.
  - Use specific training instances to make predictions without a model => Instance-based learning

Readings

- Chapter 2.5 of Pattern Recognition and Machine Learning by C. Bishop.

Summary of Today's Lecture

- Nonparametric Methods
- Kernel Density Estimator
- Nearest Neighbor Algorithm