COMP 6838 Data Mining

LECTURE 11:
Supervised classification: k-nn classifiers

Dr. Edgar Acuna
Department of Mathematics

University of Puerto Rico- Mayaguez

math.uprm.edu/~edgar

In the method of the k nearest neighbors (Fix y Hodges, 1951) the density function, from where the dataset is coming, is estimated.

In the supervised classification context the k-nn method is used to estimate the class conditional density $f(\mathbf{x}/C_j)$, of the predictors **x** in a given class C_j.

k-nn is a nonparametric method since none assumption on the distribution of the predictors is made.

K-nn univariate density estimation

- Let x_1, x_2,… ..x_n be a random sample fron a unknown density function f(x), and let t a real number where f is going to be estimated.
- Recall that the probability of x lies in the interval (t-h,t+h), can be approximated by 2hf(t), where f is teh density function and h is a constant near to zero.

  On the other hand, such propbability can also be estimated by k/n, where k is the number of observations in the interval (t-h,t+h). In k-nn estimation, k is pre-fixed and h is computed according to h.

  Formally, let d(x,y)=|x-y| be the usual distance between the points x and y of the real line.

Suppose that we have computed all the distances $d(x_i,t)=|x_i-t|$ and that they are ordered in increasing order, such as

$d_1(t) \leq d_2(t) \leq \ldots \ldots . \leq d_n(t)$

- Then the estimator of the density function f at the point t, based on the k nearest neighbors is given by

$$\hat{f}(t) = \frac{k}{2nd_k(t)}$$

Example

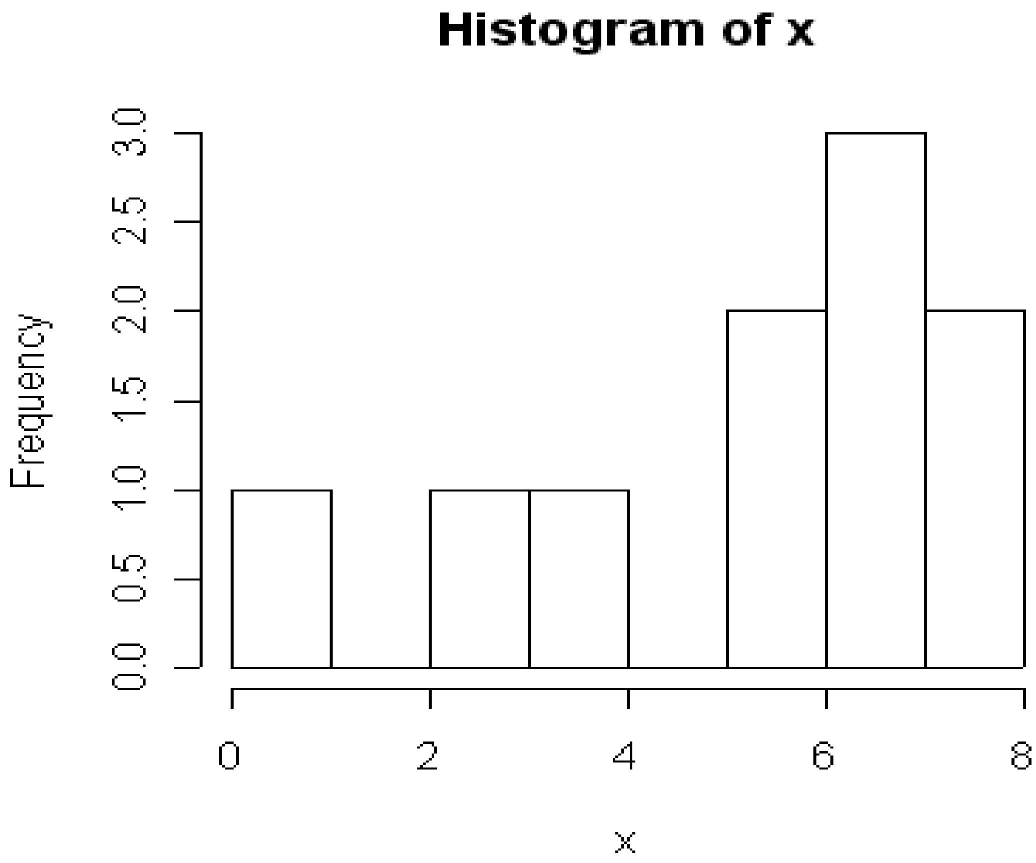
Estimate using k-nn the density fucntion corresponding to the following dataset.

7.03860 6.38018 6.95461 2.25521 7.24608 6.62930 3.92734 0.40701 5.59448 5.05751

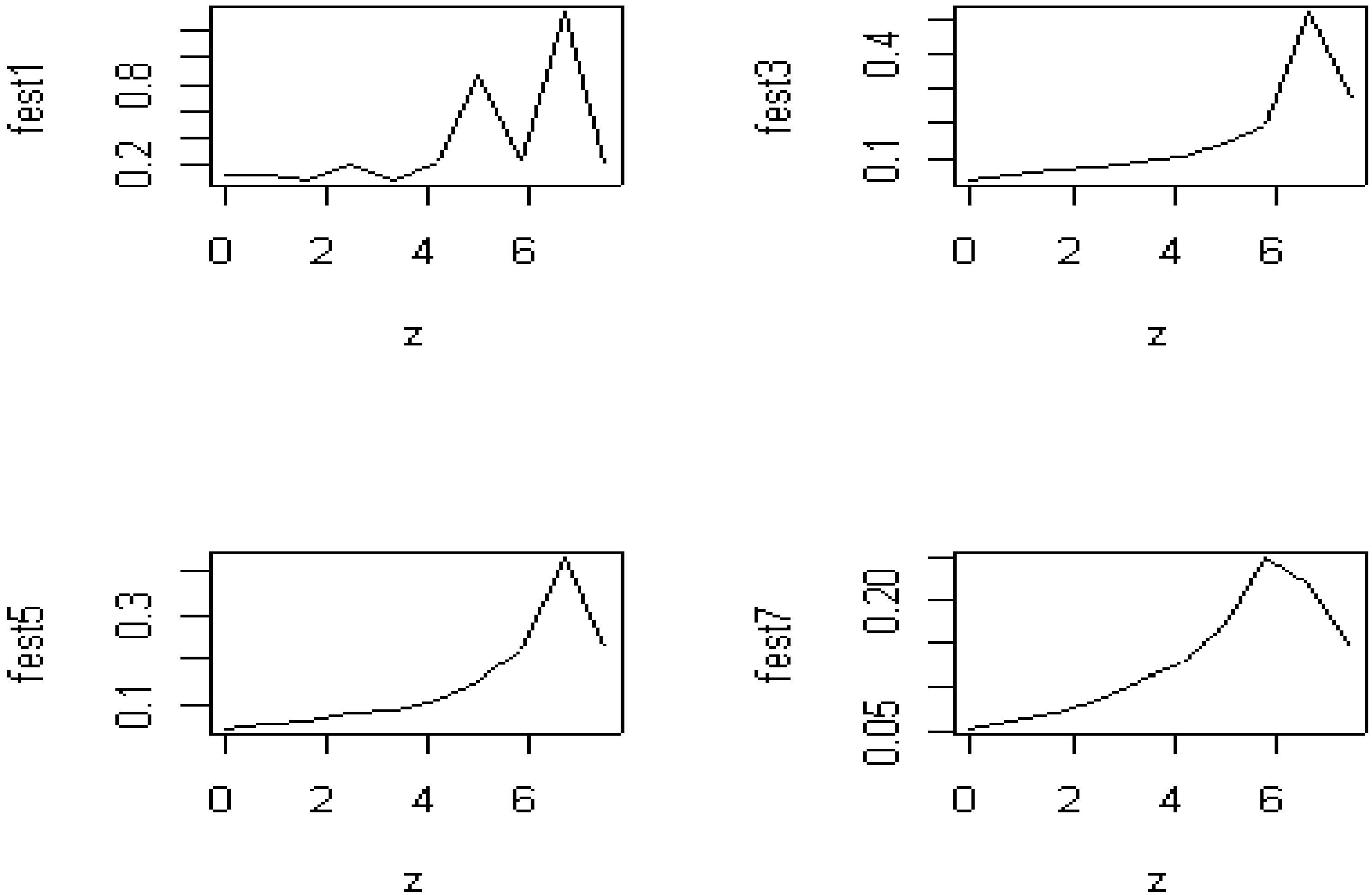
The normalized histogram (normalized such that the area under the curve is 1) for the dataset is shown in the following slide.

The k-nn estimator of the density function is shown below.

```
>y=c(7.03860,6.38018,6.95461,2.25521,7.24608,6.62930,3.92734,0.40701,5.59448,5.05751)
> fdknn(y,10,1)
 [1] 0.05000000 2.79624682 0.05656350 0.50938975 0.08448023 0.12811826
 [7] 0.20642959 0.50845160 2.79624682 0.05000000
```


Histogram of x
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
2
4
6
8
x

k-nn estimator of the density function for k=1,3,5,7



Multivariate k-nn density estimation

- The estimate of the density function is given by

$$\hat{f}(\mathrm{x}) = \frac{k}{n v_k(\mathrm{x})}$$

- where $v_k(\mathbf{x})$ is the volume of the ellipsoid centered at **x** **a**nd with radius $r_k(\mathbf{x})$, which is the distance of x to the k-th nearest point.

The k-nn classifier

In the context of supervised classification, the k-nn method is applied in a straight forward manner.
In fact, if the class conditional density, ***f(x/C$_i$)**, of* the class Ci that appears in the equation

$$P(C_i / \mathrm{x}) = \frac{f(\mathrm{x} / C_i)\pi_i}{f(\mathrm{x})}$$

Is estimated by k-nn. Then, in order to classify into the class Ci un object, with measurements given by the vector **x,** the inequality

$$\frac{k_i \pi_i}{n_i v_k(\mathrm{x})} > \frac{k_j \pi_j}{n_j v_k(x)}$$

must hold for all j≠i. Where k_i y k_j are the k neigbohrs in classes v C_i and C_j respectively.

Asuming priors proportionals to the class size (ni/n y nj/n respectivamente) the above expression is equivalent to :

$$k_i > k_j \text{ for } j \neq i$$

Then, the procedure for classifying a object x would be:

1) Find the k objects that are closer to the object **x**, k usually is a odd number, 1 or 3.
2) If the majority of those k objects belong to the class Ci then the object **x** is assigned to this class. In case of a tie the object's assignment is done randomly.

- There are two problems in the k-nn method; the choice of the distance and the choice of k.
- The most simple distance that can be chosen is the Euclidean, d(**x**,**y**)=(**x**-**y**)'(**x**-**y**). However the use of this metric can give problems when the predictors have been measured in different scale among them. It is a good idea to normalize the data before the application of the k-nn method. Other well used distance is the Manhattan distance given by, d(**x**,**y**)=|**x**-**y**|. Discrete predictors can be transformed in continuous before applying k-nn.
- Using simulation, Enas and Choi (1996), carried out an study to determine the optimum k for the two-class problem and they found out if the sample size of the two classes are similar, then $k=n^{3/8}$ if the covariance matrices of the two classes are similar, and $k=n^{2/8}$ if the covariances matrices are quite different.

- The bias of the misclassification increases as k increases, but the variance decreases.
- It has been proved that the misclassification error rate for the k-nn classifier is at most twice the optimum error rate, error rate for the Bayesian classifier where the posterior are known,(Cover y Hart, 1967).

Example

```
> knn(bupa[,1:6],bupa[,1:6],bupa[,7],k=3)
  [1] 2 2 2 2 2 2 1 1 1 1 2 2 2 1 1 2 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 2 1
 [38] 2 2 2 2 2 1 2 1 2 2 2 1 2 1 1 2 2 2 2 2 1 2 1 2 2 2 2 1 1 2 2 2 2 2 1 1 1
 [75] 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1
[112] 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2 1 1 2 1 2 2
[149] 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 1 1 2 1 1 2 1 2 2 1 2 2 1 1 2 2
[186] 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 2 2 1 1 1 1 2 1 2 2 2 2 2
[223] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 2 2 2 1 2 2 1 2 1 1
[260] 2 1 2 1 2 2 2 2 2 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 2
[297] 2 1 1 2 1 2 2 2 2 2 2 1 2 2 2 1 2 1 1 1 1 1 2 2 2 2 2 2 1 1 1 1 2 2 2 1 2
[334] 2 1 2 2 2 1 2 2 1 1 1 2
Levels: 1 2
```

Error de mala clasification por resustitucion

```
> mean(knn(bupa[,1:6],bupa[,1:6],bupa[,7],k=3)!=bupa[,7])
[1] 0.1971014
```

Example (cont.)

```
> mean(knn.cv(bupa[,1:6],bupa[,7],k=5)!=bupa[,7])
[1] 0.3391304
> mean(knn.cv(bupa[,1:6],bupa[,7],k=7)!=bupa[,7])
[1] 0.3130435
> mean(knn.cv(bupa[,1:6],bupa[,7],k=9)!=bupa[,7])
[1] 0.3043478
> mean(knn.cv(bupa[,1:6],bupa[,7],k=11)!=bupa[,7])
[1] 0.3188406
> crossval(bupa,method="knn",kvec=7,repet=10)
[1] 0.3133333
> crossval(bupa,method="knn",kvec=9,repet=10)
[1] 0.3084058
> crossval(bupa,method="knn",kvec=11,repet=10)
[1] 0.3226087
```

K-nn in WEKA

After open the data file, choose Classify and from the list of classifiers choose

Lazy and then IB1 to perform 1-nn classification and IBk to perform k-nn classification.

Example: for diabetes the misclassification error rate estimated by cross-validation with k=1 gives 29.81%, with k=3 gives 27.34% and, with k=5 gives 26.82%

Other nonparametric classifier

- Based on kernel density estimation. Its main disadvantage is that it requires a lot of computation time. The Gaussian kernel is the one most frequently used.
- In general, the nonparametric methods face the curse of dimensionality problem. In order to have a good estimation the number of data points should increase exponentially to the dimension of the dataset.