CSCI567 Fall 2015

Homework #1

Vaibhav Behl

vbehl@usc.edu

1. Density Estimation

a) Given N i.i.d samples $x_1, x_2, x_3, \ldots, x_n$, we need to use MLE to estimate parameters for the following cases-

Case No.1

Samples generated from Beta distribution having values between 0 and 1, with unknown parameter α and given β = 1.

Beta Distribution is defined by-

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

Given that $\beta = 1$,

The Beta function in the denominator is simplified as-

$$B(\alpha, \beta) = \frac{(\alpha-1)!\,(\beta-1)!}{(\alpha+\beta-1)!}$$

$$B(\alpha, 1) = \frac{1}{\alpha}$$

Substituting this value in main equation-

$$f(x; \alpha, 1) = \alpha x^{\alpha-1}$$

Its likelihood function will be as follows-

$$L(\theta) = \prod_{i=1}^{n} f(X_i;\ \alpha, 1)$$

$$= \prod_{i=1}^{n} \alpha\, x^{\alpha-1}$$

Taking log,

$$l(\theta) = \sum_{i=1}^{n} \ln(\alpha x^{\alpha-1})$$

$$= \sum \quad \ln(\alpha) + \sum \quad \ln(x^{\alpha-1})$$

Taking derivative

$$\frac{dl(\theta)}{d\alpha} = \sum \frac{1}{\alpha} + \sum \ln(x)$$

$Equating\ the\ derivative\ to\ zero\ get\ the\ maximum -$

$$\frac{N}{\alpha} + \sum \quad \ln(x) = 0$$

$$\alpha = \frac{-N}{\sum \ln(x)}$$

Case No.2

In this case the samples are generated from a Normal distribution - $N(\theta, \theta)$

So, for this distribution we have, - $\mu = \theta\ and\ \ \sigma = \sqrt{\theta}$

Using the equation for Normal Distribution we have-

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sqrt{\theta}}\ e^{-\frac{(x-\theta)^2}{2\theta}}$$

Its log-likelihood function is as follows –

$$l(\theta) = \sum \ln \frac{\theta^{-1}}{\sqrt{2}\pi} + \sum \frac{-(x-\theta)^2}{2\theta}$$

$$= -\frac{1}{2} \sum \ln(2\pi\theta)\ - \sum \frac{(x-\theta)^2}{2\theta}$$

Taking its derivative and equating it to zero,

$$\frac{dl}{d\theta} = -\frac{N}{2\theta} - \sum \frac{2\theta[2(x-\theta)(-1)] - 2(x-\theta)^2}{4\theta^2} = 0$$

Or,

$$N\theta = \sum 2\theta(x-\theta) + (x-\theta)^2$$

$$N\theta = \sum (x^2 - \theta^2)$$

$$N\theta = \sum x^2 - \sum \theta^2$$

$$N\theta^2 + \frac{N\theta}{2} - \sum x^2 = 0$$

Solving this equation we get –

$$\theta = \frac{-\frac{N}{2} \pm \sqrt{\left(\frac{N}{2}\right)^2 + 4N\sum x^2}}{2N}$$

2. Nearest Neighbor

a) We are given 10 USC students with their classes and locations as follows-

Mathematics: {(10; 49); (12; 38); (9; 47)}
Electrical Engineering: {(29; 19); (32; 31); (37; 38)}
Computer Science: {(8; 9); (30;28); (18;19); (21; 12)}

Case 1- We first normalize this data. We normalize the x and y values of the feature separately. The mean and standard deviation(std) values of these are given below-

x-mean = 8.6, x-std = 22.4806
y-mean = 19.6, y-std = 26.5589

Using these values we normalize the points as follows –

0.0623	1.1070 (M)
-0.9163	0.6928 (M)
-0.7829	1.0317 (M)
0.9074	-0.0226 (EE)
1.0409	0.4292 (EE)
1.2633	0.6928 (EE)
-0.0267	-0.3991 (CS)
0.9519	-1.7922 (CS)
-1.1832	-1.4534 (CS)
-1.3167	-0.2862 (CS)

Case 2 – Now we need to label a new student with co-ordinate- (9, 18). This new point when normalizes becomes – (0.0178, -0.0602).

Now firstly we'll be using the L2 as the distance metric in our K-NN algorithm –

We calculate the distance between this new points and all the other given points. These come out to be-

1.1681 (M)
1.1999 (M)
1.3540 (M)
0.8905 (EE)
1.1342 (EE)
1.4555 (EE)
0.3418 (CS) <- Lowest
1.9678 (CS)
1.8394 (CS)
1.3535 (CS)

Now using (K=1), we can classify the new points as – Computer Science, as the minimum distance of 0.3418 is for point (8; 9) which is classified as Computer Science.

For (K=3), we get One point belonging to CS {(8,9)}, and Two points Belonging to EE{(29; 19),(32; 31)}. So we classify it as Electrical Engineering.

Now, using L1 metric, the distance from the new point are –

1.2117 (M)
1.6872 (M)
1.8926 (M)
0.9273 (EE)
1.5126 (EE)
1.9986 (EE)
0.3834 (CS) <- Lowest
2.6661 (CS)
2.5942 (CS)
1.5604 (CS)

Now using (K=1), we can classify the new points as – Computer Science, as the minimum distance of 0.3418 is for point (8,9) which is classified as Computer Science. For (K=3), we get one point belonging to Computer Science{(8,9)}, one point belonging to Electrical Engineering{(29, 19)} and one point belonging to Mathematics{(10,49)}. So there is a 'tie'. As specified in the question, in case of a tie we take the class with the lowest distance, so we classify it as Computer Science.

Result Comparison

For L_2

Using K=1, we get Computer Science
For K=3, we get Electrical Engineering

For L_1

Using K=1, we get Computer Science,
For K=3, we have a 'tie' and we get Computer Science, after resolving that tie.

b) In this question, we have to classify an unlabeled data point x, which is D-dimensional. We are also given a sphere of volume V which can contain K labeled data points. Total points in the space are defined as N, with $\sum_c N_c = N$. Also, K_c data points are said to be inside the sphere.

Now, Density estimated with each class is modeled as - $p(X|Y = c) = \frac{k_c}{N_c V}$ and class prior probability as - $p(Y = c) = \frac{N_c}{N}$.

Now, for part 1 we need to find, p(x). We use the conditional probability equation,

$$p(X) = \sum_{c} p(X|Y = c)p(Y = c)$$

Substituting values into this equation, we get-

$$p(X) = \frac{k_c}{N_c V} \cdot \frac{N_c}{N}$$

Simplifying it, we find the p(X) to be-

$$p(X) = \frac{K}{VN}$$

Also, for part 2 we need to find the posterior probability of class membership $p(Y = c|x)$. From Bayes rule we can write the following equation –

$$P(Y = c|X).P(X) = P(X|Y = c).P(Y = c)$$

Simplifying this we get-

$$P(Y = c|X) = \frac{P(X|Y = c).P(Y = c)}{P(X)}$$

Putting the values we got above, we can see that-

$$P(Y = c|X) = \frac{k_c}{k}$$

3. Decision Tree

a) In this question we need to construct a Decision Tree with maximum depth as 3. We select nodes at each point such that it gives us minimum entropy or maximum information gain.

First we calculate entropy for each of the binary features provided.

- Entropy for Temperature

$H(Rainy|Temperature = Hot) = -\frac{23}{40}\log\frac{23}{40} - \frac{17}{40}\log\frac{17}{40} = 0.6819$

$H(Rainy|Temperature = Cool) = -\frac{13}{40}\log\frac{13}{40} - \frac{27}{40}\log\frac{27}{40} = 0.6306$

Now, finding total entropy using weighted avg.

$H(Rainy|Temperature) = \frac{4}{8}.0.6819 + \frac{4}{8}.0.6306 = 0.6563$

- Entropy for Humidity

$H(Rainy|Humidity = Low) = -\frac{13}{40}\log\frac{13}{40} - \frac{27}{40}\log\frac{27}{40} = 0.6306$

$H(Rainy|Humidity = High) = -\frac{23}{40}\log\frac{23}{40} - \frac{17}{40}\log\frac{17}{40} = 0.6819$

Now, finding total entropy using weighted avg.

$H(Rainy|Humidity) = \frac{4}{8}.0.6306 + \frac{4}{8}.0.6819 = 0.6691$

- Entropy for Sky Condition

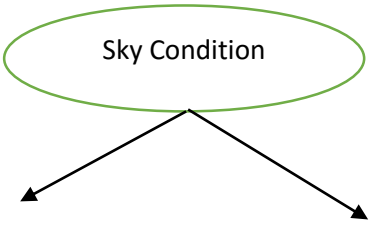
$H(Rainy|Sky\ Condition = Cloudy) = -\frac{25}{40}\log\frac{25}{40} - \frac{15}{40}\log\frac{15}{40} = 0.6616$

$H(Rainy|Sky\ Condition = Clear) = -\frac{11}{40}\log\frac{11}{40} - \frac{29}{40}\log\frac{29}{40} = 0.5882$

Now, finding total entropy using weighted avg.

$H(Rainy|Sky\ Condition) = \frac{4}{8}.0.6616 + \frac{4}{8}.0.5882 = 0.6249$

Now if we use these entropies to calculate 'Gain', we will get the highest gain for feature – Sky Condition, since it has the lowest entropy. Thus our first node will be Sky Condition.



Next, we will calculate under Sky Condition = Cloudy branch.

Calculate entropy for remaining binary features provided.

- Entropy for Temperature

$H(Rainy|Cloudy, Temperature = Hot) = -\frac{15}{20}\log\frac{15}{20} - \frac{5}{20}\log\frac{5}{20} = 0.5623$

$H(Rainy|Cloudy, Temperature = Cool) = -\frac{10}{20}\log\frac{10}{20} - \frac{10}{20}\log\frac{10}{20} = 0.6931$

Now, finding total entropy using weighted avg.

$H(Rainy|Cloudy, Temperature) = \frac{2}{4}.0.5623 + \frac{2}{4}.0.6931 = 0.6278$

- Entropy for Humidity

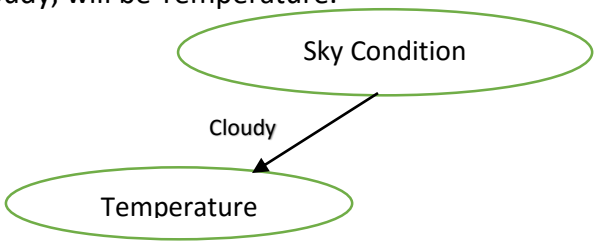
$H(Rainy|Cloudy, Humidity = Low) = -\frac{9}{20}\log\frac{9}{20} - \frac{11}{20}\log\frac{11}{20} = 0.6881$

$H(Rainy|Cloudy, Humidity = High) = -\frac{16}{20}\log\frac{16}{20} - \frac{4}{20}\log\frac{4}{20} = 1.6094$

Now, finding total entropy using weighted avg.

$H(Rainy|Cloudy, Humidity) = \frac{2}{4}.0.6881 + \frac{2}{4}.1.6094 = 1.1487$

Now if we use these entropies to calculate 'Gain', we will get the highest gain for feature – Temperature, since it has the lowest entropy. Thus our node under – Sky Condition= Cloudy, will be Temperature.



Next, we will calculate under Sky Condition = Clear branch.

Calculate entropy for remaining binary features provided.

- Entropy for Temperature

$H(Rainy|Clear, Temperature = Hot) = -\frac{8}{20}\log\frac{8}{20} - \frac{12}{20}\log\frac{12}{20} = 0.6730$

$H(Rainy|Clear, Temperature = Cool) = -\frac{3}{20}\log\frac{3}{20} - \frac{17}{20}\log\frac{17}{20} = 0.4227$

Now, finding total entropy using weighted avg.

$H(Rainy|Clear, Temperature) = \frac{2}{4}.0.6730 + \frac{2}{4}.0.4227 = 0.5479$

- Entropy for Humidity

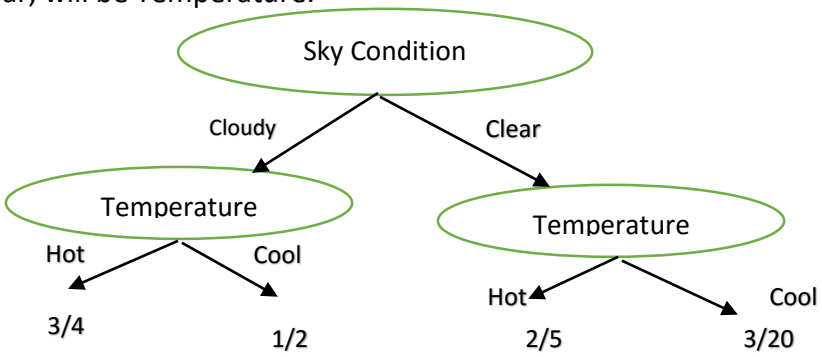
$H(Rainy|Clear, Humidity = Low) = -\frac{4}{20}\log\frac{4}{20} - \frac{16}{20}\log\frac{16}{20} = 0.5004$

$H(Rainy|Clear, Humidity = High) = -\frac{7}{20}\log\frac{7}{20} - \frac{13}{20}\log\frac{13}{20} = 0.6474$

Now, finding total entropy using weighted avg.

$H(Rainy|Clear, Humidity) = \frac{2}{4}.0.5004 + \frac{2}{4}.0.6474 = 0.5739$

Now if we use these entropies to calculate 'Gain', we will get the highest gain for feature – Temperature, since it has the lowest entropy. Thus our node under – Sky Condition= Clear, will be Temperature.



b) In this question we have to prove that, for any discrete probability distribution p with K classes, the value of the Gini-index is less than or equal to the corresponding value of the cross-entropy.

In equation terms this can be written as-

$\sum_{k=1}^{K} p_k(1-p_k) \leq -\sum_{k=1}^{K} p_k \log p_k$

This can be written as –

$$\sum_{k=1}^{K} p_k(1-p_k) + \sum_{k=1}^{K} p_k \log p_k \leq 0$$

Combining the summation and using the inner expression, we can take out p_k since it is always greater than zero and thus we can define a function as –

$$f(p_k) = 1 - p_k + \log p_k$$

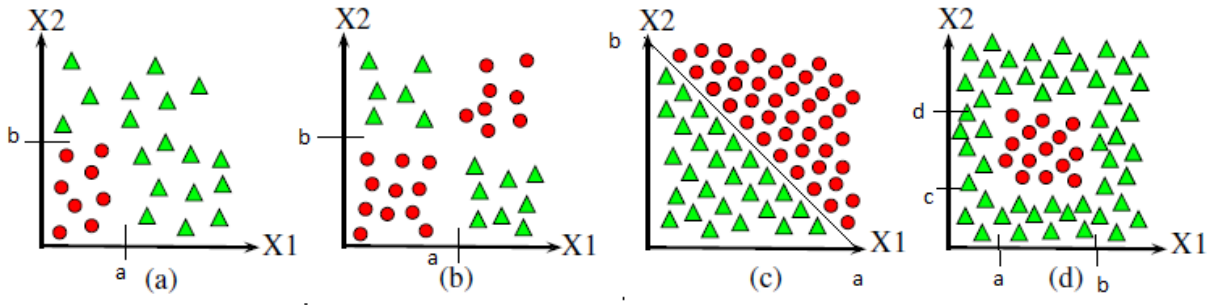
If we can prove that $f(p_k) \leq 0 \; for \; all \; values \; of \; p_k$, we can prove our original equation.

Now taking derivative of this function and equating it to zero, we get-

$$f'(p_k) = -1 + \frac{1}{p_k} = 0$$

This gives us $p_k = 1, which\ gives\ us\ f(1) = 0$. Thus at maximum value $f(p_k) should\ always\ be \leq zero$. Hence we have proved our original hypothesis.

c) Except (c), all the below figures can be classified with a tree depth less than 6.



4. Naive Bayes

b) In this question we want to show that

$$P(Y = 1|X) = \frac{1}{1 + \exp(-w_0 + w^T X)}$$

Where-$w_0 and\ w\ are\ in\ terms\ of\ \pi and\ \theta_{jk}, for\ j = 1, 2, 3, , \ldots D; where\ D\ represents\ the\ number\ of\ multiple\ choice\ questions.$

The left hand term of this equation can be represented using bayes forumuls as below-

$$P(Y = 1|X) = \frac{P(X|Y = 1).P(Y = 1)}{P(X|Y = 1).P(Y = 1) + P(X|Y = 0).P(Y = 0)}$$

Now X here is a D-dimensional binary vector representing answers to questions. These will be independent, so the above term can be further simplified as –

$$P(Y = 1|X) = \frac{(\prod_{i=1}^{D} P(x_i|Y = 1).P(x_i)\ ).P(Y = 1)}{(\prod_{i=1}^{D} P(x_i|Y = 1).P(x_i)\ ).P(Y = 1) + (\prod_{i=1}^{D} P(x_i|Y = 0).P(x_i)\ ).P(Y = 0)}$$

This can be simplified as –

$$= \frac{1}{1 + \frac{(\prod_{i=1}^{D} P(x_i|Y=0).P(x_i)\,).P(Y=0)}{(\prod_{i=1}^{D} P(x_i|Y=1).P(x_i)\,).P(Y=1)}}$$

Using values specified in the question, we can represent the term in the denominator as a function of (θ) –

$$f = \prod_{i=1}^{D} \frac{\theta_{j0}^{x_i}(1-\theta_{j0})^{1-x_i}.(1-\pi)}{\theta_{j1}^{x_i}(1-\theta_{j1})^{1-x_i}.\pi}$$

We take log of this function to simplify it ,

$$\log f = \log\frac{1-\pi}{\pi} + \sum \log\frac{\theta_{j0}^{x_i}(1-\theta_{j0})^{1-x_i}}{\theta_{j1}^{x_i}(1-\theta_{j1})^{1-x_i}}$$

$$= \log\frac{1-\pi}{\pi} + \log\sum \theta_{j0}^{x_i}(1-\theta_{j0})^{1-x_i} - \log\sum \theta_{j1}^{x_i}(1-\theta_{j1})^{1-x_i}$$

$$= \log\frac{1-\pi}{\pi} + \sum \log\frac{1-\theta_{j0}}{1-\theta_{j1}} + \sum \log\left(\frac{(\theta_{j0}(1-\theta_{j1}))}{(\theta_{j1}(1-\theta_{j0}))}\right)^{x_i}$$

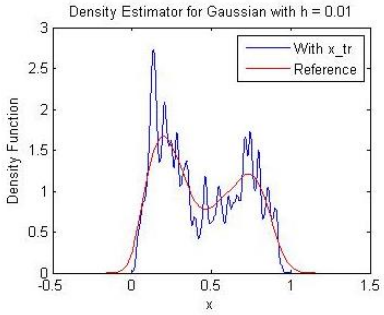
Now the first two terms in this equation are constants, so we can substitute those for w_0 and last term can be replaced for w^T. Substituting these values we can arrive at the equation form we wanted.

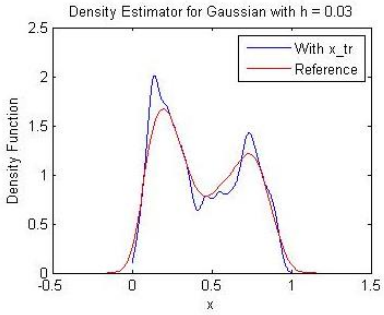
5. Programming

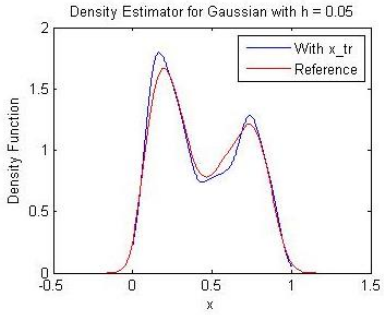
5.1 Density Estimation

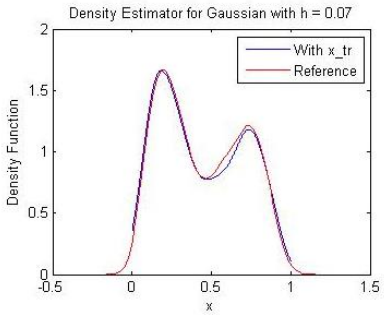
a) Following are the plots for our three kernels on the following five values of h – {0.01 0.03 0.05 0.07 0.1}. Here the blue line is plotted with the x_tr dataset and the red line is a reference line plotted using all the data (both x_tr and x_te), to act as a reference so that comparison is easy.

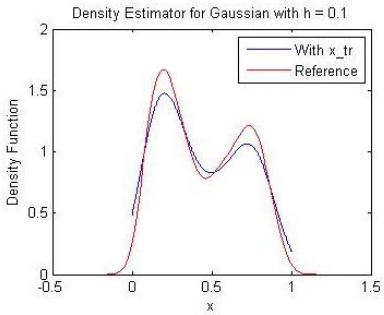
<u>Gaussian</u>



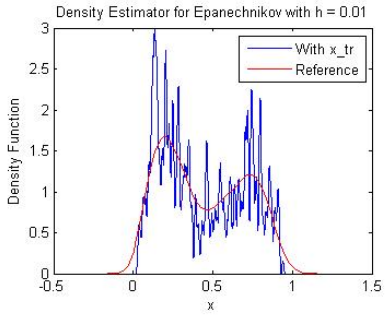


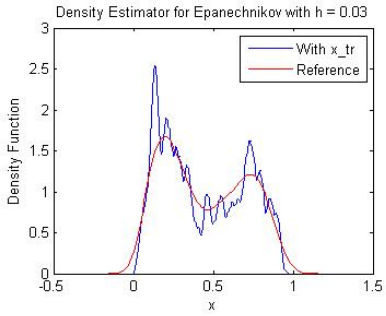


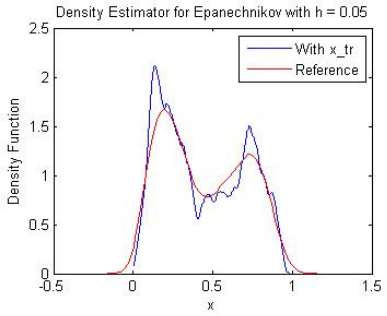


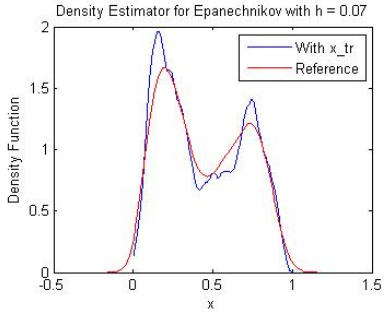


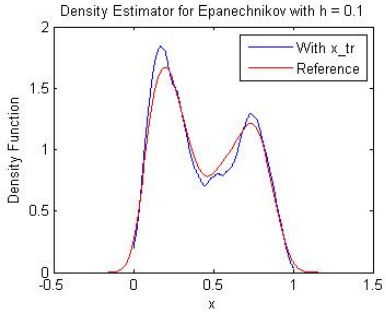
Epanechnikov



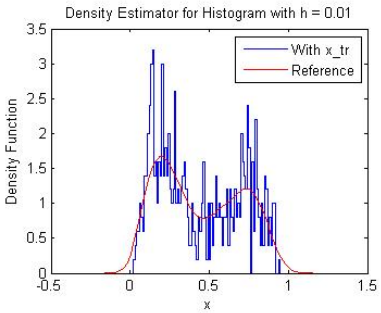


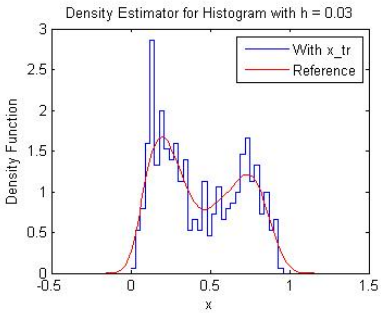


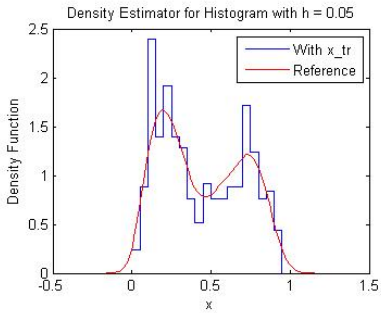


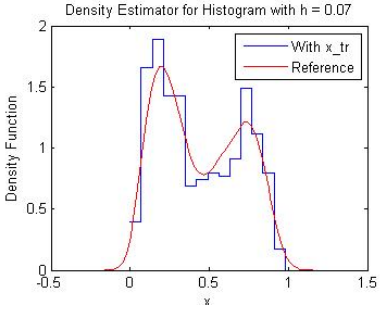


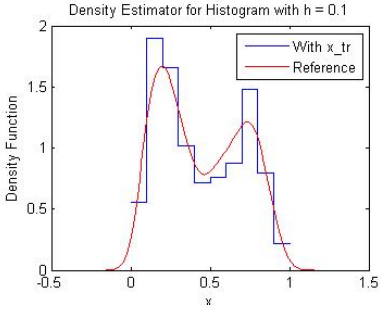
Histogram



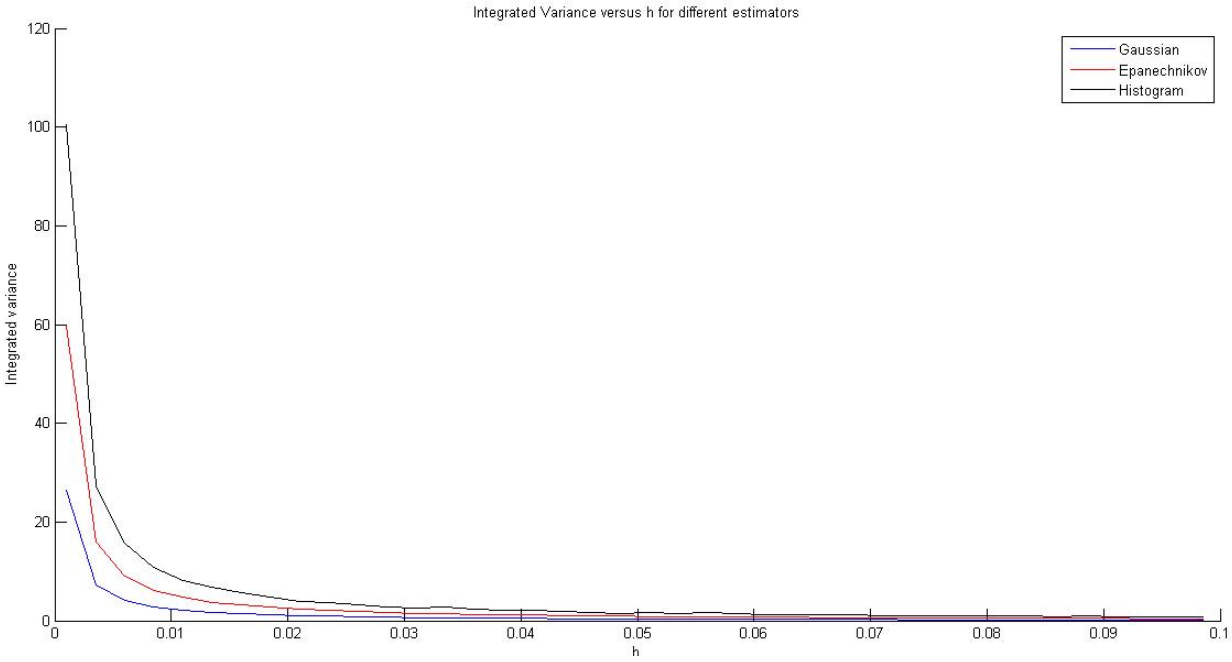








b) Now we have to plot the integrated variance as a function of h for all three types of estimator. We are plotting for h on an interval from 0 to 0.1 on many values to get a continuous function.



c) <u>Comparing Performance</u>

We can see from the figures that Gaussian and Epanechnikov are clearly better at estimating than Histogram. Also, comparing Gaussian to Epanechnikov, we can make a number of observation –

- Gaussian is usually more smooth compared to Epanechnikov for the same value of h. This can also be confirmed from the plot in 5.1(b), as the 'Integrated Variance' for Gaussian is always less than Epanechnikov for the same value of h.
- For Gaussian we get the best curve at h=0.07, but Epanechnikov curve at same 'h' value is showing more peaks(variance).
- Epanechnikov plot gets smooth as we further increase the value of 'h'.
- For selecting one of these estimators we would have to take into account how much variance we can accommodate.

5.2 Classification

a) Data Pre-processing
Refer the functions ttt_data_preprocess and nursery_data_preprocess from the code.

b) Implement Naive Bayes
Refer function naive_bayes from the code.

c) Implement K-NN
Refer the function knn_classify from the code.

d) Performance Comparison

- K-NN
  Below is the comparison of Accuracy for Training, Test and Validation dataset for different values of K. We can see a general trend here that the accuracy increases till we get to value (K=11), and then it decreases.

```
>> CSCI567_hw1_fall15
== Accuracy Reports for 5.2 (d) ==
KNN
---
```

K	Training_Data_Accuracy	Test_Data_Accuracy	Validation_Data_Accuracy
1	78.748	72.093	75.463
3	76.66	74.419	76.852
5	79.507	80	81.944
7	85.958	84.651	89.815
9	89.564	88.372	92.593
11	90.702	89.302	92.593
13	86.338	86.977	86.111
15	81.404	80.93	83.796

- Decision Tree

  Below is the comparison of Accuracy for Training, Test and Validation dataset using Decision Tree Algorithm with different values of Split Criterion and Min leaf, with prune set to false while training. Here we can notice that 'gdi'(Gini index) generally gives better accuracy than 'deviance'(Cross entrophy). This validates what we proved in question 3.b, that Gini index is a better approximation of the misclassification error.

Decision Tree

Split_Criterion	Min_Leaf	Train_Accuracy	Test_Accuracy	Valid_Accuracy
'gdi'	1	95.066	86.047	87.037
'gdi'	2	95.066	86.047	87.037
'gdi'	3	95.066	86.047	87.037
'gdi'	4	94.877	88.372	87.5
'gdi'	5	94.118	87.442	87.5
'gdi'	6	93.548	88.837	86.111
'gdi'	7	92.789	89.767	87.037
'gdi'	8	92.03	89.767	85.648
'gdi'	9	91.461	86.047	87.963
'gdi'	10	91.271	88.837	85.648
'deviance'	1	95.066	86.512	84.259
'deviance'	2	95.066	86.512	84.259
'deviance'	3	95.066	86.512	84.259
'deviance'	4	94.877	88.372	84.722
'deviance'	5	93.928	86.977	85.648
'deviance'	6	93.359	88.372	84.259
'deviance'	7	92.6	89.302	85.185
'deviance'	8	91.841	89.302	83.796
'deviance'	9	88.235	82.326	82.407
'deviance'	10	88.046	85.116	80.093

- Naive Bayes

  Below is the accuracy we get by using the Naïve-Bayes algorithm on both the data sets. There is a significant difference between the accuracies of two datasets (~20%). This is because of the inherent nature of the dataset. In tic-tac-tow dataset we have end game configuration and then a corresponding binary class label (positive or negative). But, our of the 9 features, the change in even one of them can impact whether the result is positive or negative. This is the reason why accuracy may be low.

  In contrast the nursery dataset has many categorical features and classifying it gives high accuracy, because each feature has some meaningful impact on the end data.

NAIVE BAYES

dataset	Training_Data_Accuracy	Test_Data_Accuracy	Validation_Data_Accuracy
'TTT'	70.968	70.698	71.296
'Nursery'	92.541	92.168	91.397

e) Decision Boundary

Below are the plots for the decision boundary for different values of K. As we can observe from the figure, for lower values of K (like 5, 15), our graph is very sparsely populated. But as the K value increases, the graph get much smoother.