Sri Hari Malla
Indian Institute of Technology Hyderabad
Fundamentals of Machine Learning
CS19BTECH11039

Assignment-I

Sri Hari Malla - CS19BTECH11039

September 24, 2021

1 k - Nearest Neighbours

Given n points separated into two classes of data each of n/2 points, which are overlapped to some extent in a 2-dimensional space.

1.1 Variation in Training error while k changes from n to 1:

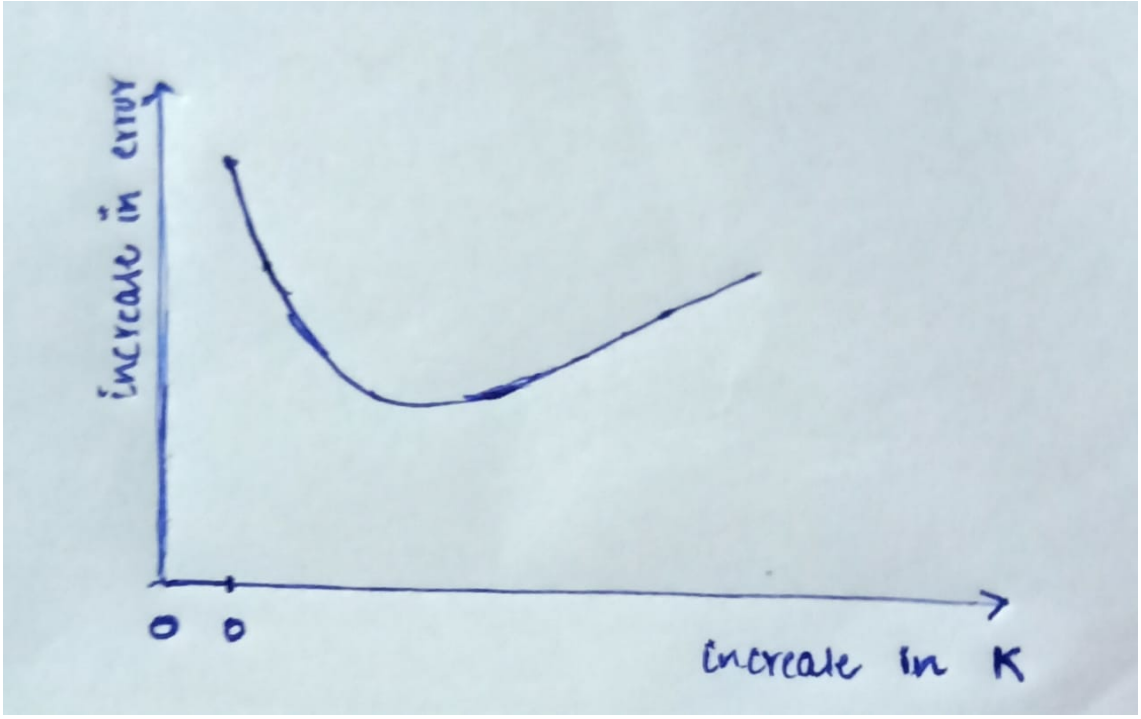
The training error is defined as the average error that results from using a method to predict the responses on the training set, the very same set which is used to train the method.

- k = n: As the model trains on all the points in the data set, we get the mean error and now we calculate the training error on all the data sets again. Which is the sum of squares of differences of each point with the mean point, which is variance.
- k = 1: As the model trains on only one neighbour, and is tested on the same neighbour again, there will be no mistake made. So the training error will be 0.
- 1 < k < n: We can't really have an idea on this case unless the points are given. It depends on the location of the point. One can say that the training error decreases while k runs from n to 1.

1.2 Generalisation Error:

- When k =1, the generalisation (or test) error is very high because only one closer example is taken and this means that the rest of data is essentially discarded.
- As k increases, up to a point, more data in the training set is utilised to make prediction and hence the error reduces. This is where bias reduces essentially.
- But after a point, more than necessary training points are considered, so the model fits more on training set and poorly generalises. So the generalising error again raises. This amounts to increase in variance.

- Finally when k reaches to n, the error will not be as high as in the case of k = 1. The reason being that model will train with some k > 1, model gets trained with many points and definitely the model will have less error prone to the initial case.



1.3 k-NN and High dimensions:

- Consider a data set in multi dimensions of very high order. Now the euclidean distance is almost helpless in this kind of situations because, all the other vectors are equidistant from the point containing vector. It mainly hinges on the point that data points being closer together which becomes a limitation when we deal with high dimensions. As dimensions increases, the closest distance between two points approaches the average distance between points, which makes the k-NN algorithm to work in an inefficient way and no valuable predictions are made.

- Storage concerns and computationally expensive, One needs to keep all the memory to get the model running. Basically it doesn't learn much, but stores all the training instances and does comparisons at the training or testing time. So while the dimensions increase, the data points increases exponentially, and as a result the method becomes slower and slower.

1.4 Univariate Decision tree?

We can't build a uni variate decision tree based on the given constraint.

The decision Boundaries for 1-NN corresponds to each point's cell boundary. They need not be parallel to the co-ordinate axes as defined in the question and can be in any orientation. The decision tree boundaries were always parallel to the co-ordinate axes, so does the uni variate decision tree. In order to approximate, it can take any number of decisions, so we can't use the decision tree here.

2 Bayes Classifier:

2.1

According to the Bayes theorem on classification probability, we have the formula:

$$p\left(c_j \mid x\right) = \frac{p\left(x \mid c_j\right) p\left(c_j\right)}{p(x)}$$

The above can be written as:

$$= \frac{p\left(x \mid c_j\right) p\left(c_j\right)}{\sum_{k=1}^{K} p\left(x \mid c_k\right) p\left(c_k\right)}$$

In this case, we have the Gaussian likelihood distribution formula:

$$p\left(x \mid c_j\right) = \frac{1}{\sqrt{2\pi\sigma_j^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_j}{\sigma_j}\right)^2}$$

Given values, in the question:

$$\hat{\sigma_1^2} = 0.0149, \implies \hat{\sigma_1} = 0.122$$

$$\hat{\sigma_2^2} = 0.0092, \implies \hat{\sigma_2} = 0.095$$

We have ML parameter estimators as:

$$\hat{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i$$

$$\hat{p}\left(c_j\right) = \frac{N_j}{\sum_{k=1}^{N} N_k}$$

On computing from the above formulas, we get;

$$\hat{\mu}_1 = \frac{\sum_{i=1}^{10} n_{1i}}{10} = \frac{2.6}{10} = 0.26,$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^{4} n_{2i}}{4} = \frac{3.54}{4} = 0.8625$$

$$\hat{p_1} = \frac{10}{14} = 0.714,$$
$$\hat{p_2} = \frac{4}{14} = 0.286,$$

$$\text{Distribution of class1 } = 3.27e^{\frac{-1}{2}\left(\frac{x-\mu_1}{\sigma_1}\right)^2}$$

$$\text{Distribution of class1 } = 4.16e^{\frac{-1}{2}\left(\frac{x-\mu_2}{\sigma_2}\right)^2}$$

$$p\left(c_1 \mid x = 0.6\right) = \frac{p(x=0.6|c_1)p(c_1)}{p(x=0.6|c_1)p(c_1)+p(x=0.6|c_2)p(c_2)}$$

$$p\left(x = 0.6 \mid c_1\right) p\left(c_1\right) = 3.27e^{\frac{-1}{2}\left(\frac{0.6-0.26}{0.122}\right)^2} \times 0.714 = 0.048$$

$$p\left(x = 0.6 \mid c_2\right) p\left(c_2\right) = 4.16e^{\frac{-1}{2}\left(\frac{0.6-0.8625}{0.0959}\right)^2} \times 0.286 = 0.028$$

$$\implies p\left(c_1 \mid x = 0.6\right) = 0.6309$$

2.2 Maximum likelihood Naive Bayes classifier:

We know that,

$$p\left(c_p \mid \vec{x}\right) = \frac{p(\vec{x}|c_p)p(c_p)}{p(\vec{x}|c_p)p(c_p)+p(\vec{x}|c_s)p(c_s)}$$

and also,

$$p\left(\vec{x} \mid c_s\right) = p\left(x_1 \mid c_s\right) \ldots p\left(x_8 \mid c_s\right)$$

$$\text{but } p\left(x_7 = 1 \mid c_s\right) = 0$$

Which means,

$$\implies p\left(\vec{x} \mid c_s\right) = 0$$

Now going to the above mentioned formula from present observation,

$$p\left(c_p \mid \vec{x}\right) = \frac{p(\vec{x}|c_p)p(c_p)}{p(\vec{x}|c_p)p(c_p)+0}$$

$$p\left(c_p \mid \vec{x}\right) = \frac{p(\vec{x}|c_p)p(c_p)}{p(\vec{x}|c_p)p(c_p)} = 1$$

$$\implies p\left(c_p \mid \vec{x}\right) = 1$$