Assignment 1

Dishank Jain
AI20BTECH11011

1 k-NN: (8 marks)

In **k-NN**, the classification is achieved by a majority vote in the vicinity of the data. Given n points, imagine two classes of data each of $n/2$ points, which are overlapped to some extent in a 2-dimensional space.

1.1 (1 mark)

Describe what happens to the training error (using all the available data) when the neighbour size k varies from n to 1.

1.1.1 Answer

When $k = 1$, the nearest neighbour to any data point will be the point itself. Therefore $training\ error = 0$. When $k \to n$, the classification will not be very accurate. In particular, when $k = n$, the classification will be random. Therefore $training\ error \to 0.5$ as $k \to n$. We can expect a generally increasing training error for the rest of the values of k.

1.2 (2 marks)

Predict and explain with a sketch, how the generalization error would change when k varies. Explain your reasoning.

1.2.1 Answer

In k-NN, we work with the assumption that the label of any record is same as the label of nearby records. However there is expected to be some noise associated with every record. Therefore the location of every record would slightly change from it's original location. Therefore we take a majority vote of the k nearest neighbours. However, similar to training error, if k is close to n, the generalization error will approach 0.5. If k is small, there will a higher effect of noise on the classification. Therefore we can expect an optimum value of k where the generalization error will be minimum. Figure 1 shows the expected sketch. On small k, we expect under-fitting. On large k, we expect over-fitting.

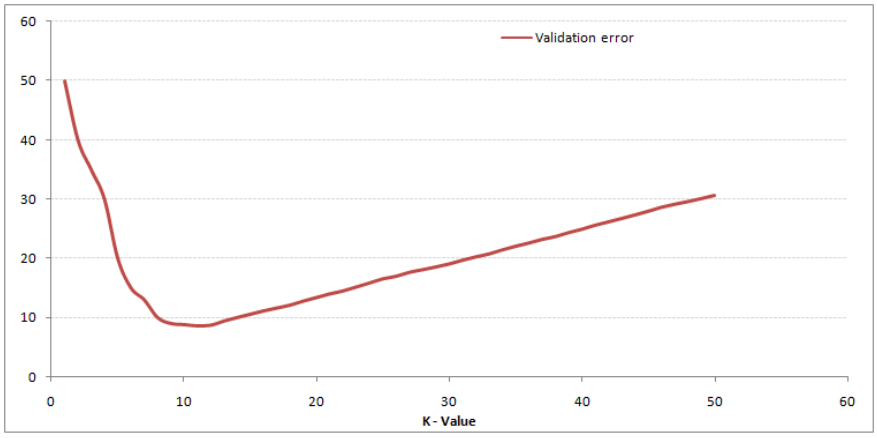

Figure 1: expected graph of generalization error

1.3 (2 marks)

Give two reasons why k-NN may be undesirable when the input dimension is high.

1.3.1 Answer

1. **Curse of dimensionality:** On high dimension input, every data point is very unique because of the enlarged feature space. Therefore the euclidean distance between records is very high. We started with the assumption that label of any new record is same as label of nearby records. In high dimensional data, there are no nearby records. Even data points in the same class are at a large distance. Therefore, either we need to have a very large dataset or we need to use some other model.

2. **High computation time:** If we choose to use a large dataset to overcome the above problem, the next problem we face is a high computation time. As we have to find the distance of a new record with every other record, the time complexity will be **O(nm)**, where m is the number of dimennsions. This becomes very high in practice if m is large.

1.4 (3 marks)

Is it possible to build a univariate decision tree which classifies exactly similar to a 1-NN using the Euclidean distance measure? If so, explain how. If not, explain why not.

1.4.1 Answer

No. In general, it is not possible to build a univariate decision tree which classifies exactly similar to a 1-NN classifier.

Imagine a simple classification problem with two records, A at (-1,1) and B at (1,-1). Suppose A belongs to class P and B belongs to class Q. Then a 1-NN classifier will classify any point above the line $y = x$ to class P and any point below $y = x$ to class Q.

However, a univariate decision tree can create only vertical or horizontal decision boundaries. Therefore, we will never be able to recreate the decision boundary $y = x$.

2 Bayes classifier: (6 marks)

2.1 (3 marks)

A training set contains of one dimensional examples from two classes. The training examples from class 1 are {0.5, 0.1, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.35, 0.25} and from class 2 are 0.9, 0.8, 0.75, 1.0. Fit a Gaussian using Maximum Likelihood to each of these two classes. Assume that variance for class 1 is 0.0149, and the variance for class 2 is 0.0092. Also estimate the class probabilities p_1 and p_2 using Maximum Likelihood. What is the probability that the test point $x = 0.6$ belongs to class 1?

2.1.1 Answer

The probability density function for a Gaussian distribution is given by

$$f(x) = \frac{1}{\sqrt{(2\pi\sigma^2)}} exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

For Maximum likelihood estimation, we have to maximize $f(x_1)f(x_2)...f(x_n)$. In other words, we have to maximize $\Sigma_i log(f(x_i))$. On taking derivative and equating it to zero, we find that

$$\mu = \frac{x_1 + x_2 + ... + x_n}{n}$$

Therefore, for Class 1,

$$\mu = \mu_1 = \frac{0.5 + 0.1 + 0.2 + 0.4 + 0.3 + 0.2 + 0.2 + 0.1 + 0.35 + 0.25}{10} = 0.26$$

For Class 2,

$$\mu = \mu_2 = \frac{0.9 + 0.8 + 0.75 + 1.0}{4} = 0.8625$$

We have been given σ in the question as

$$\sigma_1^2 = 0.0149;\ \sigma_2^2 = 0.0092$$

By finding the above parameters, we have fit the Gaussians on the classes. Next we find the class probabilities p_1 and p_2. We can say that

$$p_i = \frac{Number\ of\ entries\ in\ class\ i}{Total\ number\ of\ entries\ in\ the\ dataset}$$

Therefore

$$p_1 = \frac{10}{14};\ p_2 = \frac{4}{14}$$

Finally,

$$P(class\ 1|X = 0.6) = \frac{P(X = 0.6|class\ 1)p_1}{P(X = 0.6|class\ 1)p_1 + P(X = 0.6|class\ 2)p_2}$$

$$P(X = 0.6|class\ 1) = \frac{1}{\sqrt{(2\pi\sigma_1^2)}} exp(-\frac{(0.6 - \mu_1)^2}{2\sigma_1^2}) = 0.0675$$

$$P(X = 0.6|class\ 2) = \frac{1}{\sqrt{(2\pi\sigma_2^2)}} exp(-\frac{(0.6 - \mu_2)^2}{2\sigma_2^2}) = 0.0983$$

$$\implies P(class\ 1|X = 0.6) = 0.6319$$

2.2 (3 marks)

A set of documents is represented as row vectors, one row corresponding to each document. Each document either belongs to sports or politics. The data is as follows

$$x_{politics} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$x_{sports} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Using a maximum likelihood naive bayes classifier, what is the probability that the document $x = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$ is about politics?

2.2.1 Answer

$$P(politics|x) = \frac{P(politics)P(x|politics)}{P(politics)P(x|politics) + P(sports)P(x|sports)}$$

Using Naive assumption of independence of attributes

$$P(\begin{pmatrix}1 & 0 & 0 & 1 & 1 & 1 & 1 & 0\end{pmatrix}|politics) = \frac{2}{6} \times \frac{5}{6} \times \frac{5}{6} \times \frac{5}{6} \times \frac{5}{6} \times \frac{5}{6} \times \frac{4}{6} \times \frac{5}{6}$$

$$P(\begin{pmatrix}1 & 0 & 0 & 1 & 1 & 1 & 1 & 0\end{pmatrix}|sports) = \frac{4}{6} \times \frac{2}{6} \times \frac{5}{6} \times \frac{4}{6} \times \frac{1}{6} \times \frac{1}{6} \times \frac{0}{6} \times \frac{5}{6} = 0$$

$$\implies P(politics|\begin{pmatrix}1 & 0 & 0 & 1 & 1 & 1 & 1 & 0\end{pmatrix}) = 1$$

3 Decision Trees: (16 marks)

(5 marks) Implement your own version of decision trees on the wine dataset using binary univariate split, entropy and information gain. (5 marks) Implement 10-fold cross validation on your model and report the accuracy. (6 marks) Improve your decision tree using any two improvement strategies and report the accuracy.

3.1 Answer

The code can be found in the attached .ipynb notebook. The accuracy of the initial implementation was found to be 80.81 %. The accuracy after using Gini-index and pruning was found to be 81.90 %. Pruning was applied in a way that if probability of any class at a node is greater than 90 %, then the node becomes a leaf node. This is pre-pruning. This decreased the effects of noise. Therefore the accuracy improved. Do not that I implemented stratified randomized sampling. Therefore the accuracy will be different on each run. I have reported accuracy of one such run. Use of Gini-index did not have much impact on the accuracy.