Assignment 1

Gulshan Hatzade

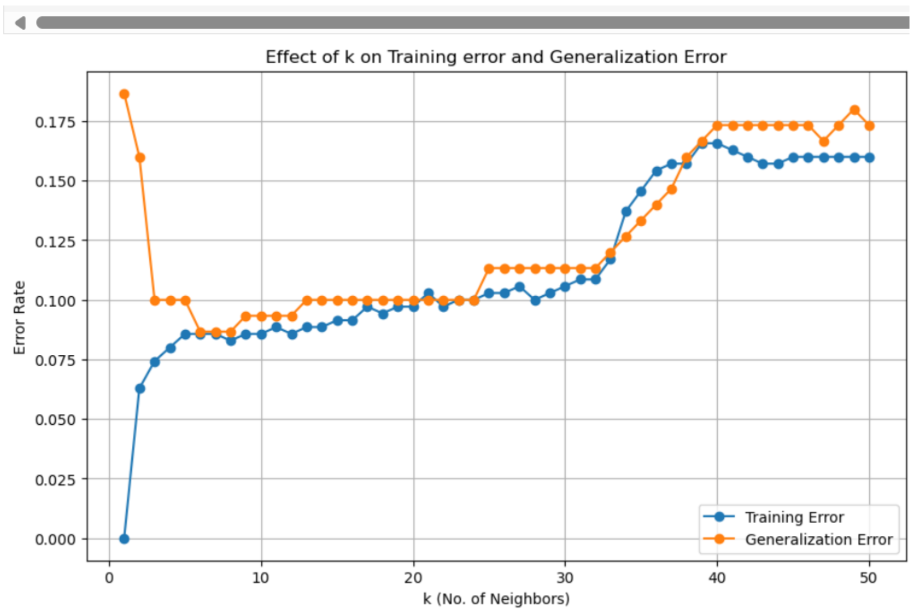
September 2024

Question 1 k-NN)

Solution a)

When the neighbour size varies from n to 1, with the larger k it is considering more points in the vicinity of each data point, which may cause misclassifivation and as when k decreases the model is considering fewer points which may reduce misclassification.
When k=n, the model considers all points means it predicts majority class, which results in high training error.
When k=1, model considers only single nearest point which causes low training error.
So, when the neighbour size varies from n to 1 the training error will decrease as k decreases.

Solution b)



The generalization error follows U shaped curve as the value of k varies.
When k is small, it results in high generalization error as in stretch due to overfitted model.
When the value of k increases, it results in decreasing generalization error which can be seen in strech due to the model starts to smooth out noise and capture the underlying pattern.
But when k is close to n, it results in an increase in generalization error due to model starts underfitting, because it is considering too many points, which increase error.

Solution c)

No, it is not possible to build a univariate decision tree which classifies exactly similar to 1-NN by using euclidean distance measure.
Univariate decision tree can only make decisions based on single feature whereas in 1-NN for calculating euclidean distance both features simultaneously.
For univariate decision tree the decision boundry is vertical or horizontal line and for 1-NN the decision boundry is a voronoi diagram which cannot be represented by a single decision tree as it has more complex boundries.
As univariate decision tree can only make decisions on single feature, it is not possible to build a univariate decision tree which classifies exactly similar to

1-NN using the euclidean distance measure.

Que 2)

Solution a)

a) Fit a Gaussian to each class

Class 1:

Training examples: $\{0.5, 0.1, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.35, 0.25$

Calculating the mean for Class 1-

$$Mean_1 = \frac{0.5 + 0.1 + 0.2 + 0.4 + 0.3 + 0.2 + 0.2 + 0.1 + 0.35 + 0.25}{10}$$

$$= 0.26$$

Given variance in question -

$$Variance_1 = 0.0149$$

Class 2:

Training examples: $\{0.9, 0.8, 0.75, 1.0\}$

Calculating the mean for Class 2:

$$Mean_2 = \frac{0.9 + 0.8 + 0.75 + 1.0}{4}$$

$$= 0.8625$$

Given variance in question-

$$Variance_2 = 0.0092$$

b) Estimate Class Probabilities

Probability of Class 1:

$$P_1 = \frac{\#Class1points}{\#Totalpoints} = \frac{10}{14} = 0.714$$

Probability of Class 2:

$$P_2 = \frac{\#Class2points}{\#Totalpoints} = \frac{4}{14} = 0.286$$

c) Calculate Probability for $x = 0.6$ by Gaussian PDF

For Class 1:

$$P(x = 0.6 \mid Class1) = \frac{1}{\sqrt{2\pi \cdot Variance_1}} \exp\left(-\frac{(x - Mean_1)^2}{2 \cdot Variance_1}\right)$$

Substitute values in above equation for class 1

$$P(x = 0.6 \mid Class1) = \frac{1}{\sqrt{2\pi \cdot 0.0149}} \exp\left(-\frac{(0.6 - 0.26)^2}{2 \cdot 0.0149}\right)$$

$$= \frac{1}{0.306} \exp\left(-\frac{0.1156}{0.0298}\right)$$

$$= 3.3 \cdot \exp(-3.878)$$

$$= 3.3 \cdot 0.02069 = 0.0682$$

For Class 2:

$$P(x = 0.6 \mid Class2) = \frac{1}{\sqrt{2\pi \cdot Variance_2}} \exp\left(-\frac{(x - Mean_2)^2}{2 \cdot Variance_2}\right)$$

Substitute values in above equation for class 2, solving for P(x=0.6— Class 2)

$$P(x = 0.6 \mid Class2) = \frac{1}{\sqrt{2\pi \cdot 0.0092}} \exp\left(-\frac{(0.6 - 0.8625)^2}{2 \cdot 0.0092}\right)$$

$$= \frac{1}{0.241} \exp\left(-\frac{0.068}{0.0184}\right)$$

$$= 4.1493 \cdot \exp(-3.745)$$

$$= 4.1493 \cdot 0.02363 = 0.098$$

Bayes' Theorem: Probability that $x = 0.6$ Belongs to Class 1

$$P(Class1 \mid x = 0.6) = \frac{P(x = 0.6 \mid Class1) \cdot P_1}{P(x = 0.6 \mid Class1) \cdot P_1 + P(x = 0.6 \mid Class2) \cdot P_2}$$

$$= \frac{0.0682 \cdot 0.714}{0.0682 \cdot 0.714 + 0.098 \cdot 0.286}$$

$$\frac{0.0486}{0.0486 + 0.028} = \frac{0.0486}{0.0766} = 0.634$$

So, required probability for x=0.6 belongs to class 1 is 0.634

Solution b)

Prior Probabilities for Document Classification

Number of sport documents=6
Number of politics documents=6
Total documents =12
Calculating Prior Probabilities:

$$Priorprobabilityforsport = \frac{\#sportdocuments}{\#Totaldocuments} = \frac{6}{12} = 0.5$$

$$Priorprobabilityforpolitics = \frac{\#politicsdocuments}{\#Totaldocuments} = \frac{6}{12} = 0.5$$

Calculating Likelihoods:
Our given document is x =(1,0,0,1,1,1,1,0) **Likelihood for politics:**

$$P(goal = 1 \mid politics) = \frac{2}{6}, \quad P(football = 0 \mid politics) = \frac{5}{6}, \quad P(golf = 0 \mid politics) = \frac{5}{6}$$

$$P(defence = 1 \mid politics) = \frac{5}{6}, \quad P(offence = 1 \mid politics) = \frac{5}{6}$$

$$P(wicket = 1 \mid politics) = \frac{1}{6}, \quad P(office = 1 \mid politics) = \frac{4}{6}, \quad P(strategy = 0 \mid politics) = \frac{1}{6}$$

Likelihood for sports:

$$P(goal = 1 \mid sports) = \frac{4}{6}, \quad P(football = 0 \mid sports) = \frac{2}{6}, \quad P(golf = 0 \mid sports) = \frac{5}{6}$$

$$P(defence = 1 \mid sports) = \frac{4}{6}, \quad P(offence = 1 \mid sports) = \frac{1}{6}$$

$$P(wicket = 1 \mid sports) = \frac{1}{6}, \quad P(office = 1 \mid sports) = \frac{0}{6}, \quad P(strategy = 0 \mid sports) = \frac{5}{6}$$

article amsmath
Given:

$$P(x|politics) = \left(\frac{2}{6}\right) \times \left(\frac{5}{6}\right) \times \left(\frac{5}{6}\right) \times \left(\frac{5}{6}\right) \times \left(\frac{5}{6}\right) \times \left(\frac{1}{6}\right) \times \left(\frac{4}{6}\right) \times \left(\frac{1}{6}\right)$$

$$P(x|politics) = 0.0297$$

$$P(x|sports) = \left(\frac{4}{6}\right) \times \left(\frac{2}{6}\right) \times \left(\frac{5}{6}\right) \times \left(\frac{4}{6}\right) \times \left(\frac{1}{6}\right) \times \left(\frac{1}{6}\right) \times \left(\frac{0}{6}\right) \times \left(\frac{5}{6}\right) = 0$$

Using Bayes' Theorem to calculate the posterior probability:

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

Substitute the values:

$$P(politics|x) = \frac{(0.0297 \times 0.5)}{(0.0297 \times 0.5) + (0 \times 0.5)}$$

$$P(politics|x) = 1$$

Probability that document x=(1,0,0,1,1,1,1,0) is about politics is 1.

Que 3)

Solution c)

The accuracy of initial implementation is 0.8868 and the accuracy of improved implementation is 1.0.
The improvement in the performance is due to hyperparameter tuning and randomm forest ensemble method where in the random forest ensemble for producing strong it combines multiple decision trees and it helps in decreasing overfitting means reduced variance.
The another method of hyperparameter tuning uses GridSearchCV method which test multiple combinations of hyperparameter and then detects the best parameters that reults in maximum accuracy.

Que 4)

Solution a)

1)K Nearest Neighbours
Traning Complexity- $O(1)$
k-NN does not require training phase, it is a lazy learning algorithm.
KNN does not build a model, it sotes entire dataset and waits to find k nearest neighbors till inference time.
Inference Complexity- $O(n * d)$
Here, n =number of instances
d = number of features

KNN requires to iterate over entire dataset for calculating distance between test instance and all training instances. It have to iterate linearly over all instances and features.

2) Decision Tree:

Training Complexity-$(n * d * \log n)$
Where, $n =$ number of instances
$d =$ number of features
Recursively partitioning the data into small subsets considering the most informative features the decision trees are obtained. Recursively partitioning process dominates in time complexity, which is $\log n$ with respect to number of instances.
Inference Complexiy- $O(d)$
Inference involves iterating over features only once for traversing decision tree from root node to leaf node. So with respect to number of features, it requires linear time complexity.

3) Naive Baye's:

Training Complexity-$O(n * d)$
Where $n =$ no. of instances,
$d =$ no. of features
It involves iterating over entire dataset for computing probability of each feature. So, it takes linear time complexity with respect to features and number of instances.
Inference Complixity-$O(c * d)$
Here c is no. of classes and d is no. of features. During inference, it involves iterating over features only once for computing the posterior probability of each class label given the instance.

Solution b)

1)Large Dataset- large number of Features

a)KNN
Strength: It dont require training phase and it is easy to implement.
Weakness:It needs to compute distances to all training points, so high inference tie. IF stores all training data, so it is memory intensive.

b)Decision Tree
Strength: Able to handle large datasets with high dimensional data and provides interpretable models.
Weakness: May overfit with high dimensional data and may become very complex.

c)Naive Bayes
Strength: Due to its simplicity, it is efficient even with large number of features.
Weakness: Can lead to suboptimal performance due to its assumption of independence between features.

2)Large Dataset- small number of Features

a)KNN
Strength: Reduced distance computation time when small number of features.
Weakness: High inference time due to large dataset size.

b)Decision Tree

Strength: Handles large datasets well and it is efficient.

Weakness: If not properly pruned, it can overfit.

c)Naive Bayes

Strength: Probability estimates are simplified by small feature space leads t better performance.

Weakness: If independence assumption is violated, it may suffer.

3)Small Dataset- large number of Features

a)KNN

Strength: Due to number of training points are small, it might overfit less.

Weakness: Less meaningfulness of distance measures can degrade performance.

b)Decision Tree

Strength: It provides intepretable models.

Weakness: Tree can perfectly memorize small database, so likely to overfit with many features.

c)Naive Bayes

Strength: Fast training and inference and it handles high dimensional data well.

Weakness: If independence assumption is violated, then performance can suffer.

4)Small Dataset- small number of Features

a)KNN

Strength: Due to lower features it can work well and fast inference.

Weakness: Depending upon choice of k, performance is sensitive.

b)Decision Tree

Strength: Overfitting can be easily avoided by pruning and can easily model the data.

Weakness: With very small dataset, it may overfit.

c)Naive Bayes

Strength: Lower complexity of both dataset and model makes it very effective.

Weakness: If assumps independence of features which doesn't meet requirement.