Weekly Report 2 - KNN

Ganji Varshitha
AI20BTECH11009

Introduction

KNN algorithm, also known as Lazy learning is a non parametric model which is used to solve classification and regression problems.

Algorithm

It follows the instance based learning as the learning part involves storing all the data points and performs the action at the time of classification or regression. Choosing the metric and value of K is very crucial in this model.

Algorithm 1 KNN Algorithm

Calculate distance of new point from all the training data points.
Sort the distances in increasing order with corresponding train data and select first K data points.
if Classification **then**
    Class is determined by the majority class of the selected K data points.
end if
if Regression **then**
    The predicted continuous target value is given by mean or median value of the K data points.
end if

How to determine K ?

Without loss of generality, K needs to be odd number to avoid ties in binary classification. For small values of K, the model has high variance and overfits the data. For example K = 1, the model is highly sensitive to outliers. Training error is low whereas test error is high.
For high values of K, model has less variance and increased bias.
We can determine optimal value of K by iterating through various values of K and choosing the one with minimum error rate.

Distance metrics

Euclidean distance is the most common metric used. Besides, there are Manhattan, Minkowski and hamming distance metrics used when needed.

Key points

- Since calculating distances involve all the features, the scale of the features may affect the outcome as one feature may dominate the rest. Therefore, we need to normalize the features.
- The computational cost is very high as it involves calculating distances from all the training data points.
- Training is fast whereas classification task is time consuming.
- After finding the optimal value of K, the model is robust to outliers.

Questions

1. Does KNN work for large dataset?

   A. Yes

   B. No

   Solution:
   B. No

2. State Manhattan distance formula.
   Solution:

$$D_m(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

3. What can be done to overcome curse of dimensionality?
   Solution:
   As number of dimensions increases, size of data space increases. This reduces the density of data. It can be avoided by reducing the number of features by dimensionality reduction.

4. Hamming distance is useful for _____

   A. Numerical features

   B. Categorical features

   Solution:
   Since hamming distance tells number of indifferent features, it is helpful in categorical input.

5. Time complexity of the algorithm is _____
   Solution:
   Assume K cluster for N data points having D dimensions.
   Time complexity is: $\mathcal{O}(\#\text{iter} * K * N * D)$