Pattern Recognition Assignment 2

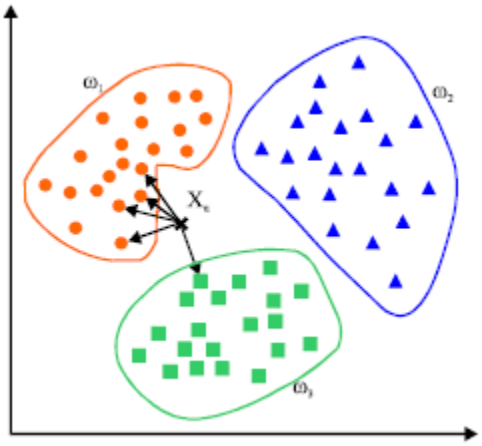
K- nearest Neighbour using Grid Search

Submitted By : Shreya Raj 2K18/SE/121
Yash Gupta 2K18/SE/136

Introduction:

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that is used to solve both classification and regression problems. It assumes that similar things exist in close proximity. In other words, similar things are near to each other.

Using K-Nearest Neighbour, we predict the category of the test point from the available class labels by finding the distance between the test point and trained k nearest feature values.



Classification using K-Nearest Neighbour

Objective:

(1) Load the satellite dataset and use tsne plot to visualize the distribution of classes.
(2) Implement the kNN algorithm from scratch.We need to find the optimal number of k using the grid search. Plot the error vs number of neighbors graph (k) and report the optimal number of neighbours.
(3) Report the training and the validation accuracy only with optimal value of k using sklearn kNN function. Comment on the accuracy obtained for optimal value of k for both the methods i.e, our implementation and the inbuilt sklearn function.

Theory:

KNN works by finding the distances between a given data point, a query and all the remaining examples in the sample, selecting the specified number examples (K) closest to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

Choosing the right value for K

To select the K that's right for our data, we run the KNN algorithm several times with different values of K and choose the K that reduces the errors we encounter while maintaining the algorithm's ability to accurately make predictions for new data.This can be easily done with the help of GridSearch.

Grid Search for Hyperparameter Tuning

Grid Search is the process of performing hyperparameter tuning in order to determine the optimal values for a given model.We know that the performance of a model significantly depends on the value of hyperparameters.Since there is no way to know in advance the best values for hyperparameters so ideally, we need to try all possible values to know the optimal values. Doing this manually could take a considerable amount of time and resources and thus we use GridSearchCV to automate the tuning of hyperparameters.

Dataset: https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)

Statlog (Landsat Satellite) Data Set is a collection of multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. The aim is to predict this classification, given the multi-spectral values. In the sample database, the class of a pixel is coded as a number.

K-nn Algorithm:

1. Load the data
2. Initialize K to your chosen number of neighbors
3. For each example in the data

3.1 Calculate the distance between the query example and the current example from the data.

3.2 Add the distance and the index of the example to an ordered collection

4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels

8. If classification, return the mode of the K labels

To classify an unknown record:

- Initialize the K value.
- Calculate the distance between test input and K trained nearest neighbors.
- Check class categories of nearest neighbors and determine the type in which test input falls.
- Classification will be done by taking the majority of votes.
- Return the class category.

How to find the optimal value of k?

For a very low value of k (suppose k=1), the model overfits on the training data, which leads to a high error rate on the validation set. On the other hand, for a high value of k, the model performs poorly on both train and validation set.

To overcome this problem, and find the best suitable k value we have used Grid Search for tuning hyperparameters for the K-nn classifier

Distance Metrics:

The distance metric is the effective hyper-parameter through which we measure the distance between data feature values and new test inputs.

Minkowski Distance – It is a metric intended for real-valued vector spaces. We can calculate Minkowski distance only in a normed vector space, which means in a space where distances can be represented as a vector that has a length and the lengths cannot be negative.

Manhattan Distance – This distance is also known as taxicab distance or city block distance, that is because the way this distance is calculated. The distance between two points is the sum of the absolute differences of their Cartesian coordinates.

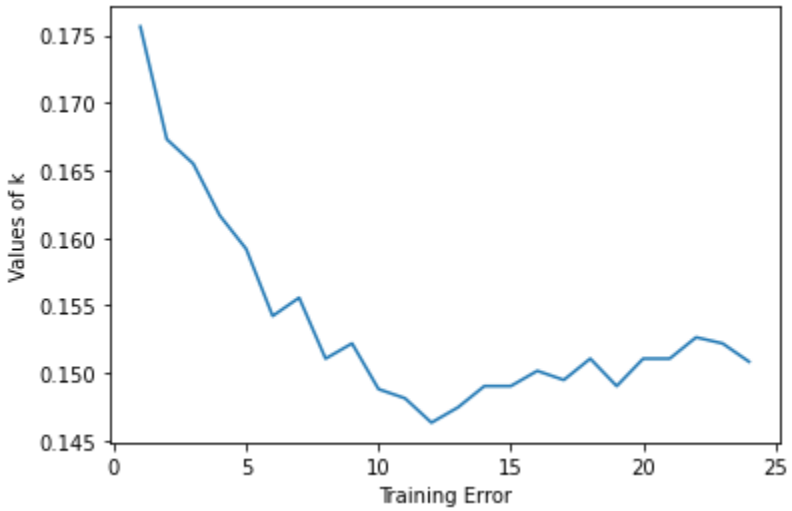
Euclidean Distance – This distance is the most widely used one as it is the default metric that SKlearn library of Python uses for K-Nearest Neighbour. It is a measure of the true straight line distance between two points in Euclidean space.

We have implemented the classifier from scratch and used **Euclidean distance** as a measure to decide k nearest neighbours.Next, we have compared the classification accuracy of our model with sklearn's KNearestNeighbour library to evaluate the model and dataset.

Results:

We found out that the optimal value of k using Grid Search came out to be 12. Below is a plot of training error vs k values. And it is clearly observed that k=12 is the best hyperparameter we can set to increase the accuracy of our model.Training accuracy reported is 0.853664036076663.

```
Best parameters from Grid Search {'n_neighbors': 12}
Training Accuracy 0.853664036076663
```



When we tested the model on test data, the accuracy came out to 0.8965 , which is in fact comparable to the training accuracy. So we can say that our model has learnt well from the training samples and used to correctly predict classes in test data.

Also , the test accuracy is the same in both cases when we used sklearn's inbuilt function or we built our own K-nn classifier from scratch.This suggests that we have successfully implemented the K nearest neighbour classifier from scratch.