Final project

Timur Musaev

COSC4345-001

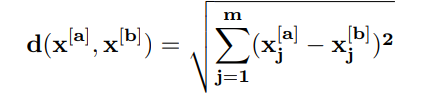
**1.Task**

– Create and Test a kNN classifier on the following given corpora.

• Dtrain = TrainingData.csv

• Dtest = TestData.csv – Your system should have the following features.

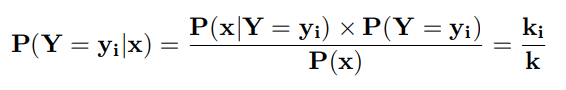
• The distance between two data instances x [a] ∈ R^m and x [b] ∈ R^m is:



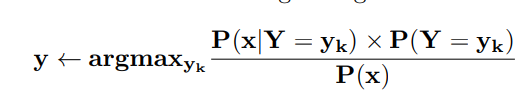
• A max priority queue which uses a max heap data structure should be created to store K nearest neighbors

• The time complexity of your system runs on T(n) = O(nlog2k), where n is the number of data instances in the training set and k is the number of nearest neighbors of a new data instance x.

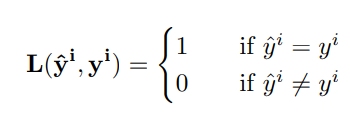
• Predicate the class label of a new data instance x among k nearest neighbors by Bayesian perspective,



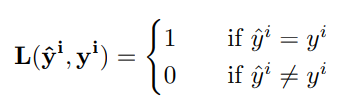
where ki is the number of data instances among k neighbors and k is k nearest neighbors.



• Evaluate your system by accuracy measurement,



where ˆy i is the assigned class label by the classifier and y i is the true class label of a data instance x [i] in Dtest and T is the number of data instances in Dtest.



**2. Algorithm**

Store training dataset

Store testing dataset

Define distance calculating function d(pass datasets)

For j=1, ……., n

For i=1, ……., n

For l=0, ……, 13

Distance=distance+sqrt((train[i][j]-test[l][j])\*\*2)

Append distance to max-heap

Define build-max-heap(A,n)

For i= n/2 downto 1

Max-heapify(A,i,n)

Define knn

Counter1=0

Counter2=0

Call d(detests)

If class label of k nearest points =1

Counter1++

If class label of k nearest points=-1

Counter2++

If counter1>counter2

Set predicted label to 1

Else

Set predicted label to -1

accuracy(datasets and predictions sets)

Define accuracy

Counter=0

For i=1, ………, n

If testset[i][14] == prediction [i]

Counter++

Print(Accuracy: counter\*100/n %)

Call knn

**3.Experiments and results discussion**

On average, increasing number of nearest neighbors was increasing accuracy level. The best accuracy that I was able to achieve ~82%. However pretty much every number of neighbors greater than 100 was giving me output accuracy around 70-80%. Compiling time was obviously increasing with more instances used. When I’ve tried using all values from training set, setting k nearest-neighbors value and number labels to predict to 1560 (amount of training set instances) I was excited for almost 37 minutes as I had to wait this whole time.

**4.Further improvements**

Due to long compiling speed, I did not check every possible combination to find the best values for n and k. Experimenting more with these two values would be beneficial for finding the most efficient values. Also, I might try to do it with the other more known to me programming language or practice more with python as feel like fluency in python would give me more options to perform the same tasks.

Also, while experimenting with number of x values from training data I was always taking first n elements, however, trying to use n random instances might change the prediction.