Applied Machine Learning

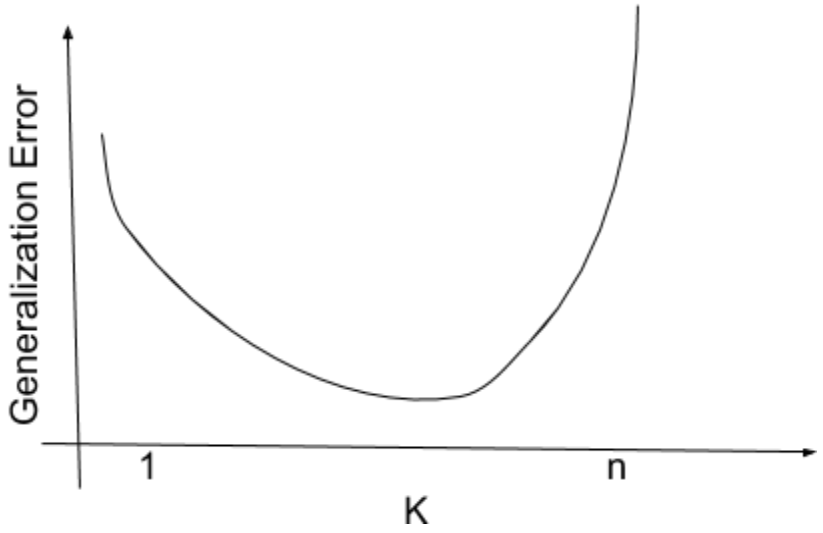
HW 1

Irfan Ali - CS16BTECH11019

Q1 :

(a) : The training error at k = 1 will be zero as every point is nearest to itself. There will be a sudden rise in training error from k =1 to k = 2. When k increases from 2 to n, training error will behave same as testing error i.e it first decrease and then increase. For k = n , training error will be 50 % as the kNN will give a random prediction.

(b) : The generalization error will first decrease and then increase as k increases from 1 to n. This will be because for low value of k, if the data is overlapping, it may give wrong answers. For large value of k, the prediction almost becomes random because it will take into account many points which may also contain points from opposite class.



(c) : When the input dimension is high, similar points may be far away in terms of euclidean distance. This may not be as intuitive as for 2,3 dimensions. Also one more reason is that the computation time will be huge as it is of O(dn) where d is the dimension.

(d) : No. It is not possible to build a univariate decision tree which classifies exactly similar to 1-NN using the euclidean distance. This can be explained by putting it as : Euclidean distance uses circles (assuming 2-dimensional space) around the points to get the nearest neighbours. But decision trees divide the space into a grid. So the classification of decision tree will not be same as 1-NN.

Q2 :

(a) : Class 1 : mean = 2.6/10 = 0.26

variance = 0.0149

Class 2 : mean = 3.45/4 = 0.8625

variance = 0.0092

p1 = 10/14 , p2 = 4/14

x = 0.6

$P(x/class1) = 1/\sqrt{2\pi * 0.0149} * e^{-(0.6-0.26)^2/(2*0.0149)} = 0.0675$

$P(x/class2) = 1/\sqrt{2\pi * 0.0092} * e^{-(0.6-0.8625)^2/(2*0.0092)} = 0.0983$

P(x) = P(x/class1)*p1 + P(x/class2)*p2 = 0.0482 + 0.0281 = 0.0763

P(class1/x) = P(x/class1)*p1/P(x) = 0.6319

Therefore, required probability is 0.6319

(b) : Using Laplace Smoothing:

P(x/sport)*P(sport)=

5/14 * 3/14 * 6/14 * 5/14 * 2/14 * 2/14 * 1/14 * 6/14 * 1/2 = 0.0000073/2

P(x/politics)*P(politics)=

3/14 * 6/14 * 6/14 * 6/14 * 6/14 * 2/14 * 5/14 * 6/14 * 1/2 = 0.0001581/2

P(x) = 0.0001654/2

P(politics/x) = P(x/politics) * P(politics) /P(x)
= 0.0001581/0.0001654 = 0.9559

∴ Probability of document x being about politics = 0.9559