Machine Learning Exercise Sheet 2

k-Nearest Neighbors and Decision Trees

Anastasia Stamatouli . 03710902

Martin Meinel · 03710370

27/10/2019

• KNN Classification

Problem ①:

L_1-norm $= \sum_i |u_i - v_i|$

L_2-norm $= \sqrt{\sum_i (u_i - v_i)^2}$

L_1-norm $d(A,B) = \left| \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix} - \begin{bmatrix} 2.0 \\ 0.5 \end{bmatrix} \right| = 1.5$

L_2-norm $d(A,B) = \sqrt{(1.0)^2 + (0.5)^2} = 1.12$

L_1-norm $d(A,C) = 1.5$

L_2-norm $d(A,C) = 1.5$

L_1-norm $d(A,D) = 4.5$

L_2-norm $d(A,D) = 3.201$

L_1-norm $d(A,E) = 7$

L_2-norm $d(A,E) = 5.14$

L_1-norm $d(A,F) = 6$

L_2-norm $d(A,F) = 4.74$

The nearest neighbor of A based on L_1 norm is B which is class 1. So class of A would be 1. L_2 norm also considers B as the closest neighbor. Therefore, in this case A would be classified as 1 too.

We follow the same process for the rest of the points. Inorder to speed up the process, and avoid spare calculations, we notice that the matrix of the distances is symmetric so by just calculating the distances A to all the points, B to all the points, C to all the points D to E (since the rest have already been calculated in previous steps, the same for D to F, and the same for E to F

The distances are as follows:

L1-norm:

	A	B	C	D	E	F
A	0	1.5	1.5	4.5	7.0	6.0
B	1.5	0	3.0	4.0	6.5	5.5
C	1.5	3.0	0	3.0	5.5	4.5
D	4.5	4.0	3.0	0	2.5	3.5
E	7.0	6.5	5.5	2.5	0	1.0
F	6.0	5.5	4.5	3.5	1.0	0

L2-norm:

	A	B	C	D	E	F
A	0	1.12	1.5	3.201	5.14	4.74
B	1.12	0	2.24	3.16	4.61	4.03
C	1.5	2.24	0	2.24	4.61	4.5
D	2.87	3.16	2.24	0	2.5	2.7
E	5.15	4.61	4.61	2.5	0	1.0
F	4.74	4.03	4.5	2.7	1.0	0

c)
We notice that in the case of D, L1-norm takes as nearest neighbor E. But L2-norm takes as nearest neighbor C. In the case of L1-norm D would be classified as 2 while in case of l2-norm as 1.

In general L2-norm is always the shortest distance to go from one point to another, which is also proved in our example.

In l2-norm the points which are further in distance are penalized harder (due to squared) while in l1-norm all the components are equally weighted.

Scanned with CamScanner

Problem (2):

a) $P_A = \frac{16}{112} = 0.14$

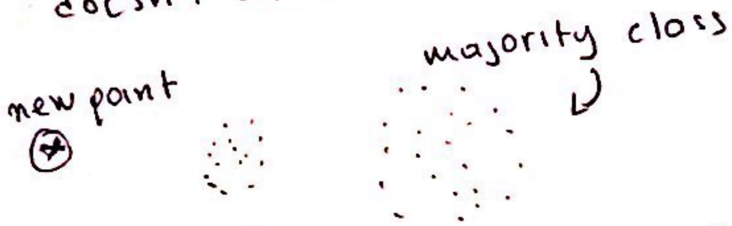
$P_B = \frac{32}{112} = 0.28$

$P_C = \frac{64}{112} = 0.57$

The new point will most probable belong to class C since $P_C > P_B > P_A$.

b) In the weighted version of k-nn, it doesn't matter which class has the highest probability because we classify based on distance. It might happen that the Xnew is further from the majority class so the term $\frac{1}{d(x, x_i)} \rightarrow 0$ and the fact that majority class has higher probability doesn't affect the decision of the probability classifier.

Example :



(the distance to majority class will be penalized hard).

Problem (3):

The condition which satisfies the plot and achieving 100% accuracy is the following: $y < x$. If $y < x$ then we get the points under the line $y = x$ and otherwise we get the elements over it. <u>But</u>, this equation is a function which doesn't split the space into cuboids. Therefore, a decision tree does not exist. In addition, decision trees are meant for non-linear separable data points.

Problem 4:

$P(y=w) = \frac{4}{10}$

$P(y=L) = \frac{6}{10}$

$H(Y) = -\frac{4}{10}\log_2\frac{4}{10} - \frac{6}{10}\log_2\frac{6}{10} = 0.971$

splitting at X1:

$p(T) = p(I) = 0.5$

$E(2,3) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$

$E(y,X1) = \frac{5}{10}E(2,3) + \frac{5}{10}E(2,3) = 0.971$

splitting at X2:

$p(M) = 0.4$

$p(P) = 0.6$

$E(2,2) = 1$

$E(2,4) = 0.822$

$E(y,X2) = 0.8932$

splitting at X3:

$p(S) = 0.5$

$p(C) = 0.5$

$E(3,2) = 0.971$

$E(1,4) = 0.722$

$E(y,X3) = 0.8455$

In order to decide at which node we will split we calculate the information gain and we choose the highest one.

$Gain(y,X1) = Entropy(y) - Entropy(y,X1) = 0.005$

$Gain(y,X2) = Entropy(y) - Entropy(y,X2) = 0.0728$

$Gain(y,X3) = Entropy(y) - Entropy(y,X3) = \underline{\underline{0.1205}}$

Therefore, we split at X3.