Machine Learning Worksheet 03

Decision Trees and Nearest Neighbours

1 iPython notebook

Load the notebook `kNN_implementationHW.ipynb` from piazza.

Problem 1: Fill in the missing code and run the notebook. Add the printed notebook to the printout of your homework.

2 Learning by doing

You are free to do these completely by hand or use a computer to help speed things up (python, MATLAB, R, Excel, . . .). You should, however, show the basic steps of your work and implement your own "helpers" instead of blindly using code. Using a machine learning toolbox and copying the result will not help you understand.

The table below gives you a feature matrix $\boldsymbol{X}$ together with the output z_i for every row i of the feature matrix. This data is also available in Piazza as *homework03.csv*.

i	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	z
A	5.5	0.5	4.5	2
B	7.4	1.1	3.6	0
C	5.9	0.2	3.4	2
D	9.9	0.1	0.8	0
E	6.9	-0.1	0.6	2
F	6.8	-0.3	5.1	2
G	4.1	0.3	5.1	1
H	1.3	-0.2	1.8	1
I	4.5	0.4	2.0	0
J	0.5	0.0	2.3	1
K	5.9	-0.1	4.4	0
L	9.3	-0.2	3.2	0
M	1.0	0.1	2.8	1
N	0.4	0.1	4.3	1
O	2.7	-0.5	4.2	1

Problem 2: Build a decision tree T for your data. Consider all possible feature tests and use the Gini index to build your tree. Build the tree only to a depth of two! Provide at least the value of the final Gini index at each node and the distribution of classes at each leaf.

Submit to `homework@class.brml.org` *with subject line* `homework sheet 03` *by 2015/11/01, 23:59 CET*

Problem 3: Use the tree from the previous problem to classify the vectors $\boldsymbol{x}_a = (4.1, -0.1, 2.2)$ and $\boldsymbol{x}_b = (6.1, 0.4, 1.3)$. Provide both your classification $\hat{z}_a$ and $\hat{z}_b$ and their respective probabilities $p(c = \hat{z}_a \mid \boldsymbol{x}_a, T)$ and $p(c = \hat{z}_b \mid \boldsymbol{x}_b, T)$

Problem 4: Classify the two vectors given in Problem 2 with the K-nearest neighbors algorithm. Use $K = 3$ and Euclidean distance.

Problem 5: Now, consider z_i to be real-valued labels rather than classes. Perform 3-NN regression to label the vectors from Problem 2.

Problem 6: Look at the data. Which problem do you see w.r.t. building a Euclidean distance-based KNN model on $\boldsymbol{X}$? How can you compensate for this problem? Does this problem also arise when training a decision tree?