Machine Learning Exercise Sheet 2

k-Nearest Neighbors and Decision Trees

1 kNN Classification

Problem 1: You are given the following dataset, with points of two different classes:

Name	x_1	x_2	class
A	1.0	1.0	1
B	2.0	0.5	1
C	1.0	2.5	1
D	3.0	3.5	2
E	5.5	3.5	2
F	5.5	2.5	2

We perform 1-NN classification with leave-one-out cross validation on the data in the plot.

a) Compute the distance between each point and its nearest neighbor using L_1-norm as distance measure.

b) Compute the distance between each point and its nearest neighbor using L_2-norm as distance measure.

c) What can you say about classification if you compare the two distance measures?

Problem 2: Consider a dataset with 3 classes $\mathcal{C} = \{A, B, C\}$, with the following class distribution $N_A = 16, N_B = 32, N_C = 64$. We use unweighted k-NN classifier, and set k to be equal to the number of data points, i.e. $k = N_A + N_B + N_C =: N$.

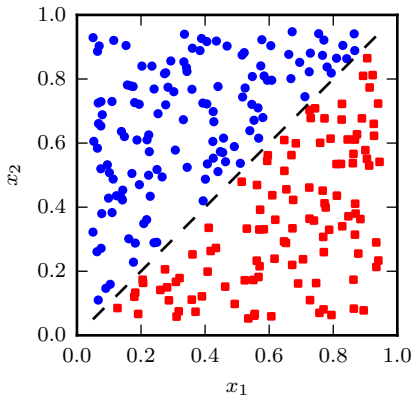
a) What can we say about the prediction for a new point x_{new}?

b) How about if we use the weighted (by distance) version of k-Nearest Neighbors?

2 Decision Trees

Problem 3: The plot below shows data of two classes that can easily be separated by a single (diagonal) line. Does there exist a decision tree of depth 1 that classifies this dataset with 100% accuracy? Justify your answer.

Upload a single PDF file with your solution to Moodle by 27.10.2019, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted.
If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.



Problem 4: You are developing a model to classify games at which machine learning will beat the world champion within five years. The following table contains the data you have collected.

No.	x_1 (Team or Individual)	x_2 (Mental or Physical)	x_3 (Skill or Chance)	y (Win or Lose)
1	T	M	S	W
2	I	M	S	W
3	T	P	S	W
4	I	P	C	W
5	T	P	C	L
6	I	M	C	L
7	T	M	S	L
8	I	P	S	L
9	T	P	C	L
10	I	P	C	L

a) Calculate the entropy $i_H(y)$ of the class labels y.

b) Build the optimal decision tree of depth 1 using entropy as the impurity measure.

3 Programming Task

Problem 5: Load the notebook `exercise_02_notebook.ipynb` from Piazza. Fill in the missing code and run the notebook. Convert the evaluated notebook to HTML using nbconvert and add it to your submission.

Note: We suggest that you use Anaconda for installing Python and Jupyter, as well as for managing packages. We recommend that you use Python 3.

For more information on Jupyter notebooks, consult the Jupyter documentation. Instructions for converting the Jupyter notebooks to PDF are provided within the notebook.