Machine Learning Exercise Sheet 2

k-Nearest Neighbors and Decision Trees

Exercise sheets consist of two parts: Homework and in-class exercises. The homework is for you to solve at home and upload to Moodle for a possible grade bonus. The in-class exercises will be solved and discussed during the tutorial along with some difficult and/or important homework exercises. You do not have to upload any solutions of the in-class exercises.

In-class Exercises

There are no additional in-class exercises this week.

Homework

kNN Classification

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

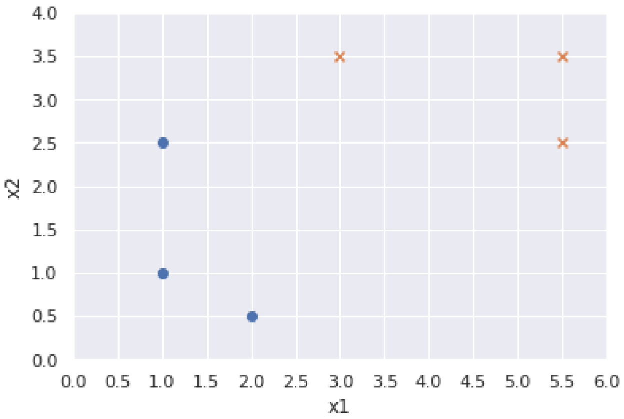
Name	x_1	x_2	class
A	1.0	1.0	1
B	2.0	0.5	1
C	1.0	2.5	1
D	3.0	3.5	2
E	5.5	3.5	2
F	5.5	2.5	2

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.

If you draw the points, you can identify the nearest neighbor without computing all distances:

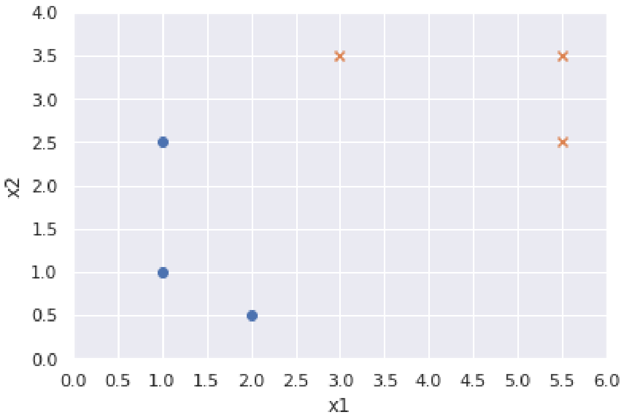
Upload a single PDF file with your homework solution to Moodle by 03.11.2021, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.



$$d_1(X,Y) = \sum_i |X_i - Y_i|$$

L_1-norm	A	B	C	D	E	F	nn	class
A	0.00	**1.50**	**1.50**	4.50	7.00	6.00	B/C	1 ✓
B	**1.50**	0.00	3.00	4.00	6.50	5.50	A	1 ✓
C	**1.50**	3.00	0.00	3.00	5.50	4.50	A	1 ✓
D	4.50	4.00	3.00	0.00	**2.50**	3.50	E	2 ✓
E	7.00	6.50	5.50	2.50	0.00	**1.00**	F	2 ✓
F	6.00	5.50	4.50	3.50	**1.00**	0.00	E	2 ✓

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



$$d_2(X,Y) = \left(\sum_i (X_i - Y_i)^2\right)^{\frac{1}{2}}$$

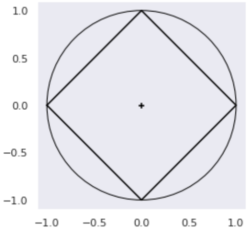
Upload a single PDF file with your homework solution to Moodle by 03.11.2021, 23:59 CET. We recommend to typeset your solution (using LaTeX or Word), but handwritten solutions are also accepted. If your handwritten solution is illegible, it won't be graded and you waive your right to dispute that.

L_2-norm	A	B	C	D	E	F	nn	class
A	0.00	**1.12**	1.50	3.20	5.15	4.74	B	1 ✓
B	**1.12**	0.00	2.24	3.16	4.61	4.03	A	1 ✓
C	**1.50**	2.24	0.00	2.24	4.61	4.50	A	1 ✓
D	3.20	3.16	**2.24**	0.00	2.50	2.69	C	2 ×
E	5.15	4.61	4.61	2.50	0.00	**1.00**	F	2 ✓
F	4.74	4.03	4.50	2.69	**1.00**	0.00	E	2 ✓

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

Different distance measures can result in a different nearest neighbor and change the class a point is assigned to. Point D (orange cross) is closest to E (orange cross) regarding L_1-norm but closest to B (blue dot) regarding L_2-norm.

Regarding the distance measures it always holds that $L_2 \leq L_1$ (see unit circles of the two norms).



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}?

It will be classified as class C. When k is equal to the number of data points, the neighborhood of a new point contains all points in the training set regardless of their distance. The majority class in the neighborhood is thus equal to the majority class in the dataset.

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

For the distance weighted variant we don't have enough information to answer the question, since the *weighted* distribution depends on the distances.

Problem 3: Assume you use a KNN-classifier on the following training data, that contains at least 100 samples of each class.

Upload a single PDF file with your homework solution to Moodle by 03.11.2021, 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.

Acceleration	max. velocity [km/h]	PS	cylinder capacity [cm^3]	weight [kg]	class
3.6	250	600	3996	2150	car
12.5	178	150	1968	2001	van
3.5	200	113	937	227	motorcycle
...	...	...	...	...	...

You observe that the obtained model performs bad on the test set. What might be the problem? Name at least two possible problems and explain how you would solve them. Would a decision tree have the same problems? Justify your answer.

Problem: different range of the features

⇒ features are equally important but have different impact on the model

Standardize the data: $\boldsymbol{x}'_i = \frac{\boldsymbol{x}_i - \mu_i}{\delta_i}$

Problem: bad hyperparameter k

⇒ optimize hyperparameter k (gird-search)

Problem: shift between training and test set

⇒ Choose training and test set such that they are from the same distribution

Problem: different range of the features ⇒ No.

Decision trees can handle features of different scale, because the splits/decision boundaries are computed based misclassification rate, entropy or Gini index. All these measures depend on the labels of the data-point and are computed based on distinguishing if the currently considered feature x is smaller or larger than a threshold. Only feature x influences these measure (for the considered split/test), the other ones don't. Thus, the scale of the features is not important.

Problem: bad hyperparameters ⇒ No, there is no hyperparameter k

Problem: shift between training and test set ⇒ Yes.

Problem 4: You want to perform 1-kNN-classification based on

i) L_1-norm

ii) L_2-norm

Prove or disprove: The L_2-distance $d_2(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{d}(x_i - y_i)^2\right)^{\frac{1}{2}}$ between two points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ is always smaller or equal than the L_1-distance $d_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{d} |x_i - y_i|$.

Upload a single PDF file with your homework solution to Moodle by 03.11.2021, 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.

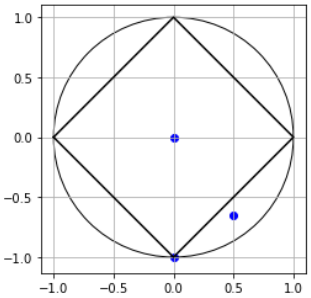
True, see proof:

$$\begin{aligned} d_2(\boldsymbol{x},\boldsymbol{y})^2 &= \sum_i (x_i - y_i)^2 \\ &= \sum_i |x_i - y_i||x_i - y_i| \\ &\leq \sum_i |x_i - y_i||x_i - y_i| + \sum_i \sum_{j,j\neq i} |x_i - y_i||x_j - y_j| \\ &= (\sum_i |x_i - y_i|)^2 \\ &= d_1(\boldsymbol{x},\boldsymbol{y})^2 \end{aligned}$$

$\Rightarrow d_2(\boldsymbol{x},\boldsymbol{y}) \leq d_1(\boldsymbol{x},\boldsymbol{y})$

Problem 5: Prove or disprove: Consider two arbitrary points $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2$. If $\boldsymbol{x}$ is the nearest neighbor of $\boldsymbol{y}$ regarding the L_2-norm then $\boldsymbol{x}$ is the nearest neighbor of $\boldsymbol{y}$ regarding the L_1-norm.

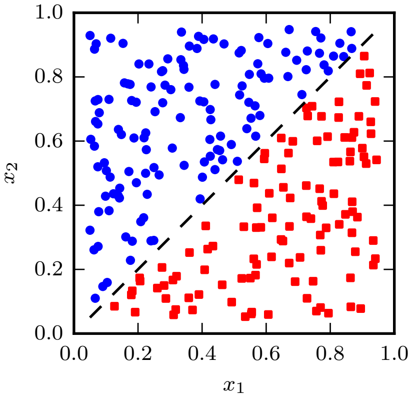
Wrong, see counterexample: Consider the three points $\boldsymbol{y} = (0,0), \boldsymbol{x}_1 = (0,-1), \boldsymbol{x}_2 = (0.5,-0.65)$. Regarding L_1-norm the closest neighbor of $\boldsymbol{y}$ is $\boldsymbol{x}_1$, regarding L_2-norm the closest neighbor of $\boldsymbol{y}$ is $\boldsymbol{x}_2$:



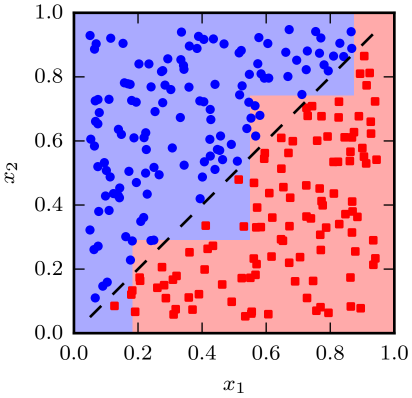
$$\begin{aligned} &d_1(\boldsymbol{y},\boldsymbol{x}_1) = 1 \\ &d_1(\boldsymbol{y},\boldsymbol{x}_2) = 1.15 && \rightarrow \text{neareast neighbor : } \boldsymbol{x}_1 \\ &d_2(\boldsymbol{y},\boldsymbol{x}_1) = 1 \\ &d_2(\boldsymbol{y},\boldsymbol{x}_2) = 0.6725 && \rightarrow \text{neareast neighbor : } \boldsymbol{x}_2 \end{aligned}$$

Decision Trees

Problem 6: 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.



No, there does not exist a decision tree of depth 1 that classifies this dataset with 100% accuracy. The feature test in a node can only use a single feature to split the training data. This leads to axis-parallel decision boundaries. Below you see the decision boundaries for a tree of depth 3. It classifies the dataset with 92.8% accuracy.



Problem 7: 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.

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

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

$$p(y=W) = \frac{4}{10}$$
$$p(y=L) = \frac{6}{10}$$

$$\begin{aligned} i_H(y) &= -p(y=W)\log p(y=W) - p(y=L)\log p(y=L) \\ &= -\frac{4}{10}\log(\frac{4}{10}) - \frac{6}{10}\log(\frac{6}{10}) \\ &\approx 0.97 \end{aligned}$$

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

Split 1, test x_1:

$$\begin{aligned} p(x_1=T) &= \frac{1}{2} \qquad p(x_1=I) = \frac{1}{2} \\ p(y=W|x_1=T) &= \frac{2}{5} \qquad p(y=L|x_1=T) = \frac{3}{5} \\ p(y=W|x_1=I) &= \frac{2}{5} \qquad p(y=L|x_1=I) = \frac{3}{5} \\ i_H(x_1=T) &= -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} \approx 0.97 \\ i_H(x_1=I) &= -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} \approx 0.97 \\ \Delta(x_1) &= i_H(y) - p(x_1=T)i_H(x_1=T) - p(x_1=I)i_H(x_1=I) \\ &= 0.97 - \frac{1}{2}\cdot 0.97 - \frac{1}{2}\cdot 0.97 \\ &= 0 \end{aligned}$$

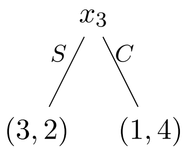
Split 1, test x_2:

$$\begin{aligned}
p(x_2 = M) &= \frac{4}{10} \qquad p(x_2 = P) = \frac{6}{10} \\
p(y = W|x_2 = M) &= \frac{2}{4} \qquad p(y = L|x_2 = M) = \frac{2}{4} \\
p(y = W|x_2 = P) &= \frac{2}{6} \qquad p(y = L|x_2 = P) = \frac{4}{6} \\
i_H(x_2 = T) &= -\frac{2}{4}\log\frac{2}{4} - \frac{2}{4}\log\frac{2}{4} = 1.0 \\
i_H(x_2 = I) &= -\frac{2}{6}\log\frac{2}{6} - \frac{4}{6}\log\frac{4}{6} \approx 0.92 \\
\Delta(x_2) &= i_H(y) - p(x_2 = M)i_H(x_2 = M) - p(x_2 = P)i_H(x_2 = P) \\
&= 0.97 - \frac{4}{10}\cdot 1.0 - \frac{6}{10}\cdot 0.92 \\
&= 0.018
\end{aligned}$$

Split 1, test x_3:

$$\begin{aligned}
p(x_3 = S) &= \frac{5}{10} \qquad p(x_3 = C) = \frac{5}{10} \\
p(y = W|x_3 = S) &= \frac{3}{5} \qquad p(y = L|x_3 = S) = \frac{2}{5} \\
p(y = W|x_3 = C) &= \frac{1}{5} \qquad p(y = L|x_3 = C) = \frac{4}{5} \\
i_H(x_3 = S) &= -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} \approx 0.97 \\
i_H(x_3 = C) &= -\frac{1}{5}\log\frac{1}{5} - \frac{4}{5}\log\frac{4}{5} \approx 0.72 \\
\Delta(x_3) &= i_H(y) - p(x_3 = S)i_H(x_3 = S) - p(x_3 = C)i_H(x_3 = C) \\
&= 0.97 - \frac{5}{10}\cdot 0.97 - \frac{5}{10}\cdot 0.72 \\
&= 0.125
\end{aligned}$$

Split 1: We would split on x_3 since it yields the highest information gain.



Problem 8: Assume you have a dataset with two-dimensional points from two different classes C_1 and C_2. The points from class C_1 are given by $A = \{(i, i^2) \mid i \in \{1...100\}\} \subseteq \mathbb{R}^2$, while the points from class C_2 are $B = \{(i, \frac{125}{i}) \mid i \in \{1...100\}\} \subseteq \mathbb{R}^2$.

Upload a single PDF file with your homework solution to Moodle by 03.11.2021, 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.

Construct a decision tree of minimal depth that assigns as many data points as possible to the correct class. Provide for each split the feature and corresponding thresholds. How many and which datapoints are missclassified?

- Split root node based on feature $1 \leq 5$
- Split both child nodes based on feature $2 \leq 25$

One point $(5, 25)$ is in both classes and missclassified.

Programming Task

Problem 9: Load the notebook `exercise_02_notebook.ipynb` from Piazza. Fill in the missing code and run the notebook. Export (download) the evaluated notebook as PDF 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.

See the solution notebook `exercise_solution_02_notebook.html`.