ILAS Machine Learning WS 2017/18
Exercise sheet 6
Group 24: Gerd Mund, Christopher Schmidt, Maren Pielka

Exercise 1

(i)
To show that $D_{Hamming}$ is a metric we have to show that $D_{Hamming}$ satisfies the usual conditions for a metric:

(a) $D_{Hamming}(x, y) \geq 0$ and $D_{Hamming}(x, y) = 0$ iff x = y
(b) $D_{Hamming}(x, y) = D_{Hamming}(y, x)$
(c) $D_{Hamming}(x, z) \leq D_{Hamming}(x, y) + D_{Hamming}(y, z)$ for any x, y, z $\in F^{(m)}$

Definition of $D_{Hamming}$:
The Hamming distance $D_{Hamming}(x, y)$ between two vectors x, y $\in F^{(m)}$ is the number of coefficients in which they differ.

$$D_{Hamming}(x, y) = |x - y| = \sum_{i=1}^{m} |x - y|$$

Proof:
(a) $D_{Hamming}(x, y) = 0$ iff x, y agree in all coordinates and this happens iff x = y.
$D_{Hamming}(x, y) \geq 0$ is given because we sum up the absolute value of x-y.
(b) Because of the absolute value of x-y in the calculation of $D_{Hamming}(x, y)$ x-y = y-x gives us the same value. So $D_{Hamming}(x, y) = D_{Hamming}(y, x)$.
(c) $D_{Hamming}(x, y)$ is equal to the minimal number of coordinate changes necessary to get from x to y. In its turn, $D_{Hamming}(y, z)$ is equal to the minimal number of coordinate changes necessary to get from y to z.
So $D_{Hamming}(x, y) + D_{Hamming}(y, z)$ changes will get us from x to z.
Hence $D_{Hamming}(x, z) \leq D_{Hamming}(x, y) + D_{Hamming}(y, z)$ which is the minimal number of coordinate changes necessary to get from x to z.

Since we can show that all conditions for a metric are fulfilled, $D_{Hamming}$ is a metric.

(ii)
To show that D_{Δ} is a metric on the power set 2^U of U, we have to show thats D_{Δ} satisfies the usual conditions for a metric:

(a) $D_{\Delta}(S_1, S_2) \geq 0$ and $D_{\Delta}(S_1, S_2) = 0$ iff $S_1 = S_2$
(b) $D_{\Delta}(S_1, S_2) = D_{\Delta}(S_2, S_1)$
(c) $D_{\Delta}(S_1, S_3) \leq D_{\Delta}(S_1, S_2) + D_{\Delta}(S_2, S_3)$ for any $S_1, S_2, S_3 \subseteq U$

Definition of D_{Δ} :

$$D_{\Delta}(S_1, S_2) = |S_1 \Delta S_2| = |(S_1 \setminus S_2) \cup (S_2 \setminus S_1)| \quad \text{for any} \quad S_1, S_2 \subseteq U$$

Proof:
(a) For $D_{\Delta}(S_1, S_2) = 0$ we need $(S_1 \setminus S_2) = \emptyset$ and $(S_1 \setminus S_2) = \emptyset$ because only the union of two empty sets gives us an empty set again. $(S_1 \setminus S_2) = \emptyset$ *iff* $S_1 \subseteq S_2$ and $(S_2 \setminus S_1) = \emptyset$ *iff* $S_2 \subseteq S_1$.

Because of $S_1 \subseteq S_2$ and $S_2 \subseteq S_1 => S_1 = S_2$. $D_{Hamming}(x, y) \geq 0$ you can see easily.

(b) $D_\Delta(S_1, S_2) = |S_1 \Delta S_2| = |(S_1 \setminus S_2) \cup (S_2 \setminus S_1) = |(S_2 \setminus S_1) \cup (S_1 \setminus S_2)| = |S_2 \Delta S_1| = D_\Delta(S_2, S_1)$

(c)

$$D_\Delta(S_1, S_3) \leq D_\Delta(S_1, S_2) + D_\Delta(S_2, S_3)$$
$$<=>$$
$$(S_1 \Delta S_3) \subseteq (S_1 \Delta S_2) \cup (S_2 \Delta S_3)$$
$$<=>$$
$$[(S_1 \setminus S_3) \cup (S_3 \setminus S_1)] \subseteq [(S_1 \setminus S_2) \cup (S_2 \setminus S_1)] \cup [(S_2 \setminus S_3) \cup (S_3 \setminus S_2)]$$
$$<=>$$
$$[(S_1 \setminus S_3) \cup (S_3 \setminus S_1)] \subseteq [(S_1 \cup S_2) \setminus (S_1 \cap S_2)] \cup [(S_2 \cup S_3) \setminus (S_2 \cap S_3)]$$
$$<=>$$
$$[(S_1 \cup S_3) \setminus (S_1 \cap S_3)] \subseteq [(S_1 \cup S_2 \cup S_3) \setminus (S_1 \cap S_2 \cap S_3)]$$
$$<=>$$
$$[(S_1 \setminus S_3) \cup (S_3 \setminus S_1)] \subseteq [(S_1 \setminus S_3) \cup (S_3 \setminus S_1)] \cup [(S_2 \setminus S_3) \cup (S_3 \setminus S_2)]$$
$$<=>$$
$$(S_1 \Delta S_3) \subseteq (S_1 \Delta S_3) \cup (S_2 \Delta S_3)$$
$$<=>$$
$$D_\Delta(S_1, S_3) \leq D_\Delta(S_1, S_3) + D_\Delta(S_2, S_3)$$
$$<=>$$
$$0 \leq D_\Delta(S_2, S_3)$$

We show in (a) that $0 \leq D_\Delta(S_2, S_3)$ is true for any $S_2, S_3 \subseteq U$, so if you go back the equation chain $D_\Delta(S_1, S_3) \leq D_\Delta(S_1, S_2) + D_\Delta(S_2, S_3)$ for any $S_1, S_2, S_3 \subseteq U$ is true.

Since we can show that all conditions for a metric are fulfilled, D_Δ is a metric on the power set 2^U of U.

Exercise 2

(a) We have to show that for each $n \in \mathbb{N}$ with $n \geq 3$ we can find a set of P with $|P| = n$ in such a way that all cells of the Voronoi diagram are unbounded.

Let P be a set of vertices of a convex n-polygon A. So $|P| = n$. Now we take a point M inside of A. Now we can construct our Voronoi diagram in such a way that we draw n half-lines with M as the vertex for all. To draw a half-line we need another point which lays on the half-line. For the first half-line we chose the middle point of the line segment P_1 and P_2 . So we go one till we chose our last middle point between P_n and P_1 . So we construct a Voronoi diagram that is consistent with: • each cell contains exactly one point from P

- for all $q \in \mathbb{R}^2$ that lie in the cell containing $p \in P$ it holds that $d(p, q) < d(p', q) for all p' \in P \setminus \{p\}$

And all cells are unbounded because we only use half-lines.
We found this construction without any limitations, so we found a way for all n.

(b) Show: A Voronoi diagram with n ≥ 3 has at most a number of vertices equal to 2n−5 and the number of its edges is at most 3n − 6.

Proof:

We want to use Euler's formula, but we need to handle the half-lines. So we add a "vertex at infinity" for all unbounded edges.
Now we can use Euler's formula:

Let v be the number of Voronoi vertices (including the vertex at infinity). Let e be the number of Voronoi edges. Let n be the number of Voronoi faces.

Now

$$(v+1) - e + n = 2$$

The sum of vertex degrees is 2e. Since each vertex has degree at least 3, we have $2e \geq 3(v+1)$, which gives a) $v \leq \frac{2}{3}e - 1$ or b) $e \leq \frac{3}{2}(v+1)$.

Plugging a) into Euler's formula, we get $2 \leq \frac{2}{3}e - e + n$ <=> $e \leq 3n - 6$

Plugging b) into Euler's formula, we get $2 \leq (v+1) - \frac{3}{2}(v+1) + n$ <=> $v \leq 2n - 5$

Exercise 3

For our implementation of kNN, we used Python 3.5 with scipy and numpy libraries. In the following, we will present our results shortly.

Part (2): k-NN classifier with k= 1, 3, 5

The accuracy results of the kNN classifier on the normalized data were as follows:

k = 1: 0.9072
k = 3: 0.8927
k = 5: 0.896

It is interesting that k=1 yields the best result, although it obtains its classification from checking only one neighbor. This suggests that there are often two samples with the same classification positioned very closely together in the feature space. (For this case, two emails with very similar content and similar classification as spam or not spam.)

Part (3): Decision Tree with depth 5

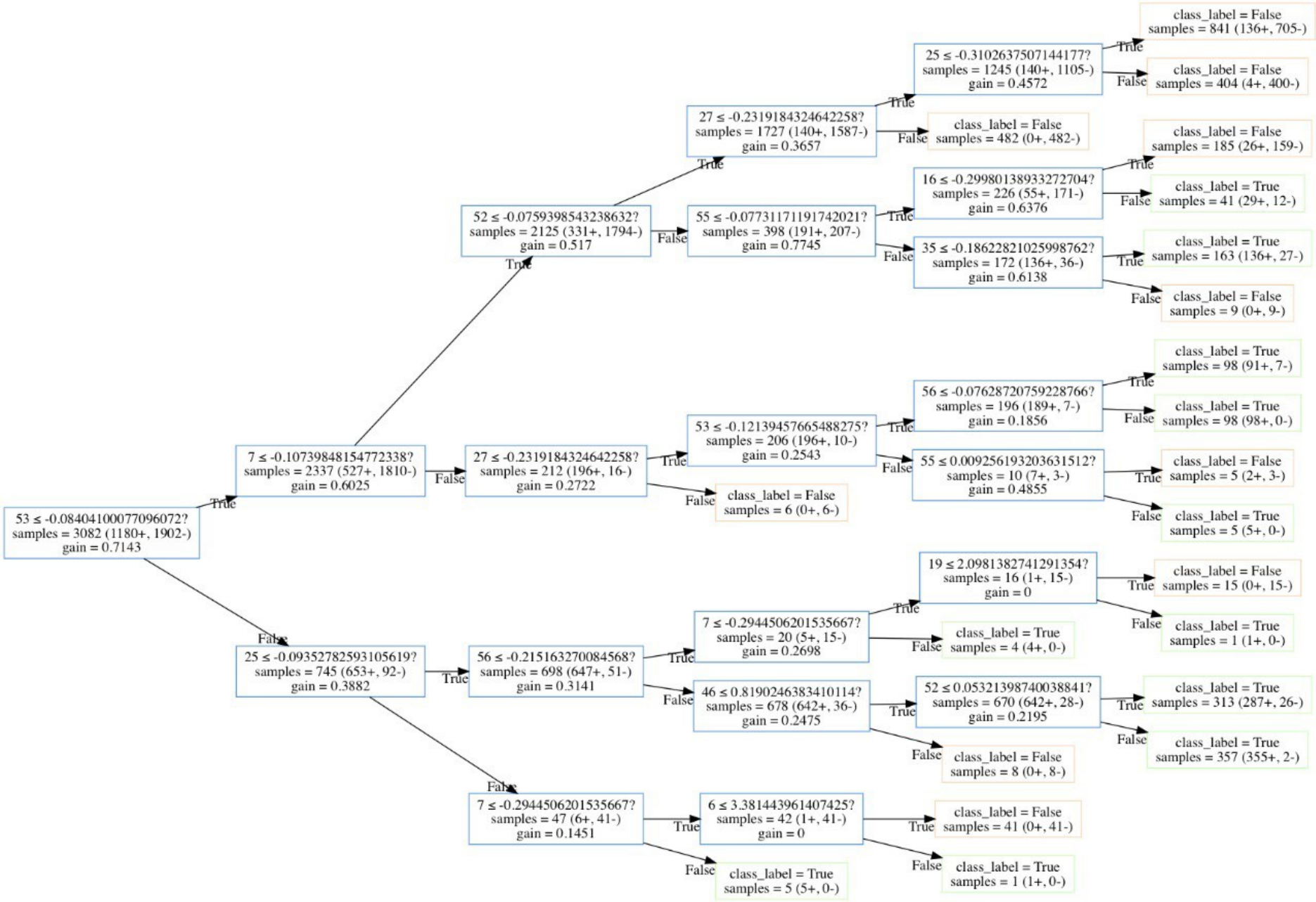
We used our previously implemented TDIDT algorithm to build a Decision Tree for the spam data. The tree was cut off at depth 5. It achieved an accuracy of 0.9078 on the test data, which is slightly better than the k-NN result for k=1. You can find a visualization of the Decision Tree at the bottom of the page. The most important attribute, according to the tree, is number 53. (We numbered the attributes 1-57 for this purpose.)
The Decision Tree is more easily understandable to humans, since it gives a clear structure of ordered choices everyone can comprehend. The k-NN model is a more abstract one, as its multi-feature-distances are not as intuitively interpretable.

Part (4.4): k-NN classifier, using only the attributes from the Decision Tree

For the group-dependent task, we ran the k-NN algorithm on a modified training- and test-set with only the features that appear in the Decison Tree. The accuracies were as follows:

k = 1: 0.8999
k = 3: 0.9045
k = 5: 0.9026

Interestingly, here the accuracy is worst for k=1 and best for k=3, which is rather what we would have expected already in (2).


53 ≤ -0.08404100077096072?
samples = 3082 (1180+, 1902-)
gain = 0.7143
True
False
7 ≤ -0.107398481547723 38?
samples = 2337 (527+, 1810-)
gain = 0.6025
True
False
52 ≤ -0.0759398543238632?
samples = 2125 (331+, 1794-)
gain = 0.517
True
False
27 ≤ -0.2319184324642258?
samples = 1727 (140+, 1587-)
gain = 0.3657
True
False
25 ≤ -0.3102637507144177?
samples = 1245 (140+, 1105-)
gain = 0.4572
True
False
class_label = False
samples = 841 (136+, 705-)
class_label = False
samples = 404 (4+, 400-)
class_label = False
samples = 482 (0+, 482-)
55 ≤ -0.077311711917420 21?
samples = 398 (191+, 207-)
gain = 0.7745
True
False
16 ≤ -0.299801389332727 04?
samples = 226 (55+, 171-)
gain = 0.6376
True
False
class_label = False
samples = 185 (26+, 159-)
class_label = True
samples = 41 (29+, 12-)
35 ≤ -0.186228210259987 62?
samples = 172 (136+, 36-)
gain = 0.6138
True
False
class_label = True
samples = 163 (136+, 27-)
class_label = False
samples = 9 (0+, 9-)
27 ≤ -0.2319184324642258?
samples = 212 (196+, 16-)
gain = 0.2722
True
False
53 ≤ -0.121394576654882 75?
samples = 206 (196+, 10-)
gain = 0.2543
True
False
class_label = False
samples = 6 (0+, 6-)
56 ≤ -0.076287207592287 66?
samples = 196 (189+, 7-)
gain = 0.1856
True
False
55 ≤ 0.0092561932036315 12?
samples = 10 (7+, 3-)
gain = 0.4855
True
False
class_label = True
samples = 98 (91+, 7-)
class_label = True
samples = 98 (98+, 0-)
class_label = False
samples = 5 (2+, 3-)
class_label = True
samples = 5 (5+, 0-)
25 ≤ -0.093527825931056 19?
samples = 745 (653+, 92-)
gain = 0.3882
True
False
56 ≤ -0.215163270084568?
samples = 698 (647+, 51-)
gain = 0.3141
True
False
7 ≤ -0.2944506201535667?
samples = 20 (5+, 15-)
gain = 0.2698
True
False
19 ≤ 2.0981382741291354?
samples = 16 (1+, 15-)
gain = 0
True
False
class_label = False
samples = 15 (0+, 15-)
class_label = True
samples = 1 (1+, 0-)
class_label = True
samples = 4 (4+, 0-)
46 ≤ 0.8190246383410114?
samples = 678 (642+, 36-)
gain = 0.2475
True
False
52 ≤ 0.0532139874003884 1?
samples = 670 (642+, 28-)
gain = 0.2195
True
False
class_label = True
samples = 313 (287+, 26-)
class_label = True
samples = 357 (355+, 2-)
class_label = False
samples = 8 (0+, 8-)
7 ≤ -0.2944506201535667?
samples = 47 (6+, 41-)
gain = 0.1451
True
False
6 ≤ 3.381443961407425?
samples = 42 (1+, 41-)
gain = 0
True
False
class_label = False
samples = 41 (0+, 41-)
class_label = True
samples = 1 (1+, 0-)
class_label = True
samples = 5 (5+, 0-)