Exercise: 5

Meeting on February 2nd / 4th

Aufgabe 1: k-Nearest Neighbor Classifier

Consider the following table of symptoms and corresponding disease classifications by a medical expert:

Case	Fever	Vomit	Diarrhea	Shivering	Classification
c_1	no	no	no	no	healthy (G)
c_2	average	no	no	no	Influenza (I)
c_3	high	no	no	yes	Influenza (I)
c_4	high	yes	yes	nein	Salmonella poisoning (S)
c_5	average	no	yes	no	Salmonella poisoning (S)
c_6	no	yes	yes	no	Bowel Inflammation (D)
c_7	average	yes	yes	no	Bowel Inflammation (D)

Table 1: Case base for disease classification using k-nearest neighbor

For computation using the Hamming-distance we need a similarity measure for new cases. Using their experience, the expert thought of a similarity measure for a given case c from the case base and a query q as well as corresponding weights w_a:

sim_F

q/c	no	average	high
no	1.0	0.7	0.2
average	0.5	1.0	0.8
high	0.0	0.3	1.0

$\text{sim}_E = \text{sim}_D = \text{sim}_Z$

q/c	yes	no
yes	1.0	0.0
no	0.2	1.0

Weights
$w_F = 0.3$
$w_E = 0.2$
$w_D = 0.2$
$w_Z = 0.3$

(a) Compute the similarity between all cases c from the case base with the query q = (high, no, no, no). Which classification do you get for $k = 1$? Which one for $k = 3$?

(b) Compute the similarity between all cases c from the case base with the query q = (high, no, yes, yes). Which classification do you get for $k = 1$? Which one for $k = 3$?

Solution: Since all attributes are given, the similarity between q and c can be computed using the Hamming distance with the formula

$$\text{Sim}(q, c) = \frac{\sum_{a \in \{F,E,D,Z\}} w_a \cdot \text{sim}_a(q.p_a, c.p_a)}{\sum_{a \in \{F,E,D,Z\}} w_a} \tag{1}$$

Since the weights w_a are already normalized, the equation simplifies to

$$\text{Sim}(q, c) = \sum_{a \in \{F,E,D,Z\}} w_a \cdot \text{sim}_a(q.p_a, c.p_a), \tag{2}$$

(a) Similarities for the first case q = (high, no, no, no) is computed using the above formular and tables:

- For c_1 = ((no, no, no, no), G):
$\text{Sim}(q, c_1) = 0.3 \cdot 0.0 + 0.2 \cdot 1.0 + 0.2 \cdot 1.0 + 0.3 \cdot 1.0 = 0.70$
- For c_2 = ((average, no, no, no), I):
$\text{Sim}(q, c_2) = 0.3 \cdot 0.3 + 0.2 \cdot 1.0 + 0.2 \cdot 1.0 + 0.3 \cdot 1.0 = 0.79$
- For c_3 = ((high, no, no, yes), I):
$\text{Sim}(q, c_3) = 0.3 \cdot 1.0 + 0.2 \cdot 1.0 + 0.2 \cdot 1.0 + 0.3 \cdot 0.2 = 0.76$
- For c_4 = ((high, yes, yes, no), S):
$\text{Sim}(q, c_4) = 0.3 \cdot 1.0 + 0.2 \cdot 0.2 + 0.2 \cdot 0.2 + 0.3 \cdot 1.0 = 0.68$
- For c_5 = ((average, no, yes, no), S):
$\text{Sim}(q, c_5) = 0.3 \cdot 0.3 + 0.2 \cdot 1.0 + 0.2 \cdot 0.2 + 0.3 \cdot 1.0 = 0.63$
- For c_6 = ((no, yes, yes, no), D):
$\text{Sim}(q, c_6) = 0.3 \cdot 0.0 + 0.2 \cdot 0.2 + 0.2 \cdot 0.2 + 0.3 \cdot 1.0 = 0.38$
- For c_7 = ((average, yes, yes, no), D):
$\text{Sim}(q, c_7) = 0.3 \cdot 0.3 + 0.2 \cdot 0.2 + 0.2 \cdot 0.2 + 0.3 \cdot 1.0 = 0.47$

The 3 highest similarities are for c_2, c_3 and c_1 in descending order. Both $k = 1$ and $k = 3$ result in "Influenza".

(b) For the second case q = (high, no, yes, yes):

- For c_1 = ((no, no, no, no), G):
$\text{Sim}(q, c_1) = 0.3 \cdot 0.0 + 0.2 \cdot 1.0 + 0.2 \cdot 0.0 + 0.3 \cdot 0.0 = 0.20$
- For c_2 = ((average, no, no, no), I):
$\text{Sim}(q, c_2) = 0.3 \cdot 0.3 + 0.2 \cdot 1.0 + 0.2 \cdot 0.0 + 0.3 \cdot 0.0 = 0.29$
- For c_3 = ((high, no, no, yes), I):
$\text{Sim}(q, c_3) = 0.3 \cdot 1.0 + 0.2 \cdot 1.0 + 0.2 \cdot 0.0 + 0.3 \cdot 1.0 = 0.80$
- For c_4 = ((high, yes, yes, no), S):
$\text{Sim}(q, c_4) = 0.3 \cdot 1.0 + 0.2 \cdot 0.2 + 0.2 \cdot 1.0 + 0.3 \cdot 0.0 = 0.54$
- For c_5 = ((average, no, yes, no), S):
$\text{Sim}(q, c_5) = 0.3 \cdot 0.3 + 0.2 \cdot 1.0 + 0.2 \cdot 1.0 + 0.3 \cdot 0.0 = 0.49$
- For c_6 = ((no, yes, yes, no), D):
$\text{Sim}(q, c_6) = 0.3 \cdot 0.0 + 0.2 \cdot 0.2 + 0.2 \cdot 1.0 + 0.3 \cdot 0.0 = 0.24$
- For c_7 = ((average, yes, yes, no), D):
$\text{Sim}(q, c_7) = 0.3 \cdot 0.3 + 0.2 \cdot 0.2 + 0.2 \cdot 1.0 + 0.3 \cdot 0.0 = 0.33$

The 3 highest similarities in descending order are reached by c_3, c_4 and c_5. For $k = 1$ the classification result is "Influenza", whereas for $k = 3$ it is "Salmonella poisoning".

Aufgabe 2: Expectation Maximization Algorithm

A friend of your's wants to play a game and explains: He has two coins A and$B$, for which you are supposed to determine wether the coin is biased. This means that instead of the usual 50/50 chance the probabilities for "heads" are given by parameters θ_A and θ_B (and the probability for "tails" is given by $1 - \theta_{A/B}$.

Your friend allows you to test the coins 5 times, however he will only give you one coin at a time randomly (50/50 chance) and you don't know, which one it is. During each experiment you toss the coin 10 times and write down the results. This leads to Tab. 2. The used coin is only known for task (a).

# Experiment	Heads	Tais	Coin
1	5	5	B
2	9	1	A
3	8	2	A
4	4	6	B
5	7	3	A

Table 2: Series of 5 experiments with 10 tosses each. The coin is given here as additional information, but is unkown for the EM-algorithm task.

The probability of observing k heads for a coin $i \in A, B$ with parameter θ is given by a binomial distribution:

$$p_i(k, \theta) = \binom{10}{k} \theta_i^k (1 - \theta_i)^{10-k} \tag{3}$$

(a) Use the maximum likelihood method to compute θ_A und θ_B with the information given in the table.

(b) The log-likelihood is often used instead of the simple likelihood. Compute this for the binomial distribution.

(c) You now want to use the expectation maximization algorithm to approximate $\hat{\theta}_A$ and $\hat{\theta}_B$ from the given observations. You use $\hat{\theta}_A = 0.6$ and $\hat{\theta}_B = 0.5$ as start values. Compute the first iteration of the EM-algorithm by hand. *Hint:* Since the probability for both coins is the same ($P(A) = P(B) = 0.5$) you can leave them out for simplicity.

(d) Now program the EM-algorithm until convergence. The convergence criterium is $\Delta < 0.001$ with $\Delta = \max(|\hat{\theta}_A^{t+1} - \hat{\theta}_A^t|, |\hat{\theta}_B^{t+1} - \hat{\theta}_B^t|$.

Solution:

(a) Using the maximum likelihood method with known coin the parameters are simply computed by counting the corresponding cases of "heads" as:

$$\hat{\theta}_A = \frac{24}{24+6} = 0.80 \quad \hat{\theta}_B = \frac{9}{9+11} = 0.45 \tag{4}$$

(b) The log-likelihood of the binomial distribution is computed as:

$$\begin{aligned}\log(p(n,k,\theta)) &= \log(\binom{n}{k}\theta^k(1-\theta)^{n-k}) \\ &= \log(\binom{n}{k}) + \log(\theta^k) + \log((1-\theta)^{n-k}) \\ &= \log(\binom{n}{k}) + k\cdot\log(\theta) + (n-k)\cdot\log((1-\theta))\end{aligned}$$

(c) Since it is unkown which coin is used at which experiment, we compute the expectation for each experiment separately. Using Eq. 3 we compute the probabilities for coin A and B and get the weights w_A and w_B for one experiment. Our initial parameter values are $\hat{\theta}_A^{(0)} = 0.6$ and $\hat{\theta}_B^{(0)} = 0.5$. For the first experiment this results in:

$$\begin{aligned}p_A(0.5, 0.6) &= \binom{10}{5}0.6^5 \cdot 0.4^5 = 0.201 \\ p_B(0.5, 0.5) &= \binom{10}{5}0.5^5 \cdot 0.5^5 = 0.246 \\ w_A &= \frac{0.201}{0.201+0.246} = 0.45 \quad w_B = \frac{0.246}{0.201+0.246} = 0.55\end{aligned}$$

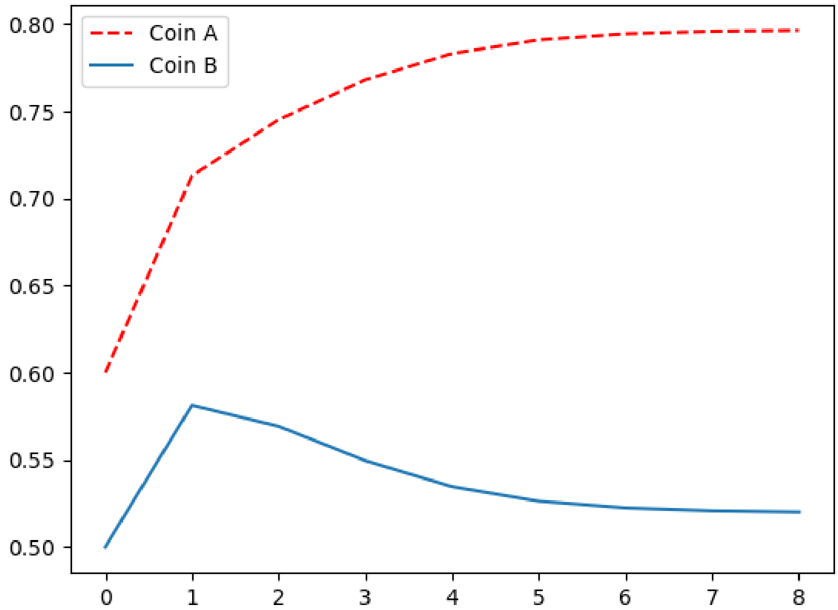
With these values we distribute the tosses for (heads, tails) $= (5,5)$ to both coins. So, for coin A: $w_A \cdot (5,5) = 0.45 \cdot (5,5) = (2.2, 2.2)$ and for coin B: $w_B \cdot (5,5) = 0.55 \cdot (5,5) = (2.8, 2.8)$. Applied to the whole series:

Coin A			Coin B		
w_A	Heads	Tails	w_B	Heads	Tails
0.45	2.2	2.2	0.55	2.8	2.8
0.80	7.2	0.8	0.20	1.8	0.2
0.73	5.9	1.5	0.27	2.1	0.5
0.35	1.4	2.1	0.65	2.6	3.9
0.65	4.5	1.9	0.35	2.5	1.1
Summe	= 21.3	= 8.6		= 11.7	= 8.4

Finally, the maximization step:

$$\begin{aligned}\hat{\theta}_A^{(1)} &= \frac{21.3}{21.3+8.6} \approx 0.71 \\ \hat{\theta}_B^{(1)} &= \frac{11.7}{11.7+8.4} \approx 0.58\end{aligned}$$

(d) The parameter curve could look like this:



Example code in Python:

```python
import numpy as np
import math
import matplotlib.pyplot as plt


def get_binomial_likelihood(n, k, theta):
    """ Return the likelihood of obs, given the probs"""
    # n: total number of trials
    # k: number of outcome 1 (in our case heads)
    # theta: probability for outcome k (heads)
    binomial_coeff = math.factorial(n) / (math.factorial(n - k) * math.factorial(k))
    prod = theta**k * (1 - theta)**(n-k)

    return binomial_coeff * prod


# 1st:  Coin B, {HTTTHHTHTH}, 5H,5T
# 2nd:  Coin A, {HHHHTHHHHH}, 9H,1T
# 3rd:  Coin A, {HTHHHHHTHH}, 8H,2T
# 4th:  Coin B, {HTHTTTHHTT}, 4H,6T
# 5th:  Coin A, {THHHTHHHTH}, 7H,3T
# so, from MLE: pA(heads) = 0.80 and pB(heads)=0.45

# represent the experiments
head_counts = np.array([5,9,8,4,7])
tail_counts = 10 - head_counts
experiments = list(zip(head_counts, tail_counts))

# convergence criterium
delta = 0.001
```

```python

# initialise probability lists
pA_heads = [0.60]
pB_heads = [0.50]

while True:
    expectation_A = []
    expectation_B = []

    for e in experiments:
        # compute the binomial likelihood of experiment for distributions A and B
        lh_A = get_binomial_likelihood(np.sum(e), e[0], pA_heads[-1])
        lh_B = get_binomial_likelihood(np.sum(e), e[0], pB_heads[-1])

        # compute the corresponding weights of A and B proportional to their likelihoods
        weightA = lh_A / (lh_A + lh_B)
        weightB = lh_B / (lh_A + lh_B)

        # compute expectations
        expectation_A.append(np.dot(weightA, e))
        expectation_B.append(np.dot(weightB, e))

    # Maximization
    pA_heads.append(np.sum(expectation_A, axis=0)[0]/np.sum(expectation_A))
    pB_heads.append(np.sum(expectation_B, axis=0)[0]/np.sum(expectation_B))

    improvement = max(abs(pA_heads[-1] - pA_heads[-2]), abs(pB_heads[-1] - pB_heads[-2]))

    if improvement <= delta:
        break

plt.figure()
plt.plot(pA_heads, 'r--')
plt.plot(pB_heads)
plt.show()

print(pA_heads[-1], pB_heads[-1])
```