Homework 1

Miriam Wagner
373045

3. Mai 2018

Question 1

I used for solving this exercise Matlab. My code you can see here:

```
amount_req = [7000, 15000, 23112, 6000, 10000,2500,
    19000,
    5000,13000,35000,5000,7500,6000,6500,25000,10000,7800,8500,17000,3000];

case_duration =
    [0.29309,28.43964,0.000532,0.048206,12.95023,0.021134,19.78213,29.51885,0.515

total_activities =
    [6,74,3,25,26,7,45,46,10,88,25,11,22,6,41,3,52,3,12,15];

c1 = [50000, 1.5, 6];
c2 = [310000, 0.7, 10];
c3 = [30000, 4, 26];

clust1x = [];
clust1y = [];
clust1z = [];
clust2x = [];
clust2y = [];
clust2z = [];
clust3x = [];
clust3y = [];
clust3z = [];
change = 100000;
k = 0;
while change > 10^-15
    for i = 1:length(amount_req)
        distc1(i) = distance([amount_req(i),case_duration
            (i),total_activities(i)],c1);
```

```
22          distc2(i) = distance([amount_req(i),case_duration
                (i),total_activities(i)],c2);
23          distc3(i) = distance([amount_req(i),case_duration
                (i),total_activities(i)],c3);
24
25
26          if (distc1(i) == min([distc1(i),distc2(i),distc3(
                i)]))
27              clust1x = [ clust1x , amount_req(i)];
28              clust1y = [ clust1y , case_duration(i)];
29              clust1z = [ clust1z , total_activities(i)];
30          elseif(distc2(i) == min([distc1(i),distc2(i),
                distc3(i)]))
31              clust2x = [ clust2x , amount_req(i)];
32              clust2y = [ clust2y , case_duration(i)];
33              clust2z = [ clust2z , total_activities(i)];
34          else
35              clust3x = [ clust3x , amount_req(i)];
36              clust3y = [ clust3y , case_duration(i)];
37              clust3z = [ clust3z , total_activities(i)];
38          end
39      end
40      c1new = centroid([clust1x;clust1y;clust1z]);
41      c2new = centroid([clust2x;clust2y;clust2z]);
42      c3new = centroid([clust3x;clust3y;clust3z]);
43      change = distance(c1,c1new) + distance(c2,c2new) +
            distance(c3,c3new);
44      c1 = c1new;
45      c2 = c2new;
46      c3 = c3new;
47      k = k + 1;
48      namen = {'distance_to_cluster_1','
            distance_to_cluster_2','distance_to_cluster_3'};
49      T = table(distc1','distc2','distc3','VariableNames',
            namen);
50      filename = 'Question1.xlsx';
51      writetable(T,filename,'Sheet',k,'Range','A1')
52      namen2 = {'centroid_1','centroid_2','centroid_3'};
53      T = table(c1',c2',c3','VariableNames',namen2);
54      filename = 'Question1centroids.xlsx';
55      writetable(T,filename,'Sheet',k,'Range','A1')
56  end
57  c1
58  c2
59  c3
60
```

```
61 k
62
63      namen3 = {'amount_req','case_duration','
            total_activities'};
64      T = table(clust1x','clust1y','clust1z','VariableNames',
            namen3);
65      filename = 'Question1cluster1.xlsx';
66      writetable(T,filename,'Sheet',1,'Range','A1')
67
68      namen4 = {'amount_req','case_duration','
            total_activities'};
69      T = table(clust2x','clust2y','clust2z','VariableNames',
            namen4);
70      filename = 'Question1cluster2.xlsx';
71      writetable(T,filename,'Sheet',1,'Range','A1')
72      namen5 = {'amount_req','case_duration','
            total_activities'};
73      T = table(clust3x','clust3y','clust3z','VariableNames',
            namen5);
74      filename = 'Question1cluster3.xlsx';
75      writetable(T,filename,'Sheet',1,'Range','A1')
```

The functions used I does not write down them here. 'distance' calculates the euclidian distance and centroid sums up all vectors and divides by the number of vectors.
My clustering algorithm now calculates for every Datavector the distances to every centroid and then checks which centroid is closed. The vector is add to the cluster. When all datavectors are put in a cluster the new cluster centroids are calculated.
My abort criterion is the change between the centroids. If they still change I will apply the algorithms again. Because of the computer accuracy I do not use 0, but 10^{-15}. The algorithm finds after two rounds already good enough clusters.
The distances in the first round are to see in 1
The new centroids are in 2
And the centroids in 4 The next round the distances are in 3
Cluster1 contains 5
Cluster 2 contains as in 6
Cluster 3 in 7

Tabelle 1: Distances first round

distance_to_cluster_1	distance_to_cluster_2	distance_to_cluster_3
0	4	20
68	64	48
3	7	23
19	15	1
20	16	0
1	3	19
39	35	19
40	36	20
4	0	16
82	78	62
19	15	1
5	1	15
16	12	4
0	4	20
35	31	15
3	7	23
46	42	26
3	7	23
6	2	14
9	5	11

Tabelle 2: centroid

centroid_1	centroid_2	centroid_3
7000	13000	15000
0,29309	0,515625	28,43964
6	10	74

Tabelle 3: Distances second round

distance_to_cluster_1	distance_to_cluster_2	distance_to_cluster_3
0	4	68
68	64	0
3	7	71
19	15	49
20	16	48
1	3	67
39	35	29
40	36	28
4	0	64
82	78	14
19	15	49
5	1	63
16	12	52
0	4	68
35	31	33
3	7	71
46	42	22
3	7	71
6	2	62
9	5	59

Tabelle 4: centroids second round

centroid_1	centroid_2	centroid_3
7000	13000	15000
0,29309	0,515625	28,43964
6	10	74

Tabelle 5: Cluster 1

amount_req	case_duration	total_activities
7000	0,29309	6
23112	0,000532	3
2500	0,021134	7
6500	0,002049	6
10000	0,000799	3
8500	0,000486	3
7000	0,29309	6
23112	0,000532	3
2500	0,021134	7
6500	0,002049	6
10000	0,000799	3
8500	0,000486	3

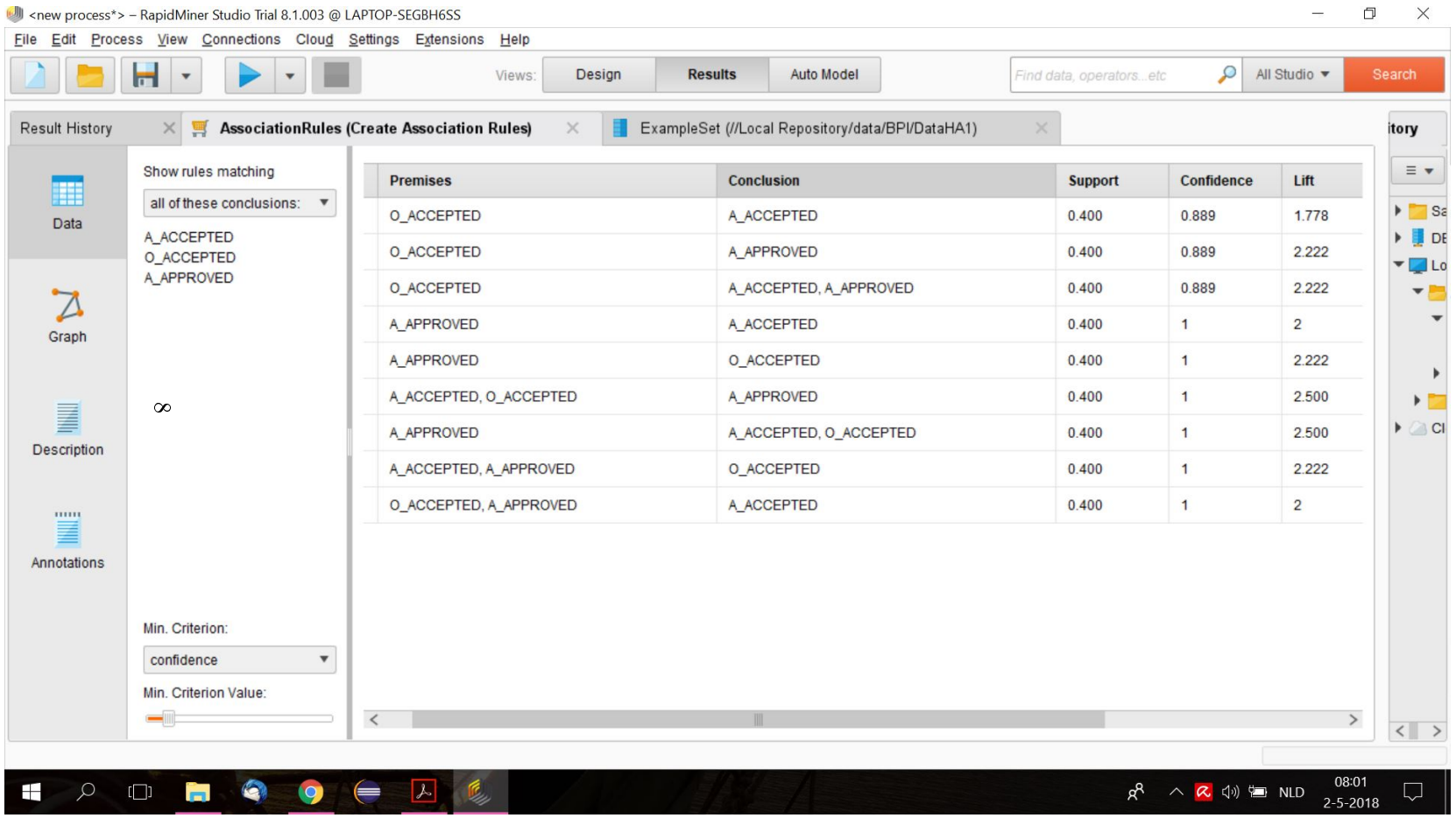
Tabelle 6: Cluster 2

amount_req	case_duration	total_activities
13000	0,515625	10
7500	0,489861	11
17000	0,01375	12
3000	6,955382	15
6000	0,048206	25
10000	12,95023	26
13000	0,515625	10
5000	7,612419	25
7500	0,489861	11
6000	6,503808	22
25000	21,14506	41
17000	0,01375	12
3000	6,955382	15

Tabelle 7: cluster 3

amount_req	case_duration	total_activities
15000	28,43964	74
6000	0,048206	25
10000	12,95023	26
19000	19,78213	45
5000	29,51885	46
35000	19,74352	88
5000	7,612419	25
6000	6,503808	22
25000	21,14506	41
7800	19,1099	52
15000	28,43964	74
19000	19,78213	45
5000	29,51885	46
35000	19,74352	88
7800	19,1099	52

Question 2


<new process*> – RapidMiner Studio Trial 8.1.003 @ LAPTOP-SEGBH6SS
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Auto Model
Find data, operators...etc
All Studio
Search
Result History
AssociationRules (Create Association Rules)
ExampleSet (//Local Repository/data/BPI/DataHA1)
Data
Graph
Description
Annotations
Show rules matching
all of these conclusions:
A_ACCEPTED
O_ACCEPTED
A_APPROVED
Min. Criterion:
confidence
Min. Criterion Value:
Premises | Conclusion | Support | Confidence | Lift
O_ACCEPTED | A_ACCEPTED | 0.400 | 0.889 | 1.778
O_ACCEPTED | A_APPROVED | 0.400 | 0.889 | 2.222
O_ACCEPTED | A_ACCEPTED, A_APPROVED | 0.400 | 0.889 | 2.222
A_APPROVED | A_ACCEPTED | 0.400 | 1 | 2
A_APPROVED | O_ACCEPTED | 0.400 | 1 | 2.222
A_ACCEPTED, O_ACCEPTED | A_APPROVED | 0.400 | 1 | 2.500
A_APPROVED | A_ACCEPTED, O_ACCEPTED | 0.400 | 1 | 2.500
A_ACCEPTED, A_APPROVED | O_ACCEPTED | 0.400 | 1 | 2.222
O_ACCEPTED, A_APPROVED | A_ACCEPTED | 0.400 | 1 | 2
NLD
08:01
2-5-2018

I would pick the rule {A_ACCEPTED, O_ACCEPTED} ⇒ {A_APPROVED} and {A_APPROVED} ⇒ {A_ACCEPTED, O_ACCEPTED}, because they have the highest lift, confidence and support. When you have a closer look you will see that the sets are probably logical equivalent. Always pick the rule with the best lift, confidence and support.

Question 3

The found decision tree is

If you check the Confusionmatrix

accuracy: 100.00%

	true Low	true High	class precision
pred. Low	14	0	100.00%
pred. High	0	6	100.00%
class recall	100.00%	100.00%	

you see, that this decision tree classifys the data perfectly. So you can predict by just knowing the case duration the total activities. If the case duration is higher, than also the total activities are high. This seems to be logical, if you have to do a lot this takes most of the times longer and otherwise around, if you do not need long you mostly did not do a lot of different things in the time.

Question 4

1.

$$L1 = [\langle a, b, e, f\rangle, \langle a, b, e, c, d, b, f\rangle, \langle a, b, c, e, d, b, f\rangle, \langle a, b, c, d, e, b, f\rangle, \langle a, e, b, c, d, b, f\rangle]$$

The $\alpha-$Algorithm gives the following:

$$T_L = \{a, b, c, d, e, f\}$$
$$T_I = \{a\}$$
$$T_O = \{f\}$$

	a	b	c	d	e	f
a	#	$\rightarrow$	#	#	$\rightarrow$	#
b	#	#	$\rightarrow$	#	$\|\|$	$\rightarrow$
c	#	#	#	$\rightarrow$	$\|\|$	#
d	#	$\rightarrow$	#	#	$\|\|$	#
e	#	$\|\|$	$\|\|$	$\|\|$	#	$\rightarrow$
f	#	$\leftarrow$	#	#	#	$\leftarrow$

$X_L = \{(\{a\}, \{b\}), (\{a\}, \{e\}), (\{b\}, \{c\}), (\{b\}, \{f\}), (\{c\}, \{d\}), (\{b\}, \{c, f\}), (\{d\}, \{b\}), (\{e\}, \{f\}), (\{a, d\}, \{b\})\}$
$Y_L = \{(\{a\}, \{e\}), (\{b\}, \{c, f\}), (\{b\}, \{f\}), (\{c\}, \{d\}), (\{e\}, \{f\}), (\{a, d\}, \{b\})\}$