Data Mining

Lizaveta Anikeyeva 1353369,
Pieter Derks 0679959,
Volkan Gumuskaya 0753447,
Fabian Maurer 1350657,
Dennis Rizvic 1020540

September 2019

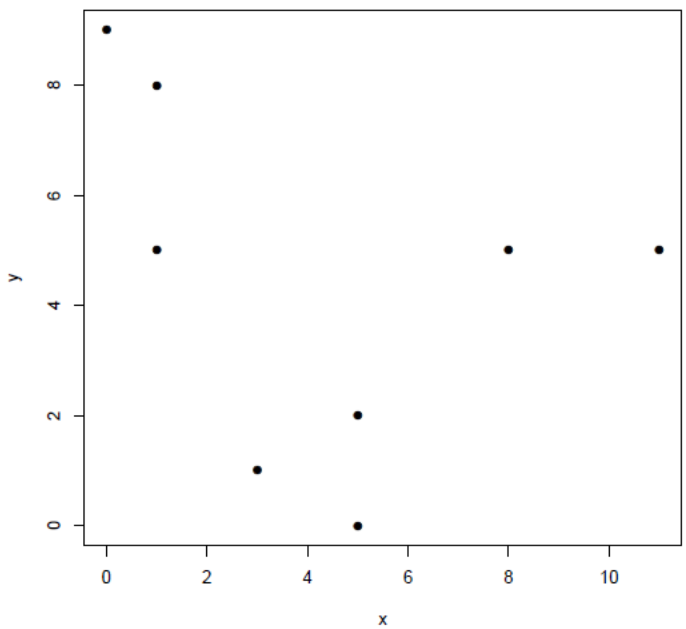
1 Assignment 1A

Clustering with k-means

PART 1

Table 1: Dataset for clustering task

	γ	A	B	C	D	E	F	G	H
α	x	0	1	1	3	5	5	8	11
	y	9	8	5	1	2	0	5	5

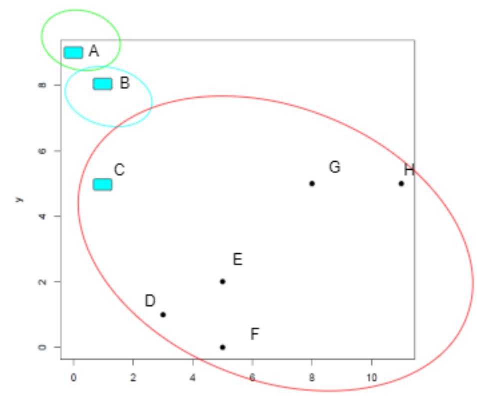


Cluster the dataset with the k-means algorithm, for k = 3, using the records A, B and C as the initial centroids.

Data points need to be assigned to cluster centroids based on the Euclidean distance.

$$CG = \sqrt{(1-8)^2 + (5-5)^2} = \sqrt{49}$$
$$BG = \sqrt{(1-8)^2 + (8-5)^2} = \sqrt{49+9} = \sqrt{58}$$
$$HA = \sqrt{(11-0)^2 + (5-9)^2} = \sqrt{121+16} = \sqrt{137}$$
$$HB = \sqrt{(11-1)^2 + (5-8)^2} = \sqrt{100+9} = \sqrt{109}$$
$$HC = \sqrt{(11-1)^2 + (5-5)^2} = \sqrt{100}$$

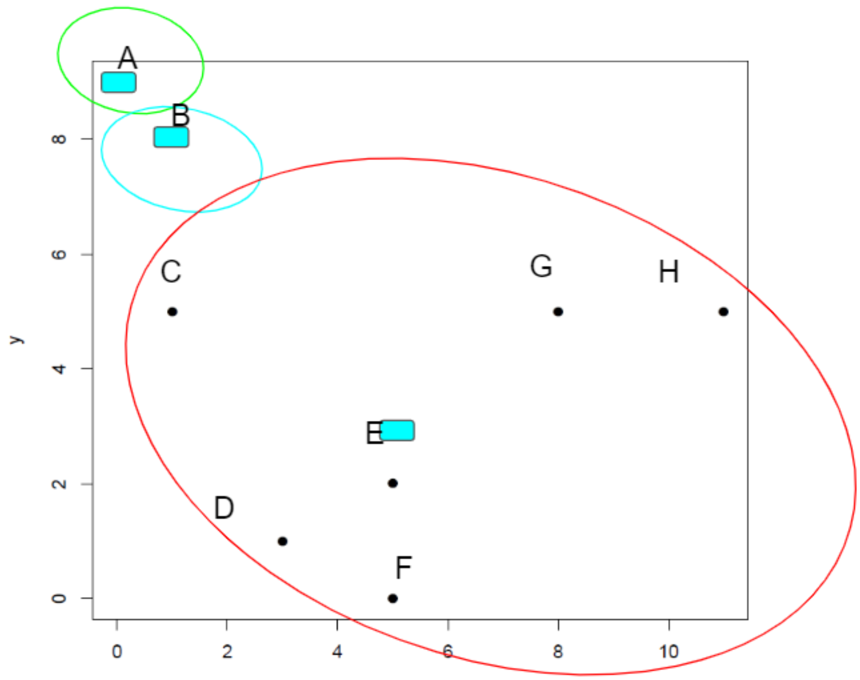
The decision is based on minimizing the distance between a data point and a cluster centroid. It can be concluded from the graph, that data points D, E and F are closer to the centroid C. For H and G it was calculated and they belong to the cluster centroid C, as, for example, between the distance from H to A, B or C, we choose minimum distance, which is HC.



Initial clusters are represented in the graph above.

Find the new centroids for the third cluster.

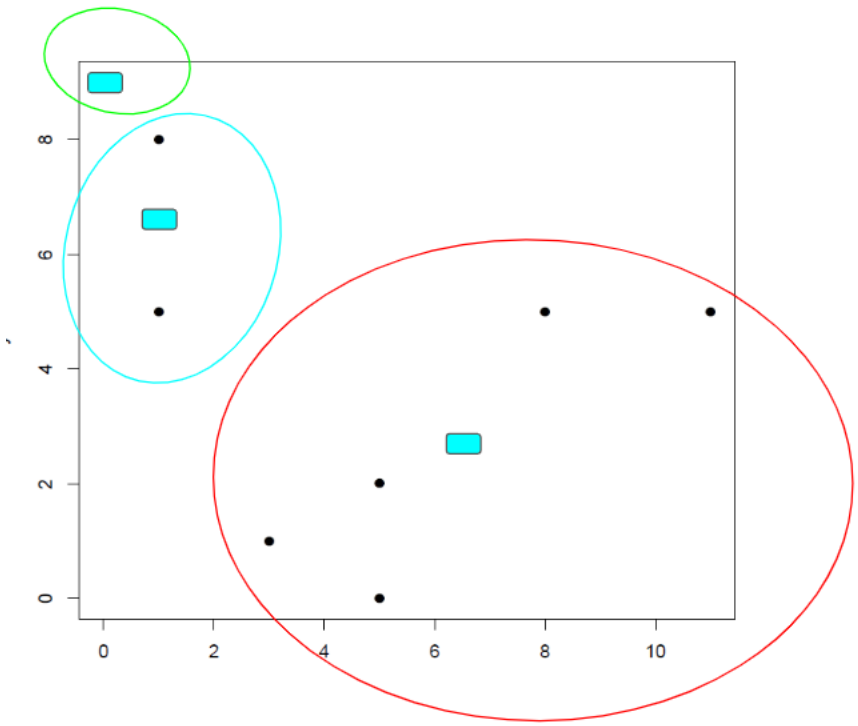
$$CentroidC(x) = \frac{3+5+5+8+11}{5} = 6,4$$
$$CentroidC(y) = \frac{1+0+2+5+5}{5} = 2,6$$



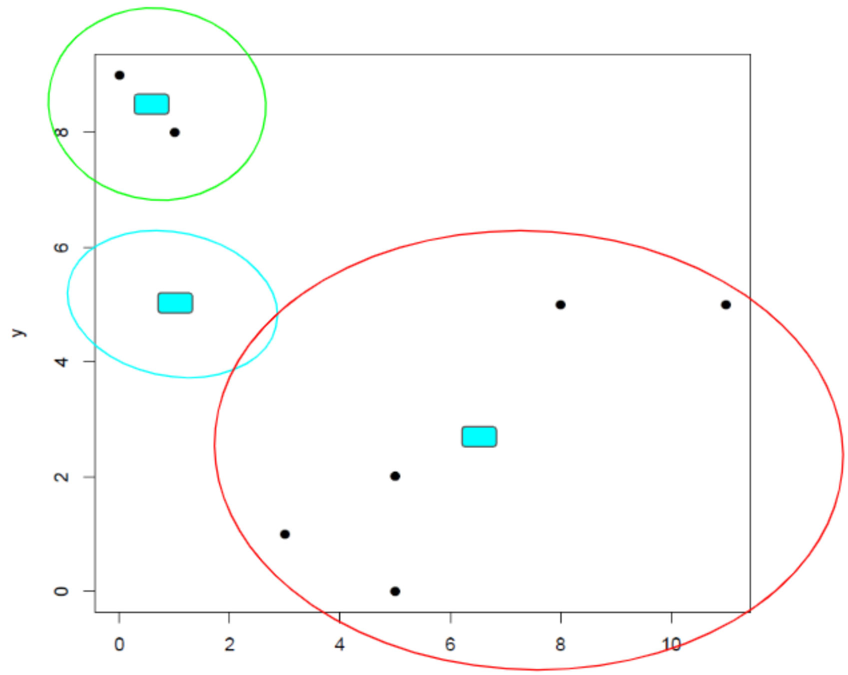
That way, we get another cluster centroid, X1 instead of the data point C. It is important to reassign the points to the new centroids. For that the new distance is recalculated.

$$X1E = \sqrt{1.25}$$
$$X1F = \sqrt{9.25}$$
$$X1C = \sqrt{24.25}$$
$$X2C = \sqrt{9}$$
$$BC = \frac{\sqrt{9}}{2} = 1.5$$

Data point C belongs now to a blue cluster. E and F stay within cluster X3 (red). By calculating the distance between B and C, it can be concluded that X2(1;6.5) is the new centroid of the blue cluster.



New cluster centroids are found. Now it is important to repeat the reassigning step, to make sure the minimum distance criteria is fulfilled.
New red cluster centroid is found. Data point B is closer to the centroid X2 of the blue cluster. Final clustering is represented in the graph below.

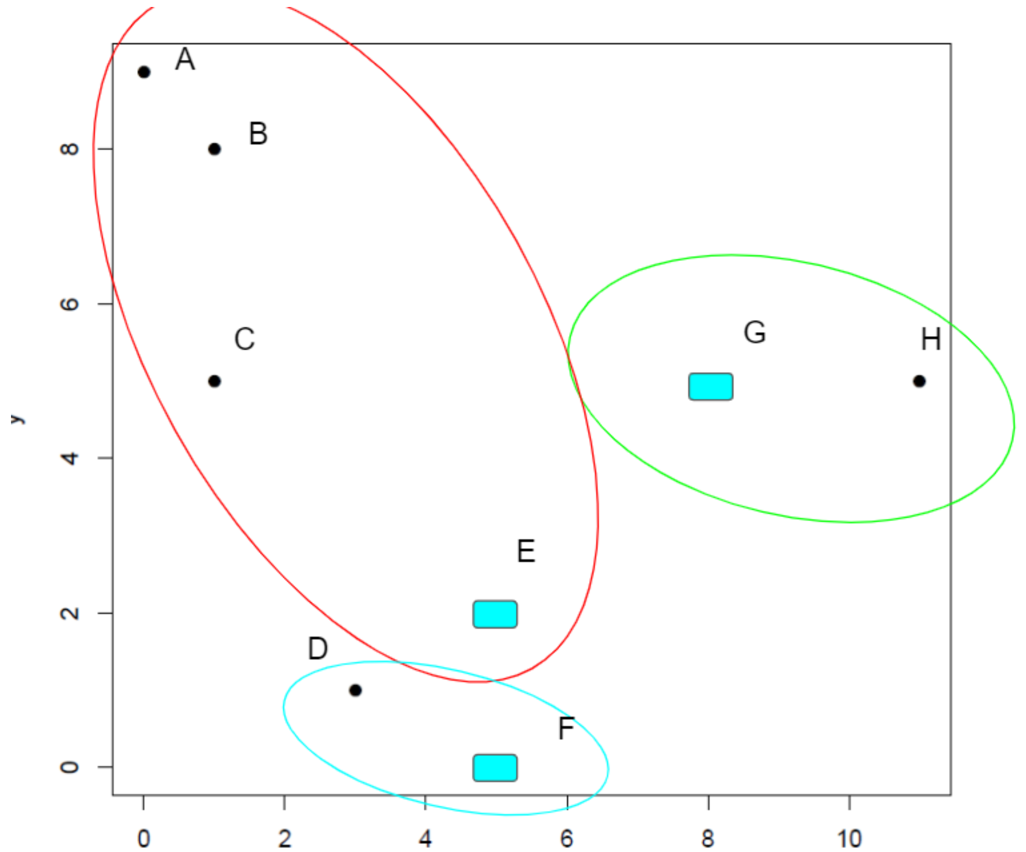


PART 2

Define, which data points belong to which cluster centroids based on the Euclidean distance.

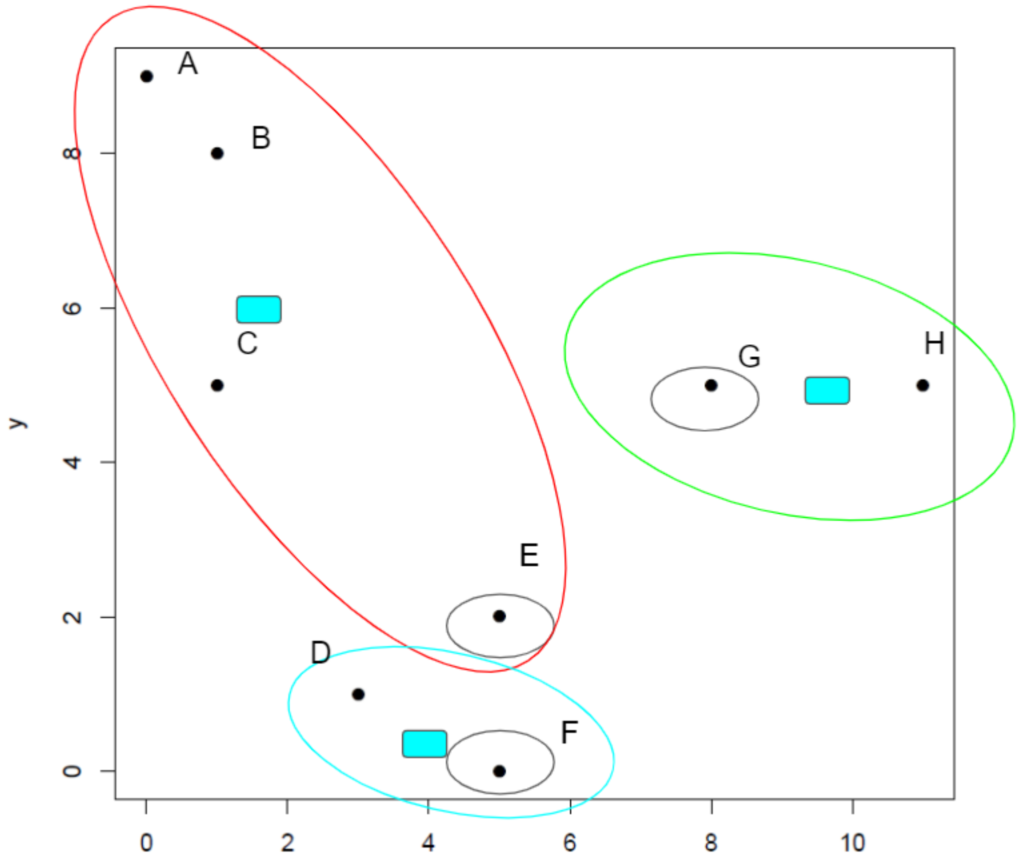
$$EC = \sqrt{25}$$
$$GE = \sqrt{49}$$
$$EB = \sqrt{52}$$
$$GB = \sqrt{58}$$
$$EA = \sqrt{74}$$
$$GA = \sqrt{82}$$
$$DE = \sqrt{5}$$
$$FD = \sqrt{5}$$

The distance from D to E equals the distance from F to D. For the purpose of this tasks, D was assigned to cluster centroid F.



Calculating new centroids:

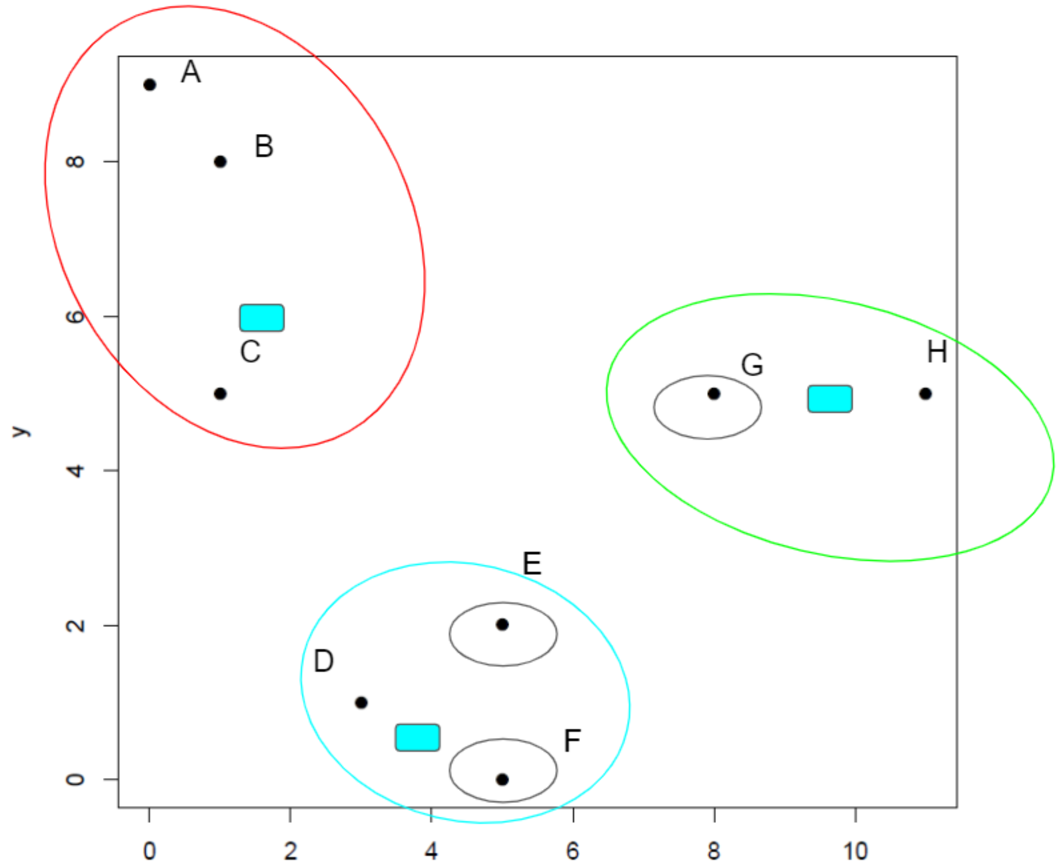
$$Center1(A,B,C,E) = x1(1,75;6)$$
$$Center2(D,F) = x2(4;0,5)$$
$$Center3(G,H) = x3(9,5;5)$$



In the next step it is important to calculate the distance between the new cluster centroids and data points and reassign them if needed. From the last graph it can be seen, that the data points are roughly assigned correctly, except for data point E.

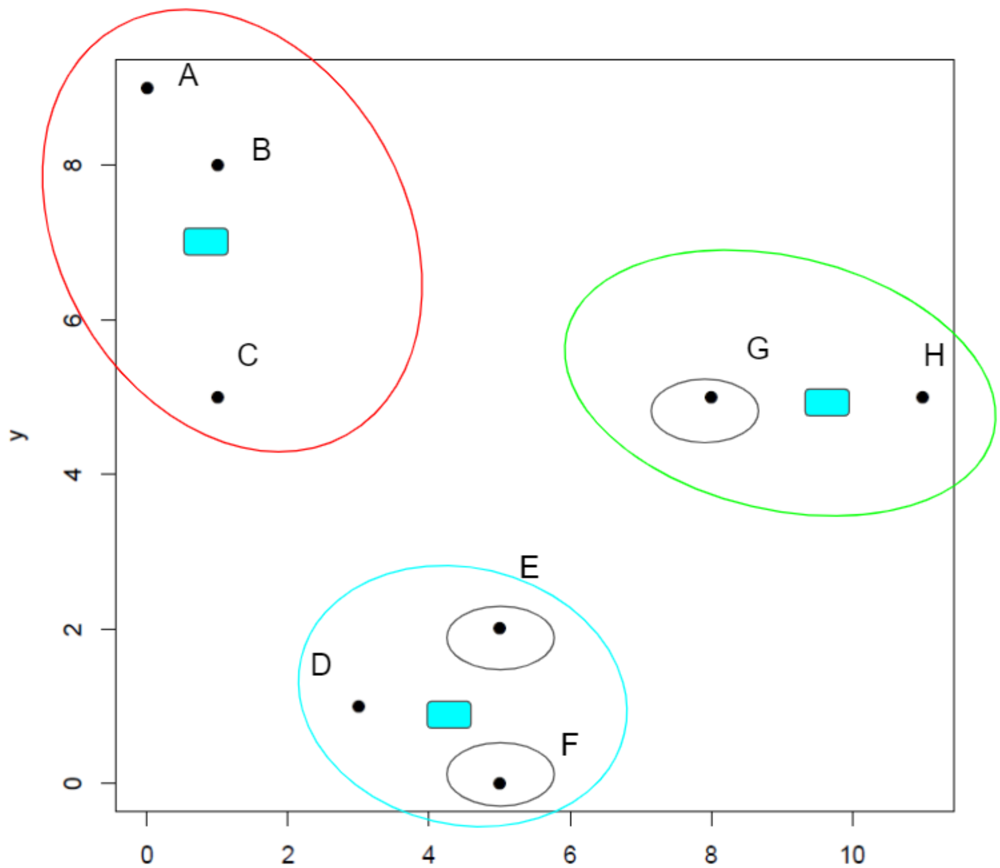
$$X1E = \sqrt{27.22}$$
$$X2E = \sqrt{3,25}$$

The data point E is now assigned to the blue cluster, as the distance between that cluster centroid and E is smaller.



Recalculated cluster centroids:

$$Center1(A,B,C) = x1(0,66;7,33)$$
$$Center2(D,F,E) = x2(4,3;1)$$
$$Center3(G,H) = x3(9,5;5)$$



After calculating the new cluster centroids it can be observed on the graph, that no further reassignment of the data points is needed.

PART 3

For evaluation of clustering efficiency, it is important to calculate within-cluster distance W(C) and the between-cluster dissimilarity B(C).

For PART 1 dataset:

$$
\begin{aligned}
WC1 &= \sqrt{2} = 1,42 \\
WC2 &= 0 \\
WC3 &= DE + DF + DG + DH + EF + EG + EH + FG + FH + GH \\
&= \sqrt{5} + \sqrt{5} + \sqrt{41} + \sqrt{80} + \sqrt{4} + \sqrt{18} + \sqrt{45} + \sqrt{34} + \sqrt{61} + \sqrt{9} \\
&= 39,5 \\
BC1 &= AC + AD + AE + AF + AG + AH + BC + BD + BE + BF + BG \\
&= 96,81 \\
BC2 &= CA + CB + CD + CE + CF + CH + CG = 39,99 \\
BC3 &= DA + DB + DC + EA + EB + EC + FA + FB + FC + GA + GB \\
&+ GC + HA + HB + HC = 122,56
\end{aligned}
$$

For PART 2 dataset:

$$
\begin{aligned}
WC1 &= AB + AC + BC = \sqrt{2} + \sqrt{17} + \sqrt{9} = 8.54 \\
WC2 &= DF + DE + EF = \sqrt{5} + \sqrt{5} + \sqrt{4} = 6.47 \\
WC3 &= GH = 3 \\
BC1 &= AD + AE + AF + AG + AH + BD + BE + BF + BG + BH \\
&+ CD + CE + CF + CG + CH = 122,56 \\
BC2 &= DA + DB + DC + DG + DH + EA + EB + EC + EG + EH \\
&+ FA + FB + FC + FG + FH = 106,69 \\
BC3 &= GA + GB + GC + GD + GE + GF + HA + HB + HC + HD \\
&+ HE + HF = 95,75
\end{aligned}
$$

Ideally, the within-cluster distance should be minimized, and the between-cluster dissimilarity should be maximized. From the calculation it can be concluded that the second clustering outcome is more preferred, since the BC distances are dispersed better. WC of the second dataset shows that the variation of the results is not big, which indicates that the datapoint are close to each other and the distribution was done corretly.

2 Assignment 1B

2.1 Apriori

For brevity, abbreviations are defined for each album as provided in Table 2.

Table 2: Problem data with the abbreviations defined for each album

Abb	Album	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10
RS	Rubber Soul	1	0	0	0	1	1	1	1	1	0
R	Revolver	1	1	1	1	0	1	1	0	0	0
LH	Sgt. Pepper's LHCB	0	0	0	0	1	0	1	1	0	0
M	Magical Mystery Tour	1	1	0	0	1	1	1	1	1	0
B	The Beatles	1	0	0	0	1	1	1	1	0	1
Y	Yellow Submarine	0	1	0	0	0	0	1	0	1	0
A	Abbey Road	1	1	0	1	0	0	0	0	0	0
LIB	Let It Be	1	1	0	0	0	1	1	1	1	0

2.1.1 $\alpha, s_{min} = 2/5$

In this part, Apriori algorithm will be applied to find the frequent itemsets when $s_{min=2/5}$. At each iteration k, first a candidate set C_{k+1} will be determined. Afterwards, C_{k+1} will be pruned to find L_{k+1}, i.e. each itemset in candidate set will be picked one by one, and if the support of the itemset is at least s_{min} the itemset will be deemed to be a frequent itemset and included in L_{k+1}.

- **k=0**

  1. Find C_1
     C_1 is actually trivial. Each one of the albums will be a candidate. Hence;

     $$C_1 = \{\{RS\}\{R\}\{LH\}\{M\}\{B\}\{Y\}\{A\}\{LIB\}\}$$

  2. Find L_1
     Here, the support of each itemset in C_1 will be calculated and compared w.r.t. $s_{min} = 2/5$. If the support is at least 2/5, the itemset will be deemed to be frequent and included in L_1. For example, LH is owned only by three aficionados (z_5, z_7, z_8), support is 3/10, and thus **will not be** in L_1. On the other hand, RS is owned by 6 aficionados, support is 6/10, and thus **will be** in L_1. Calculation of the support of each itemset is excluded for brevity. This yields;

     $$L_1 = \{\{RS\}\{R\}\{M\}\{B\}\{LIB\}\}$$

- **k=1**

  1. Find C_2
     Here, we will check all the pairs one by one in L_1 to see if their union will be a candidate for C_2. Since, each itemset in L_1 is formed by only one album, this task is fairly trivial (See next iteration for a non-trivial example). Hence by taking each pairs from L_1, in C_2 becomes;

     $$C_2 = \{\{RS, R\}\{RS, M\}\{RS, B\}\{RS, LIB\}\{R, M\}\{R, B\}\{R, LIB\} \\ \{M, B\}\{M, LIB\}\{B, LIB\}\}$$

  2. Find L_2
     Here, similar to what we did while finding L_1, the support of each itemset in C_2 will be calculated and compared w.r.t. $s_{min} = 2/5$. The only difference is that we will check the number of aficionados owning both of the albums of a selected itemset. For example, $\{RS, R\}$ is both owned only by three aficionados (z_1, z_6, z_7), support is 3/10, and thus **will not be** in L_2. On the other hand, $\{RS, M\}$ is owned by 6 aficionados, support is 6/10, and thus **will be** in L_2. Calculation of the support of each itemset is excluded from this report for brevity. This yields;

     $$L_2 = \{\{RS, M\}\{RS, B\}\{RS, LIB\}\{R, M\}\{R, LIB\} \\ \{M, B\}\{M, LIB\}\{B, LIB\}\}$$

- **k=2**

  1. Find C_3
     Here, forming C_3 is not trivial. The pairs to be selected from L_2, say l_1 and l_2 should satisfy two conditions: 1) The two itemsets to be unioned should have 1 albums in common (i.e. $k-1$) 2) The candidate itemset, which consists of three albums, should be such that all 2-subsets have to be frequent (See *Candidate Generation* in Lecture notes in 2B-FIM). For example, $\{RS, M\}$ and $\{R, LIB\}$ cannot form a candidate itemset because they do not have a common album (Condition 1 violated). On the other hand, although $\{RS, M\}$ and $\{R, M\}$ satisfy condition 1, the union set $\{RS, M, R\}$ cannot be a candidate itemset because a subset of this itemset, $\{RS, R\}$ is not in L_2 (Condition 2 violated). By applying the rules for all pairwise combinations in L_2, C_3 below is generated. For brevity, the explanation of each pair (28 in total) is not included in this report.

$$C_3 = \{\{RS, M, B\}\{RS, M, LIB\}\{RS, B, LIB\}\{R, M, LIB\}$$
$$\{M, B, LIB\}\}$$

2. Find L_3
   Here, similar to what we did earlier, the support of each itemset in C_3 will be calculated and compared w.r.t. $s_{min} = 2/5$. The only difference is that we will check the number of aficionados owning all three of the albums of a selected itemset. The support of the elements of C_3 is as follows:
   As a result, all elements in C_3 all frequent, which yields;

$$L_3 = \{\{RS, M, B\}\{RS, M, LIB\}\{RS, B, LIB\}\{R, M, LIB\}$$
$$\{M, B, LIB\}\}$$

- **k=3**

1. Find C_4
   We follow the same method in previous step, i.e. check all the pairs in L_3, select the pairs with two albums in common, check if all 3-subset of the union (4 album itemset) of the two selected itemsets are frequent, if yes add the itemset in C_4. When checked all the pairwise combinations of the itemsets in L_3 have two albums in common, and lead to only one single itemset, that is $\{RS, M, B, LIB\}$. This yields;

$$C_4 = \{\{RS, M, B, LIB\}\}$$

2. Find L_4
   The support of $\{RS, M, B, LIB\}$, the only itemset in C_4 is $4/10$ which is indeed at least as large as s_min. Hence;

$$L_4 = \{\{RS, M, B, LIB\}\}$$

   Since there is a single element in L_4, no need to check C_5. We stop.

2.1.2 $\alpha, s_{min} = 3/5$

In this part, Apriori algorithm will be applied to find the frequent itemsets when $s_{min=3/5}$. Everything will be similar to the previous case where $s_{min} = 2/5$ except the s_{min} value is now $3/5$. Therefore, the explanations will not be repeated again for brevity.

- **k=0**

1. Find C_1
   C_1 is actually trivial. Each one of the albums will be a candidate. Hence;

$$C_1 = \{\{RS\}\{R\}\{LH\}\{M\}\{B\}\{Y\}\{A\}\{LIB\}\}$$

2. Find L_1
   Here, the support of each itemset in C_1 will be calculated and compared w.r.t. $s_{min} = 3/5$. Calculation of the support of each itemset is excluded for brevity. This yields;

$$L_1 = \{\{RS\}\{R\}\{M\}\{B\}\{LIB\}\}$$

- **k=1**

1. Find C_2
   Here, we will check all the pairs one by one in L_1 to see if their union will be a candidate for C_2. By taking each pair from L_1, in C_2 becomes;

$$C_2 = \{\{RS, R\}\{RS, M\}\{RS, B\}\{RS, LIB\}\{R, M\}\{R, B\}\{R, LIB\}$$
$$\{M, B\}\{M, LIB\}\{B, LIB\}\}$$

2. Find L_2
   Here, similar to what we did before, the support of each itemset in C_2 will be calculated and compared w.r.t. $s_{min} = 3/5$. The only difference is that we will check the number of aficionados owning both of the albums of a selected itemset (i.e. at least 6 is aficionados required this time out of 10). Calculation of the support of each itemset is excluded from this report for brevity. This yields, as expected a smaller set compared to the case $s_{min} = 2/5$;

$$L_2 = \{\{RS, M\}\{M, LIB\}\}$$

- **k=2**

1. Find C_3
   Here, forming C_3 is not trivial. The pairs to be selected from L_2, say l_1 and l_2 should satisfy two conditions as before: 1) The two itemsets to be unioned should have 1 albums in common (i.e. $k - 1$) 2) The candidate itemset, which consists of three albums, should be

such that all 2-subsets have to be frequent (See *Candidate Generation* in Lecture notes in 2B-FIM). However, since there is only one pair, i.e. $\{RS, M\}$ and $\{M, LIB\}$, this task is straightforward. Since $\{M, LIB\}$, which is a 2-subset of the union itemset $\{RS, M, LIB\}$ is not frequent, there is no itemset as candidate:

$$C_3 = \{\}$$
$$L_3 = \{\}$$

We stop.

As a result, we observe that a smaller s_{min} is less limiting and leads to more frequent itemsets.

β

Largest itemset is {RS,M,B,LIB}.

Assiciation rules with confidence at least $\frac{4}{5}$:

$$\begin{aligned}
\{RS, B\} &\rightarrow \{M, LIB\} \\
\{RS, LIB\} &\rightarrow \{M, B\} \\
\{M, B\} &\rightarrow \{RS, LIB\} \\
\{B, LIB\} &\rightarrow \{RS, M\} \\
\{RS, M, B\} &\rightarrow \{LIB\} \\
\{RS, M, LIB\} &\rightarrow \{B\} \\
\{RS, B, LIB\} &\rightarrow \{M\} \\
\{M, B, LIB\} &\rightarrow \{RS\}
\end{aligned}$$

FP-Growth

Table 3: data with counts for each album

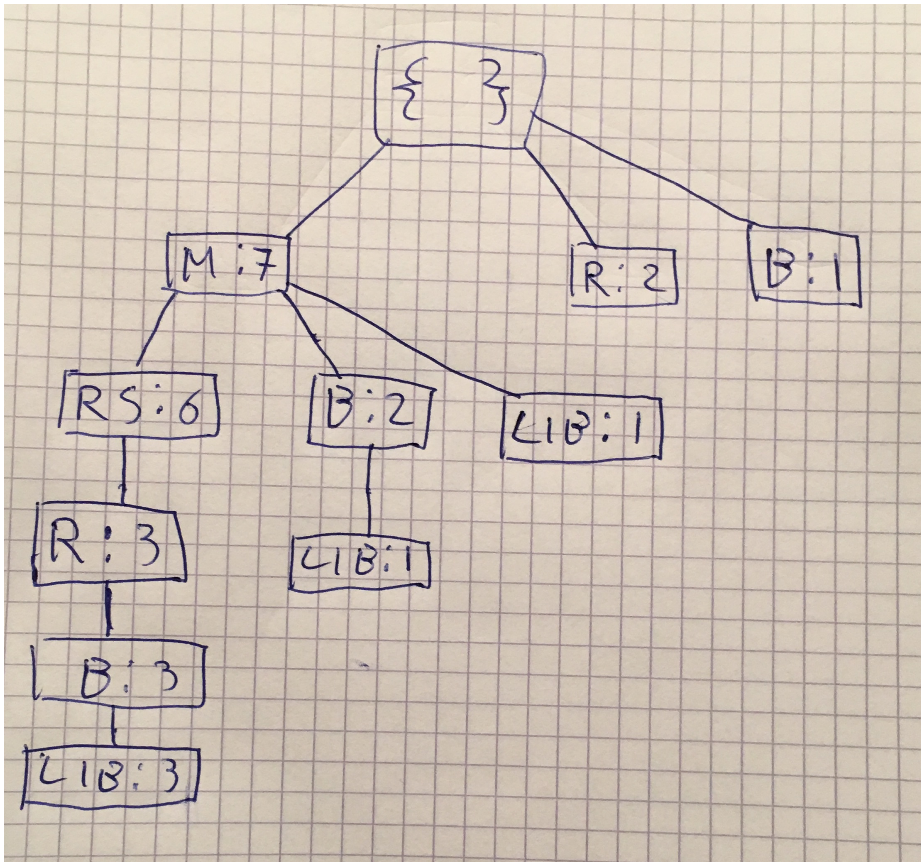
Abb	Album	z1	z2	z3	z4	z5	z6	z7	z8	z9	z10	frequency count
RS	Rubber Soul	1	0	0	0	1	1	1	1	1	0	6
R	Revolver	1	1	1	1	0	1	1	0	0	0	6
LH	Sgt. Pepper's LHCB	0	0	0	0	1	0	1	1	0	0	3
M	Magical Mystery Tour	1	1	0	0	1	1	1	1	1	0	7
B	The Beatles	1	0	0	0	1	1	1	1	0	1	6
Y	Yellow Submarine	0	1	0	0	0	0	1	0	1	0	3
A	Abbey Road	1	1	0	1	0	0	0	0	0	0	3
LIB	Let It Be	1	1	0	0	0	1	1	1	1	0	5

Table 4: transaction dataset sorted by frequency

Abb	Album	frequency count
M	Magical Mystery Tour	7
RS	Rubber Soul	6
R	Revolver	6
B	The Beatles	6
LIB	Let It Be	5
LH	Sgt. Pepper's LHCB	3
Y	Yellow Submarine	3
A	Abbey Road	3

γ from the first table we simply count how often each album occurs by counting the number of people that have each album. this an be seen in the rightmost colum. Ommiting the rest of the table and then sorting on frequency gives the transaction dataset sorted by frequency

δ we filter the data to only contain items with minimal support 2/5, which is at least 4 in 10 records thus LH, Y and A which only occur 3 times fall off. we sort by frequency and then count all sequences of occurrences and put this in a tree to construct the fptree



ϵ for each item we look at conditional pattern bases. this is done by looking at the possible paths that lead up to an item within the fp-tree

Table 5: conditional pattern base

Abb	Album	conditional pattern base
M	Magical Mystery Tour	{}
RS	Rubber Soul	{M:6}
R	Revolver	{M-RS:3, M:1}
B	The Beatles	{M:RS-R:3, M-RS:2}
LIB	Let It Be	{M-RS-R-TB:3, M-RS-TB:1, M-RS:1}

ζ to obtain conditional fp tree's we look at the conditional pattern bases and squash infrequent items, resulting in the following fp tree's

Table 6: conditional FP-tree

Abb	Album	conditional FP tree
M	Magical Mystery Tour	{}
RS	Rubber Soul	{M:6}
R	Revolver	{M:4}
B	The Beatles	{M:5, RS:5}
LIB	Let It Be	{M:5, RS:5}

3 Assignment 1C

3.1 Frequent Subsequence Mining

α)
For the single elements, we have the following initial candadates: $< a >, < b >, < c >, < d >, < e > and < f >$.

$s_{min} = 0.75 = 3/4$

Cand	Supp	
a	4/4	frequent subsequence
b	4/4	frequent subsequence
c	4/4	frequent subsequence
d	3/4	frequent subsequence
e	3/4	frequent subsequence
f	2/4	infrequent subsequence
g	1/4	infrequent subsequence

β)
From the frequent length-1 candidates, we combine them to create the following lenght-2 candidates.

	$< a >$	$< b >$	$< c >$	$< d >$	$< e >$
$< a >$	$< aa >$	$< ab >$	$< ac >$	$< ad >$	$< ae >$
$< b >$	$< ba >$	$< bb >$	$< bc >$	$< bd >$	$< be >$
$< c >$	$< ca >$	$< cb >$	$< cc >$	$< cd >$	$< ce >$
$< d >$	$< da >$	$< db >$	$< dc >$	$< dd >$	$< de >$
$< e >$	$< ea >$	$< eb >$	$< ec >$	$< ed >$	$< ee >$

	$< a >$	$< b >$	$< c >$	$< d >$	$< e >$
$< a >$		$< (ab) >$	$< (ac) >$	$< (ad) >$	$< (ae) >$
$< b >$			$< (bc) >$	$< (bd) >$	$< (be) >$
$< c >$				$< (cd) >$	$< (ce) >$
$< d >$					$< (de) >$
$< e >$					

We then check which of these candidates are frequent. The results are displayed in the table following table.

$s_{min} = 0.75 = 3/4$

Cand	Supp	
$<$ aa $>$	2/4	infrequent subsequence
$<$ ab $>$	4/4	frequent subsequence
$<$ ac $>$	4/4	frequent subsequence
$<$ ad $>$	2/4	infrequent subsequence
$<$ ae $>$	1/4	infrequent subsequence
$<$ ba $>$	2/4	infrequent subsequence
$<$ bb $>$	2/4	infrequent subsequence
$<$ bc $>$	3/4	frequent subsequence
$<$ bd $>$	2/4	infrequent subsequence
$<$ be $>$	1/4	infrequent subsequence
$<$ ca $>$	2/4	infrequent subsequence
$<$ cb $>$	3/4	frequent subsequence
$<$ cc $>$	3/4	frequent subsequence
$<$ cd $>$	1/4	infrequent subsequence
$<$ ce $>$	1/4	infrequent subsequence
$<$ da $>$	1/4	infrequent subsequence
$<$ db $>$	2/4	infrequent subsequence
$<$ dc $>$	3/4	frequent subsequence
$<$ dd $>$	0/4	infrequent subsequence
$<$ de $>$	1/4	infrequent subsequence
$<$ ea $>$	2/4	infrequent subsequence
$<$ eb $>$	2/4	infrequent subsequence
$<$ ec $>$	2/4	infrequent subsequence
$<$ ed $>$	1/4	infrequent subsequence
$<$ ee $>$	0/4	infrequent subsequence
$<$ (ab) $>$	3/4	frequent subsequence
$<$ (ac) $>$	1/4	infrequent subsequence
$<$ (ad) $>$	1/4	infrequent subsequence
$<$ (ae) $>$	1/4	infrequent subsequence
$<$ (bc) $>$	2/4	infrequent subsequence
$<$ (bd) $>$	0/4	infrequent subsequence
$<$ (be) $>$	0/4	infrequent subsequence
$<$ (cd) $>$	0/4	infrequent subsequence
$<$ (ce) $>$	0/4	infrequent subsequence
$<$ (de) $>$	0/4	infrequent subsequence

We find that the length-2 frequent candidates are: $<$ ab $>$, $<$ ac $>$, $<$ bc $>$, $<$ cb $>$, $<$ cc $>$, $<$ dc $>$, $<$ (ab) $>$.

γ)
We currently have $(((5*5)+4+3+2+1)=)$ 35 length-2 candidates. If we did not prune the two length-1 candidates, we would have gotten $(((7*7)+6+5+4+3+2+1))=)$ 70 length-2 candidates. This means that by pruning the two length-1 candidates, we removed 50% of the length-2 candidates.

δ)

Now we create length-3 candidates by combining the length-2 candidate according to the joint operator algorithm.

Joint operator						
s_1	s_2	result		s_1	s_2	result
< ab >	< ab >	-		< cc >	< ab >	-
< ab >	< ac >	-		< cc >	< ac >	-
< ab >	< bc >	< abc >		< cc >	< bc >	-
< ab >	< cb >	-		< cc >	< cb >	< ccb >
< ab >	< cc >	-		< cc >	< cc >	< ccc >
< ab >	< dc >	-		< cc >	< dc >	-
< ab >	< (ab) >	-		< cc >	< (ab) >	-
< ac >	< ab >	-		< dc >	< ab >	-
< ac >	< ac >	-		< dc >	< ac >	-
< ac >	< bc >	-		< dc >	< bc >	-
< ac >	< cb >	< acb >		< dc >	< cb >	< dcb >
< ac >	< cc >	< acc >		< dc >	< cc >	< dcc >
< ac >	< dc >	-		< dc >	< dc >	-
< ac >	< (ab) >	-		< dc >	< (ab) >	-
< bc >	< ab >	-		< (ab) >	< ab >	-
< bc >	< ac >	-		< (ab) >	< ac >	-
< bc >	< bc >	-		< (ab) >	< bc >	< (ab)c >
< bc >	< cb >	< bcb >		< (ab) >	< cb >	-
< bc >	< cc >	< bcc >		< (ab) >	< cc >	-
< bc >	< dc >	-		< (ab) >	< dc >	-
< bc >	< (ab) >	-		< (ab) >	< (ab) >	-
< cb >	< ab >	-				
< cb >	< ac >	-				
< cb >	< bc >	< cbc >				
< cb >	< cb >	-				
< cb >	< cc >	-				
< cb >	< dc >	-				
< cb >	< (ab) >	-				

We now have the following length-3 candidates:< abc >, < acb>, < acc >, <bcb >, <bcc >, < cbc >, < ccb >, < ccc >, < dcb >, < dcc > and < (ab)c >.

ϵ) A subsequence s is closed if no strict supersequence s' for s $\subset$ s' such that s and s' have the same support. In order to check whether < abc > is closed within Ω, We first check what the support is of < abc >. We find that the support for < abc > is equal to 2/4. We now generate all possible supersequences for < abc > with a minimal length. To do this we create length-4 subsequence which consists of < abc > and a single element either added to the front or the back. So we get:

front	back	Result	support
< abc >	< a >	< abca >	0/4
< abc >	< b >	< abcb >	0/4
< abc >	< c >	< abcc >	1/4
< abc >	< d >	< abcd >	1/4
< abc >	< e >	< abce >	0/4
< abc >	< f >	< abcf >	1/4
< abc >	< g >	< abcg >	0/4
< a >	< abc >	< aabc >	0/4
< b >	< abc >	< babc >	0/4
< c >	< abc >	< cabc >	0/4
< d >	< abc >	< dabc >	0/4
< e >	< abc >	< eabc >	1/4
< f >	< abc >	< fabc >	0/4
< g >	< abc >	< gabc >	1/4

We know that the support of a subsequence can only decrease or stay equal if you add another candidate to it. So we know that there exists no supersequence of < abc > which has the same support as < abc >. Thus we can conclude that < abc > is closed within Ω

Now we do the same for < adc >. We find that the support for < adc > is 2/4.We now generate all possible supersequences for < adc > with a minimal length. To do this we create length-4 subsequence which consists of < adc > and a single element either added to the front or the back. So we get:

front	back	Result	support
< adc >	< a >	< adca >	0/4
< adc >	< b >	< adcb >	1/4
< adc >	< c >	< adcc >	0/4
< adc >	< d >	< adcd >	0/4
< adc >	< e >	< adce >	0/4
< adc >	< f >	< adcf >	0/4
< adc >	< g >	< adcg >	0/4
< a >	< adc >	< aadc >	1/4
< b >	< adc >	< badc >	1/4
< c >	< adc >	< cadc >	1/4
< d >	< adc >	< dadc >	0/4
< e >	< adc >	< eadc >	1/4
< f >	< adc >	< fadc >	1/4
< g >	< adc >	< gadc >	1/4

We know that the support of a subsequence can only decrease or stay equal if you add another candidate to it. So we know that there exists no supersequence of < adc > which has the same support as < adc >. Thus we can conclude that < adc > is closed within Ω

3.2 Frequent Subgraph Mining

ζ) Graph representations
The graphs G_1, G_2, G_3 are characterized by the vertex, edge and label sets below. The labels of vertices are characterized by the color of the node (Red, Black, Blue), while the edges are characterized by the line type (solid, dashed). Note that for convenience (i.e. ease of finding a black pen and better visibility) color of gray node is assumed to be black.

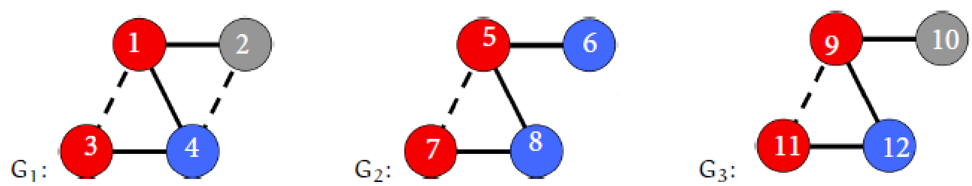


Figure 1: Graphs

- G_1
  $G_1 = (V(G_1), E(G_1)$
  $V(G_1) = \{V_1, V_2, V_3, V_4\}$
  $E(G_1) = \{E_1 = (V_1, V_2), E_2 = (V_1, V_3), E_3 = (V_1, V_4), E_4 = (V_2, V_4), E_5 = (V_3, V_4)\}$
  $l(V_1) = Red,\ \ l(V_2) = Black,\ \ l(V_3) = Red,\ \ l(V_4) = Blue$
  $l(E_1) = Solid,\ \ l(E_2) = Dashed,\ \ l(E_3) = Solid,\ \ l(E_4) = Dashed,\ \ l(E_5) = Solid$

- G_2
  $G_2 = (V(G_2), E(G_2)$
  $V(G_2) = \{V_5, V_6, V_7, V_8\}$
  $E(G_2) = \{E_6 = (V_5, V_6), E_7 = (V_5, V_7), E_8 = (V_5, V_8), E_9 = (V_7, V_8)\}$
  $l(V_5) = Red,\ \ l(V_6) = Blue,\ \ l(V_7) = Red,\ \ l(V_8) = Blue$
  $l(E_6) = Solid,\ \ l(E_7) = Dashed,\ \ l(E_8) = Solid,\ \ l(E_9) = Solid$

- G_3
  $G_3 = (V(G_3), E(G_3)$
  $V(G_3) = \{V_9, V_{10}, V_{11}, V_{12}\}$
  $E(G_3) = \{E_{10} = (V_9, V_{10}), E_{11} = (V_9, V_{11}), E_{12} = (V_9, V_{12}), E_{13} = (V_{11}, V_{12})\}$
  $l(V_9) = Red,\ \ l(V_{10}) = Black,\ \ l(V_{11}) = Red,\ \ l(V_{12}) = Blue$
  $l(E_{10}) = Solid,\ \ l(E_{11}) = Dashed,\ \ l(E_{12}) = Solid,\ \ l(E_{13}) = Solid$

η, θ) Frequent 1 and 2-subgraphs
Checking each of the edges if they exist at least in two graphs (since $s_{min} = 2/3$), the frequent 1-subgraphs are found as in Figure 2 below. Afterwards, these 1-subgraphs are paired to find 2-subgraphs. Note that since blue and black colors might be confused, we put a dot in blue nodes:

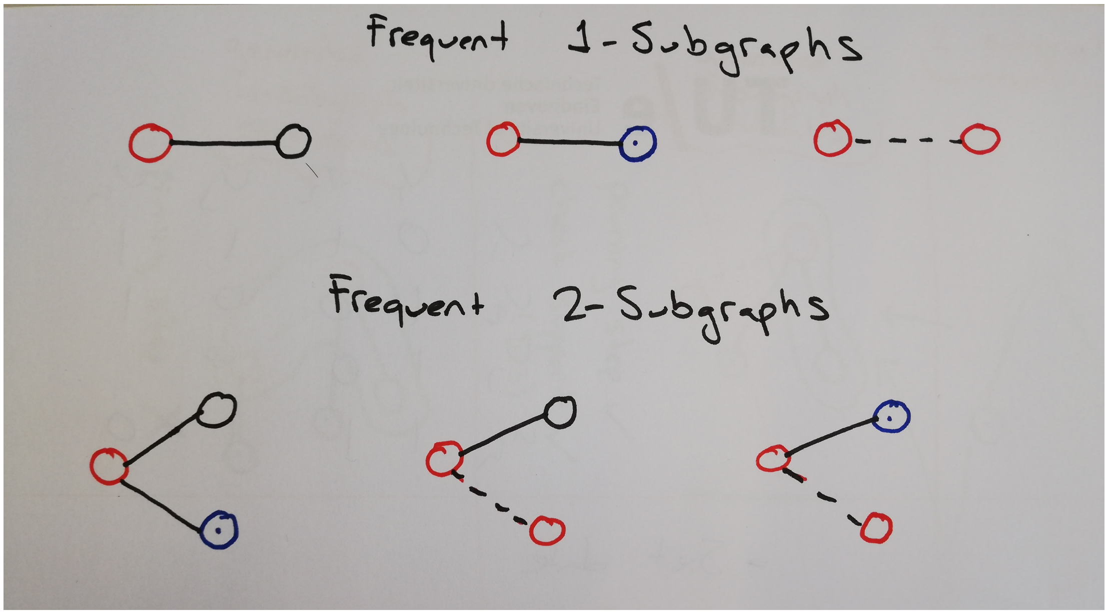


Figure 2: Frequent 1 and 2-subgraphs

ι) **FSG algorithm**

In this part, the set of all frequent subgraphs are found. As can be seen in Figure 3, we found two frequent 3-subgraphs and one frequent 4-subgraphs. As noted in the question description, only the graphs are drawn with a sense of what has been done at each step k. To summarize, at each step k;

1. Generating k-subgraphs: Two frequent (k-1)-subgraphs are selected, their cores are selected (the cores can be seen in the corresponding k-subgraph within green areas). Based on the cores, possible k-subgraph candidates are generated with the two (k-1) subgraphs (depicted with green arrows).

2. Pruning step: Each candidate k-subgraph is checked for pruning, i.e. whether all of the (k-1) subgraphs of the graph in question is frequent. The reason why they are pruned are excluded from this report for brevity.

3. Check support: For the graphs, who are not pruned, the support is checked. If there is a mapping of the k-subgraphs at least to two of the graphs G_1, G_2, G_3, the subgraph is deemed to be frequent.

Also it should be noted that, in 3-subgraph candidates, three of them are the same (i.e. automorphism), thus only one of them is selected for candidate generation in the next step.

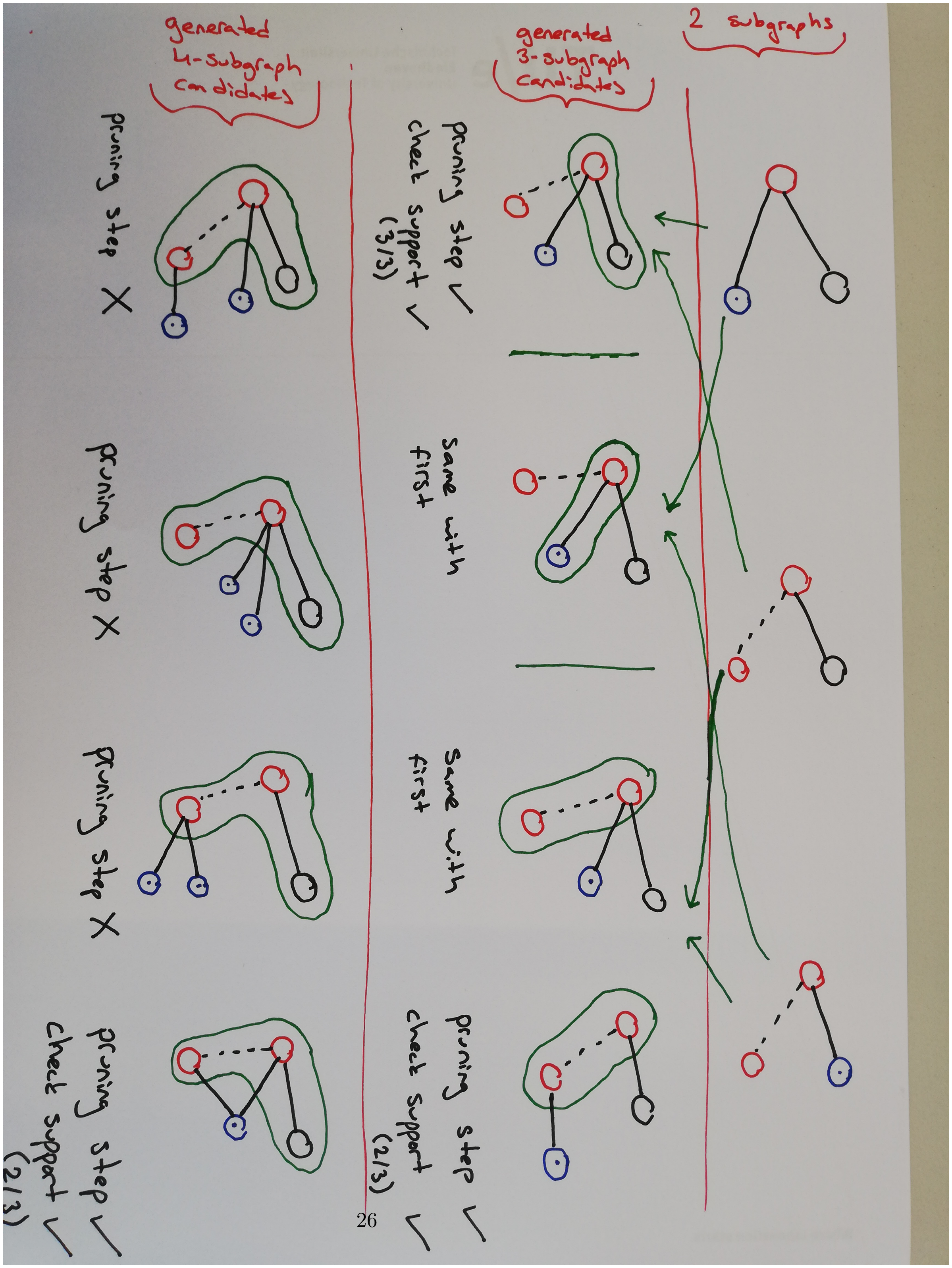


Figure 3: Subgraphs found by FSG

4 Assignment 1D

Axis-Parallel Subspace Clustering

α

	Cluster1	Cluster2	Cluster3
Setosa	50	0	0
Versicolor	0	48	2
Virginica	0	14	36

In total 134 out of 150 entries are being classified correctly.
The accuracy of this clustering is $\frac{134}{150} = 89\%$
Cluster 1 matches perfectly with Iris Setosa while the other 2 clusters make mistakes. Cluster 2 appears to be the least accurate one and Iris Virginica is most often wrongly classified.

β

	Cluster1	Cluster2	Cluster3
Setosa	50	0	0
Versicolor	0	38	12
Virginica	0	15	35

The accuracy of this clustering is $\frac{123}{150} = 82\%$
$\rightarrow$ While Iris Setosa is still identified correctly, the results clearly deteriorate with respect to the other plants

γ The best clustering is achieved with only "petallength" and "petalwidth". The result is as follows:

	Cluster1	Cluster2	Cluster3
Setosa	50	0	0
Versicolor	0	48	2
Virginica	0	4	46

The accuracy of this clustering is $\frac{144}{150} = 96\%$
Iris Virginica is the plant that ends up in the wrong cluster most often (62 times across 6 2D-combinations)
Most difficult records:

Record-#	# of errors
77	6
106	6
119	6
126	5
138	5
113	4
121	4
123	4
127	4

Records #77, #106 and #119 were never classified correctly.

Assignment 2

Linear regression

1

Temperature (c)	Power
13,42	468,82
20,77	442,85
8,29	483,26
30,98	433,59
31,96	433,04

2 For our data transformation we choose to first transform the temperature data from Celsius to Kelvin. This because Kelvin is the more scientific method of measuring temperature as its zero is also the absolute zero temperature. We then get the following table:

Temperature (K)	Power
286,42	468,82
293,77	442,85
281,29	483,26
303,98	433,59
304,96	433,04

From here we normalize the data to zero mean and unit standard deviation. This gives the following table which we will use for linear regression.

X-Normalized	Y
-0,731782159	468,82
-0,02998168	442,85
-1,221610248	483,26
0,944900345	433,59
1,038473742	433,04

3 We preformed the least square regression by following the slides from Lecture LinM: Linear Models. We got the following table:

X	Y
-0.73178	468,82
-0.02998	442,85
-1.22161	483,26
0.9449	433,59
1.038474	433,04

$\theta_0 + -0.73178\theta_1 = 468.82$
$\theta_0 + -0.02998\theta_1 = 442.85$
$\theta_0 + -1.22161\theta_1 1 = 483.26$
$\theta_0 + 0.9449\theta_1 = 433.59$
$\theta_0 + 1.038474\theta_1 = 433.04$

$SSE = [468.82 - (\theta_0 - 0.73178\theta_1)]^2 + [442.85 - (\theta_0 + -0.02998\theta_1)]^2 + [483.26 - (\theta_0 + -1.22161\theta_1)]^2 + [433.59 - (\theta_0 + 0.9449\theta_1)]^2 + [433.04 - (\theta_0 + 1.038474\theta_1)]^2 =$

$$\theta_0^2 - 1.463556\theta_0\theta_1 - 937.64\theta_0 + 0.535502\theta_1^2 - (937.64 * -0.73178\theta_1) + 468.82^2$$

$$+$$

$$\theta_0^2 - 0.05996\theta_0\theta_1 - 885.7\theta_0 + +0.0008988\theta_1^2 - (885.7 * -0.02998\theta_1) + 442.85^2$$

$$+$$

$$\theta_0^2 - 2.443222\theta_0\theta_1 - 966.52\theta_0 + 1.49233\theta_1^2 - (966.52 * -1.22161\theta_1) + 483.26^2$$

$$+$$

$$\theta_0^2 + 1.889984\theta_0\theta_1 - 867.18\theta_0 + 0.892874\theta_1^2 - 867.18 * 0.9449\theta_1 + 433.59^2$$

$$+$$

$$\theta_0^2 + 2.076950\theta_0\theta_1 - 866.08\theta_0 + 1.07843\theta_1^2 - 866.08 * 1.038474\theta_1 + 433.04^2$$

$$=$$

$$5\theta_0^2 + 0\theta_0\theta_1 - 4523.12\theta_0 + 4.000\theta_1^2 + 174.3905\theta_1 +$$

$$468.82^2 + 442.85^2 + 483.26^2 + 433.59^2 + 433.04^2$$

Now we differentiate the result for both $\theta_0\theta_1$. We get the following :

$$\frac{\partial SSE}{\partial \theta_0} = 0 = 10\theta_0 - 4523.12$$

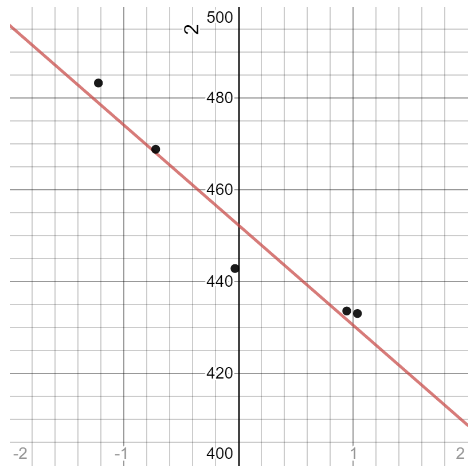
For this we find $\theta_0 = 452.312$

$$\frac{\partial SSE}{\partial \theta_1} = 0 = 8\theta_1 + 174.3905$$

For this we find that $\theta_1 = -21.7988$

Now we can calculate the regression parameter and plot the regression line with the data points.

$SSE(452.312, -21.7998) = (0.5561)^2 + (-10.1155)^2 + (4.3184)^2 + (1.8757)^2 + (3.3655)^2 = 136.13$



4 for linear regression we have a prediction output formula which is : $Y_{pred} = a + b * x$ where x are the normalized temperatures and y is the given output power. for the cost function we define: cost $J = \sum \frac{1}{2n} * (Y - Y_{pred})^2$ a squared cost function penalized big differences more then small ones. this should lead to a good fit. multiplication by $\frac{1}{2}$ is added because then this function will have a nice simple derivative. The partial derivatives will then be $\frac{\delta J}{\delta a} = Y - Y_{pred}$ and $\frac{\delta J}{\delta b} = (Y - Y_{pred}) * x$ the only thing that is left is to define the learning rate α which has been set to 0.25 through experimentation. this seems to give a reasonable pace at iterating and slowly converge through a optimum without risk of overshooting for initialisation for the variables a and b for our prediction line we have no way of knowing what they are in advance so they are arbitrarily chosen to both be equal to 1.

5 first iteration

a	b	x	y	y_{pred}	J	$\frac{\delta J}{\delta a}$	$\frac{\delta J}{\delta b}$
1	1	-0,73178	468,82	0,268218	109770,4	-468,552	342,8778
1	1	-0,02998	442,85	0,970018	97628,96	-441,88	13,2483
1	1	-1,22161	483,26	-0,22161	116877,2	-483,482	590,6261
1	1	0,9449	433,59	1,9449	93158,75	-431,645	-407,862
1	1	1,038474	433,04	2,038474	92881,16	-431,002	-447,584
Average					102063,3	-451,312	18,2613

the table is in similar format as that within the lecture slides. for each point we compute the error we made. we then average the errors with respect to the direction of a and b to obtain our final averaged sum over all of our data points. this is in the last row. from the losses we update our a and b values accordingly to the formulas $a = a - \alpha * \frac{\delta J}{\delta a}$ $b = b - \alpha * \frac{\delta J}{\delta b}$ and thus obtain $a = 1 - 0.25 * -451,31 = 113.828$ and $a = 1 - 0.25 * 18.26 =$

-3.56 thus we get the line:

$$y_{pred} = 113.828 + -3.56 * x$$

6 second iteration yields $y_{pred} = 198.45 + -7.21 * x$ and the following table of intermediary values

a	b	x	y	y_{pred}	J	$\frac{\delta J}{\delta a}$	$\frac{\delta J}{\delta b}$
113,828	-3,56534	-0,73178	468,82	116,4371	62086,87	-352,383	257,8676
113,828	-3,56534	-0,02998	442,85	113,9349	54092,57	-328,915	9,861428
113,828	-3,56534	-1,22161	483,26	118,1835	66640,44	-365,077	445,9812
113,828	-3,56534	0,9449	433,59	110,4591	52206,79	-323,131	-305,326
113,828	-3,56534	1,038474	433,04	110,1255	52136,89	-322,915	-335,338
Average					57432,72	-338,48	14,608

7 third iteration second iteration yields $y_{pred} = 261.91 + -16.08 * x$ and the following table of intermediary values

a	b	x	y	y_{pred}	J	$\frac{\delta J}{\delta a}$	$\frac{\delta J}{\delta b}$
198,449	-7,21762	-0,73178	468,82	203,7307	35136,16	-265,089	193,9876
198,449	-7,21762	-0,02998	442,85	198,6654	29813,06	-244,185	7,321065
198,449	-7,21762	-1,22161	483,26	207,2661	38086,31	-275,994	337,157
198,449	-7,21762	0,9449	433,59	191,6291	29272,55	-241,961	-228,629
198,449	-7,21762	1,038474	433,04	190,9537	29302,89	-242,086	-251,4
Average					32322.2	-253,864	11,687

8 31th iteration table:

a	b	x	y	y_{pred}	J	$\frac{\delta J}{\delta a}$	$\frac{\delta J}{\delta b}$
452,2314	-21,7985	-0,73178	468,82	468,1831	0,202805	-0,63687	0,466054
452,2314	-21,7985	-0,02998	442,85	452,885	50,35019	10,03496	-0,30086
452,2314	-21,7985	-1,22161	483,26	478,8606	9,677267	-4,39938	5,374326
452,2314	-21,7985	0,9449	433,59	431,634	1,9129	-1,95597	-1,84819
452,2314	-21,7985	1,038474	433,04	429,5943	5,936495	-3,44572	-3,57829
Average					13,615	-0,0805	0,0226

32th iteration table:

a	b	x	y	y_{pred}	J	$\frac{\delta J}{\delta a}$	$\frac{\delta J}{\delta b}$
452,2516	-21,8041	-0,73178	468,82	468,2074	0,187633	-0,61259	0,448282
452,2516	-21,8041	-0,02998	442,85	452,9053	50,55429	10,05528	-0,30147
452,2516	-21,8041	-1,22161	483,26	478,8877	9,558615	-4,37233	5,341278
452,2516	-21,8041	0,9449	433,59	431,6488	1,884044	-1,94116	-1,8342
452,2516	-21,8041	1,038474	433,04	429,6086	5,887392	-3,43144	-3,56346
Average					13,614	-0,0604	0,0180

at this point we see that there is very little difference between iterations so our gradient descent has iterated to a optimum location. thus we obtain $y_{pred} = 452.255 + -21.805 * x$

9 we see that linear regression and the least squares method both come to the same result as we would also expect since its trying to fit on the same data. linear regression iterates to a optimum location whereas leased squares method directly computes this point. in terms of run time it would therefor be favourable to use least squares because many iterations might be needed for the linear regression. Also linear regression takes a extra parameter alpha as input to define the 'learning' rate. This might also be impractical.

Polynomial regression

Let's start with finding the partial derivatives, which will help us find the rules for iterations. In general form we have;

$$h_\theta(x) = \theta_2 x^2 + \theta x + \theta_0$$
$$J(\theta) = \frac{1}{4n}\sum_{i=1}^{n}(h_\theta(x_i) - y_i)^4$$
$$\theta_j \leftarrow \theta_j - \alpha\frac{\partial}{\partial\theta_j}J(\theta)$$

Hence, we need partial derivatives for θ;

$$\frac{\partial}{\partial\theta}J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(h_\theta(x_i) - y_i)^3 \frac{\partial}{\partial\theta_j}(h_\theta(x_i) - y_i)$$
$$\frac{\partial}{\partial\theta}J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_2 x^2 + \theta x + \theta_0 - y_i\right)^3 \frac{\partial}{\partial\theta_j}\left(\theta_2 x^2 + \theta_1 x + \theta_0 - y_i\right)$$

Then for each θ_j;

$$\frac{\partial}{\partial\theta_0}J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_2 x^2 + \theta_1 x + \theta_0 - y_i\right)^3$$
$$\frac{\partial}{\partial\theta_1}J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_2 x^2 + \theta_1 x + \theta_0 - y_i\right)^3 (x_i)$$
$$\frac{\partial}{\partial\theta_2}J(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_2 x^2 + \theta_1 x + \theta_0 - y_i\right)^3 (x_i)^2$$

Then the update rules become;

$$\theta_0 \leftarrow \theta_0 - \frac{\alpha}{n}\sum_{i=1}^{n}\left(\theta_2 x^2 + \theta_1 x + \theta_0 - y_i\right)^3$$
$$\theta_1 \leftarrow \theta_1 - \frac{\alpha}{n}\sum_{i=1}^{n}\left(\theta_2 x^2 + \theta_1 x + \theta_0 - y_i\right)^3 (x_i)$$
$$\theta_2 \leftarrow \theta_2 - \frac{\alpha}{n}\sum_{i=1}^{n}\left(\theta_2 x^2 + \theta_1 x + \theta_0 - y_i\right)^3 (x_i)^2$$

For the numerical analysis, first, we tried to use Excel as in linear regression to find a solution. However, this time it was not easy due to two problems: With the previous learning rate of 0.05, overshooting occurred, i.e. the cost function started to increase and exploded in some point. On the other hand, when we decreased the learning rate considerably to avoid overshooting, say to 0.000001, we needed much more number of iterations. Therefore, we coded gradient descent in C, which is super fast. We picked $\alpha = 0.000001$ by trial and error (Larger learning rates led to overshooting), and observed that after 10 000 000 iterations the cost function converges. We chose two different initial points:

1. $\theta_0 = 0, \theta_1 = 0, \theta_2 = 0$
2. The solution to linear regression,$\theta_0 = 452, \theta_1 = -21, \theta_2 = 0$.

As a result, for both initial points, we reached the following:

$$
\begin{aligned}
J(\theta) &= 7.659 \\
\theta_0 &= 444.79 \\
\theta_1 &= -22.03 \\
\theta_2 &= 9.26
\end{aligned}
$$

Table 7: Results of polynomial regression

X	Y	$\hat{Y}$	Error squared
-0.731782159	468.82	465.8751178	8.672331014
-0.02998168	442.85	445.4602334	6.813318491
-1.221610248	483.26	485.530062	5.153181277
0.944900345	433.59	432.2410256	1.819731822
1.038473742	433.04	431.8981982	1.303711393
		Total	23.762274

As we can see, the squared error is significantly less than that of linear regression, which shows that a polynomial equation may be a better idea.

We plotted the lines at each iteration fit the data better and better for $\alpha = 0.000001$ and initial point (0,0,0) as in Figure 4. Note that the lines correspond to the iterations 1,2,3,....10,100,1000,10000,...,10000000. We chose these points due to the excess number of iterations.

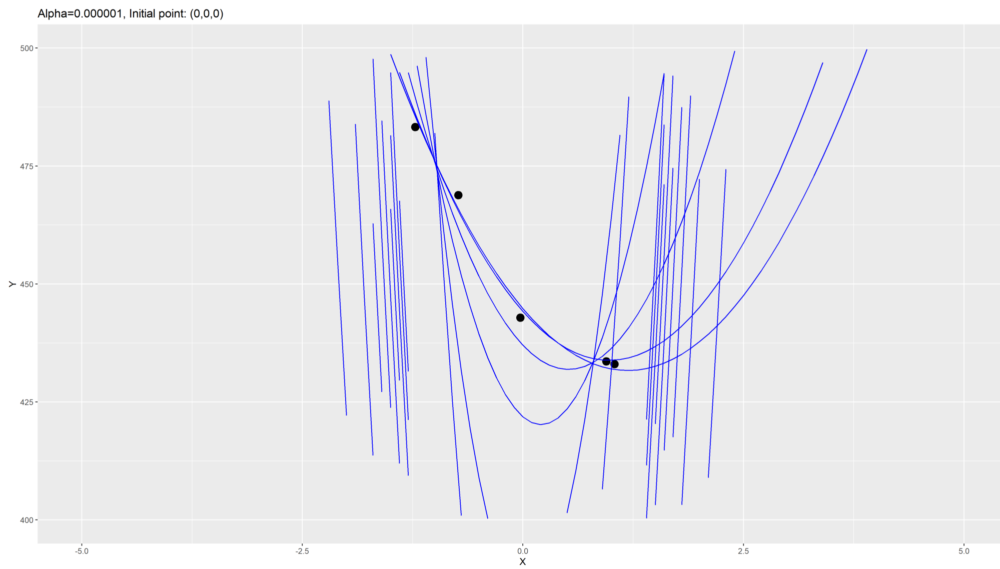


Figure 4: Fits vs the data points with initial point (0,0,0)

In comparison, we also plotted in the same way, but using the result of linear regression as the initial point as in Figure 5. This time, we observe that the initial point fit is better and improvements are incremental. Note that the end result is the same.

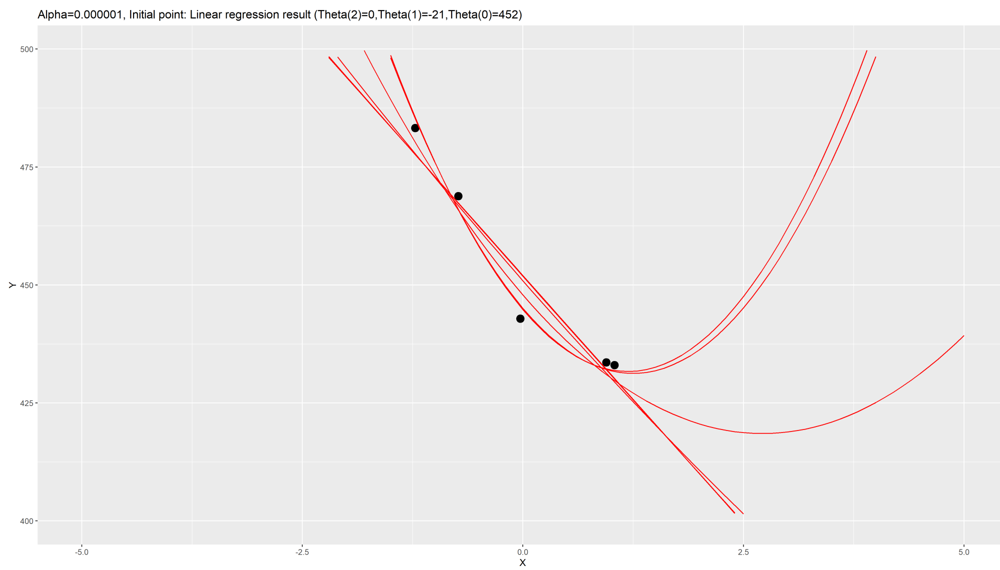


Figure 5: Fits vs the data points with linear regression solution as the initial point

To conclude, in comparison with linear regression;

- In both cases, the cost function decreases incrementally. First, the rate is higher then the decrease rate stabilizes.
- We had overshooting problem in polynomial regression.
- We needed a much smaller learning rate in polynomial regression.
- We needed much more number of iterations polynomial regression.
- We observed that the initial starting point is important for converging faster in polynomial regression although we did not test this for linear regression.
- We observed that the polynomial fit is better than linear regression fit for this dataset.

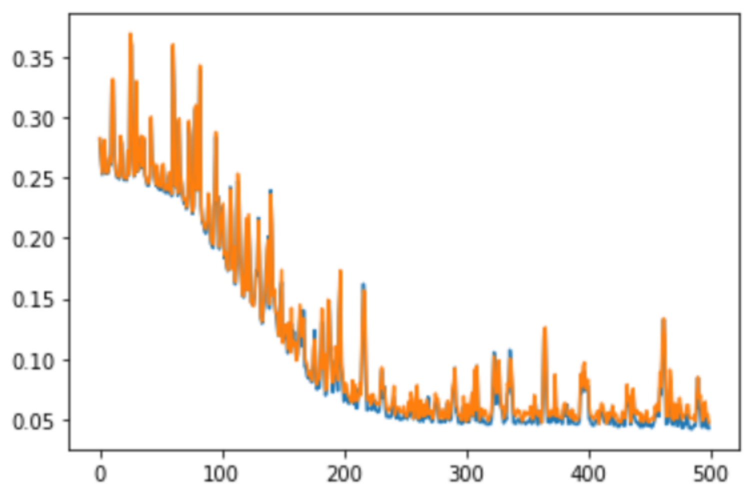
Assignment 3

MLP

1 $\phi^{(0)}$: ReLu
$\phi^{(1)}$: ReLu
$\phi^{(2)}$: ReLu
We chose mean squared error as our loss function.
These activation and loss functions are easy to implement and absolutely sufficient for this task.

2 We tried out different learning rates and chose to go with 0.01. Lower learning rates converge more smoothly but also take longer to converge.
For the batchsize we chose 10 batches of 41 datapoints each. With smaller batches it converged slower but still just fine. With larger batches it did not seem to converge correctly.
First we initialized our weights randomly between -1 and 1, but we noticed that for certain configurations it got stuck at a local minimum and did not improve towards an error of 0. So we took a random initialization that worked and hard coded it as our starting weights. Biases are initialized as 1.0.

3 Plot: (blue=training data; orange=validation data)



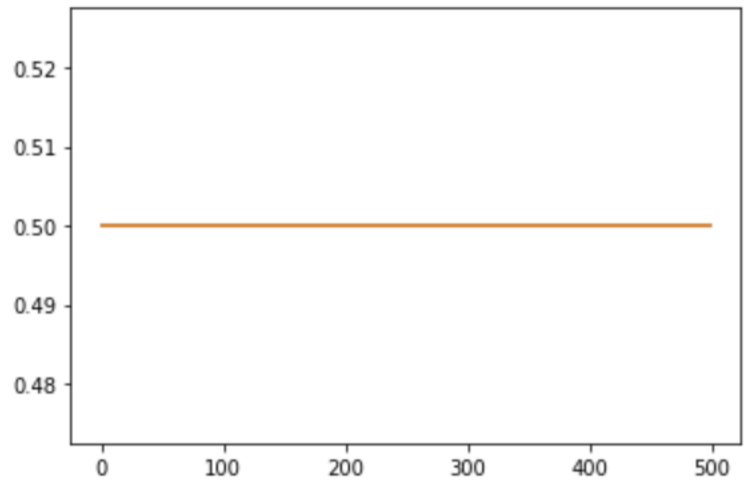
We chose an error of 0.05 towards the training data as our stopping criterium since that is roughly what it converges towards.
$Accuracy = \frac{80}{82} = 97.6\%$
Confusion Matrix:

	Predicted: 1's	Predicted 0's	Total
Actual 1's	40	1	41
Actual 0's	1	40	41
Total	41	41	

4 add source code

Hyperparameters Optimization

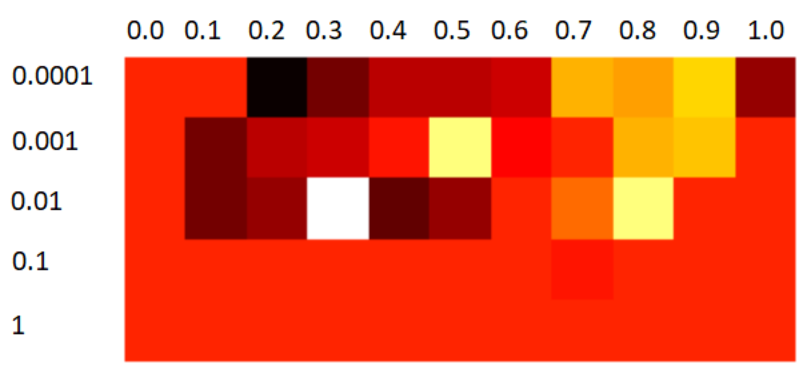
1 Loss over time:



$Accuracy = \frac{41}{82} = 50\%$ Confusion Matrix:

	Predicted: 1's	Predicted 0's	Total
Actual 1's	0	41	41
Actual 0's	0	41	41
Total	0	82	

Training with these input values does not work at all for out configuration. Everything is classified as '0' and the weights are not being updated in a way that would improve it. This is because if all weights are 0 then the output will always be 0 and thus no gradient can be computed so the network collapses.



2

The following parameters prevent the neural net from learning at all:

- High learning rate

- Sigma=0

The best result is achieved at sigma=0.3 and decent results are around sigma=0.8.

3

4

Peer Review

Lizaveta Anikeyeva

Overall I feel like everyone contributed to the creation of the final version of the project. We had a relatively good communication and discussions, as we were transparent about how far we are with our tasks. Everyone in our group has a slightly different background so it definitely helped with some of the tasks.

Pieter Derks

We worked quite little together because the assignments (especially the first ones) were all quite small. Assignment sets 1A,1B,1C,1D were all split up and parts done individually. I feel everybody has done their part in that. Volkan, Dennis and me sat together for set 2. But after looking into it together most work there was also done separately. Fabian and Liza were supposed to pick up assignment set 3, however that set turned out to be a bit more difficult so we all set together at some point to discuss this and afterwards dennis helped out there alot. think liza could have done more on the assignment set 3.

overall: little teamwork, mostly because Assignments didn't lend themselves very well for actual teamwork. I'm happy with everybody's performance as everybody did finish their parts.

Volkan Gumuskaya

We mostly split the work in team members, and have other people check what is being done. For HW1 we shared the workload among people, and I think the workload was fair. For HW2 Pieter, Dennis and me worked mostly, and let HW3 for Liza and Fabian, which turned out to be the hardest one. In HW3, I joined the first meeting but could not contribute afterwards at all. Overall, we shared the workload among people based on their availability, no one complained about unfair work, from time to time people stepped up to do more. Everyone was nice, easygoing and respectful. None of us know each other before, so i think we did pretty well in that sense.

I would grade all the others the same, and let them judge my work.

Fabian Maurer

In the beginning we started working on the tasks together (1A and 1B) but the others decided that it made more sense to distribute work among the group. 5 people are probably too many to work together efficiently, so I respected that decision. Once we started distributing the work there were some differences between the members in how well they can work on their own. Pieter and Dennis did good work and went over what the others did and gave them feedback. Volkan could not attend some of our team meetings due to travelling, but offered his help where he could and did good work. In the end we were in quite a hurry

since HW3 turned out to be more work than we expected. It was mostly Dennis and me that put a lot of time into that this past week. Liza struggled with her tasks at first, but in the end offered a valuable contribution to HW3 that helped me get the backpropagation code right.
Overall a decent experience, I've been in groups that were much worse.

Dennis Rizvic

Overall, I think the teamwork within this group could have been better. During this course I felt that there was a lack of communication and sometimes even participation. Certain deadlines were not made for trivial reasons such as forgetting or "just not feeling it" and during the meetings we often had 2 people absent and the people that were present arrived late by at least 30 min. This made some group meetings ineffective. While the teamwork within this group was not that great, we did divide the tasks up so that we could work individually and thus the contribution of most group members was equal until the last two weeks.
The one except was Liza who was unable to do her exercises because of either negligence or being unaware on how to deal with the problem. This might have caused her to contribute less than the other group members, but it also made it hard to estimate the amount of work she put in. Nevertheless, she was also one of the group members who I felt was more active with communication within the group. Overall, I feel like everyone's contribution was sufficient and thus I would assign everyone the same grade.