1

Solutions for Tutorial exercises
Association Rule Mining.

Exercise I. Apriori

Trace the results of using the Apriori algorithm on the grocery store example with support threshold s=33.34% and confidence threshold c=60%. Show the candidate and frequent itemsets for each database scan. Enumerate all the final frequent itemsets. Also indicate the association rules that are generated and highlight the strong ones, sort them by confidence.

Transaction ID	Items
T1	HotDogs, Buns, Ketchup
T2	HotDogs, Buns
T3	HotDogs, Coke, Chips
T4	Chips, Coke
T5	Chips, Ketchup
T6	HotDogs, Coke, Chips

Solution:

Support threshold =33.34% => threshold is at least 2 transactions.

Applying Apriori

Pass (k)	Candidate k-itemsets and their support	Frequent k-itemsets
k=1	HotDogs(4), Buns(2), Ketchup(2), Coke(3), Chips(4)	HotDogs, Buns, Ketchup, Coke, Chips
k=2	{HotDogs, Buns}(2), ~~{HotDogs, Ketchup}~~(1), {HotDogs, Coke}(2), {HotDogs, Chips}(2), ~~{Buns, Ketchup}~~(1), ~~{Buns, Coke}~~(0), ~~{Buns, Chips}~~(0), ~~{Ketchup, Coke}~~(0), ~~{Ketchup, Chips}~~(1), {Coke, Chips}(3)	{HotDogs, Buns}, {HotDogs, Coke}, {HotDogs, Chips}, {Coke, Chips}
k=3	{HotDogs, Coke, Chips}(2)	{HotDogs, Coke, Chips}
k=4	{}	

Note that {HotDogs, Buns, Coke} *and* {HotDogs, Buns, Chips} *are not candidates when* k=3 *because their subsets* {Buns, Coke} *and* {Buns, Chips} *are not frequent.*

Note also that normally, there is no need to go to k=4 *since the longest transaction has only 3 items.*

<u>All Frequent Itemsets:</u> {HotDogs}, {Buns}, {Ketchup}, {Coke}, {Chips}, {HotDogs, Buns}, {HotDogs, Coke}, {HotDogs, Chips}, {Coke, Chips}, {HotDogs, Coke, Chips}.

<u>Association rules:</u>

{HotDogs, Buns} would generate: HotDogs → Buns (2/6=0.33, 2/4=0.5) and
Buns → HotDogs (2/6=0.33, 2/2=1);

{HotDogs, Coke} would generate: HotDogs → Coke (0.33, 0.5) and
Coke → HotDogs (2/6=0.33, 2/3=0.66);

{HotDogs, Chips} would generate: HotDogs → Chips (0.33, 0.5) and
Chips → HotDogs (2/6=0.33, 2/4=0.5);

{Coke, Chips} would generate; **Coke → Chips (3/6=0.5, 3/3=1)** and
Chips → Coke (3/6=0.5, 3/4=0.75);

{HotDogs, Coke, Chips} would generate: HotDogs → Coke ^ Chips (2/6=0.33, 2/4=0.5),
Coke → Chips ^ HotDogs (2/6=0.33, 2/3=0.66),
Chips → Coke ^ HotDogs (2/6=0.33, 2/4=0.5),
HotDogs ^ Coke → Chips(2/6=0.33, 2/2=1),
HotDogs ^ Chips → Coke(2/6=0.33, 2/2=1) and
Coke ^ Chips → HotDogs(2/6=0.33, 2/3=0.66).

With the confidence threshold set to 60%, the Strong Association Rules are (sorted by confidence):

1. Coke → Chips (0.5, 1)
2. Buns → HotDogs (0.33, 1);
3. HotDogs ^ Coke → Chips(0.33, 1)
4. HotDogs ^ Chips → Coke(0.33, 1)
5. Chips → Coke (0.5, 0.75);
6. Coke → HotDogs (0.33, 0.66);
7. Coke → Chips ^ HotDogs (0.33, 0.66)
8. Coke ^ Chips → HotDogs(0.33, 0.66).

Exercise 2. FP-tree and FP-Growth

a) Use the transactional database from the previous exercise with same support threshold and build a frequent pattern tree (FP-Tree). Show for each transaction how the tree evolves.
b) Use Fp-Growth to discover the frequent itemsets from this FP-tree.

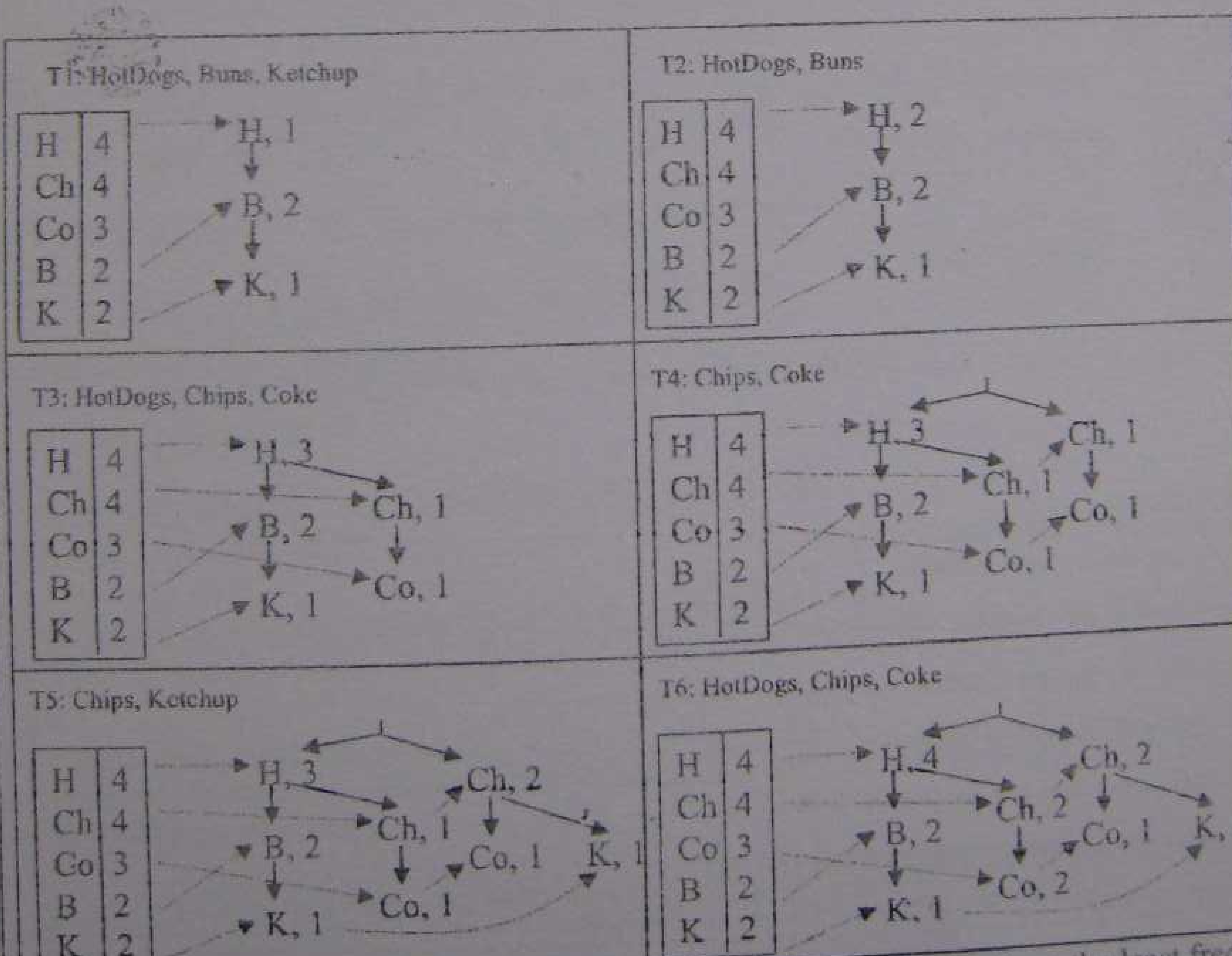
Solution:

a) The first scan of the database generates the list of frequent 1-itemsets and builds the header table where the items are sorted by frequency.

Error!

Item	Code	Support
HotDogs	H	4 = 66%
Chips	Ch	4 = 66%
Coke	Co	3 = 50%
Buns	B	2 = 33%
Ketchup	K	2 = 33%

The second scan is used to create the FP-tree. Each transaction is sorted by item support.



b) We need to build a conditional tree for each frequent item starting from the least frequent.
- For Ketchup (K), we have two branches H-B-K and Ch-K but since K has a support of 1 in each branch, this would eliminate all items (since support threshold is 2) leaving only <K:2>. This leads to the

③

discovery of {Ketchup} (2) as frequent item.
- For Buns (B), we have only one branch H-B. The sub-transaction {HotDogs, Buns} appears twice. We have thus the patterns <B:2, H:2> and <B:2>. This leads to the discovery of {HotDogs, Buns} (2) and {Buns}(2) as frequent itemsets.
- For Coke (Co), we have two branches: H-Ch-Co and Ch-Co resulting in the tree Co(3)→Ch(3)→H(2). We have thus 3 patterns: <Co:2, Ch:2, H:2>, <Co:3, Ch:3> and <Co:3>. This leads to the discovery of the following frequent itemsets: {Coke, Chips, HotDogs}(2), {Coke, Chips}(3) and {Coke}(3).
- For Chips (Ch), we have two paths H-Ch and Ch, giving the following tree Ch(4)→H(2). This gives the patterns <Ch:2, H:2> and <Ch:4>. Thus, the itemsets {Chips, HotDogs}(2) and {Chips}(4) are frequent.
- For HotDogs (H), The only and obvious pattern is <H:4> leading to the discovery of {HotDogs}(4) as frequent itemset.

All Frequent Itemsets (like in previous exercise): {HotDogs}, {Buns}, {Ketchup}, {Coke}, {Chips}, {HotDogs, Buns}, {HotDogs, Coke}, {HotDogs, Chips}, {Coke, Chips}, {HotDogs, Coke, Chips}.

Notice that there was no candidacy generation. Frequent itemsets were generated directly.

Exercise 3: Using WEKA

Load a dataset described with nominal attributes, e.g. weather.nominal. Run the Apriori algorithm to generate association rules.

Solution:

Running Weka with the default parameters:
Apriori –N 10 –T 0 –C 0.9 –D 0.05 –U 1.0 –M 0.1 –S -1.0

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -
1.0
Relation:     weather.symbolic
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
=== Associator model (full training set) ===

Apriori
=======

Minimum support: 0.15
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 47

Size of set of large itemsets L(3): 39

Size of set of large itemsets L(4): 6

Best rules found:

 1. humidity=normal windy=FALSE 4 ==> play=yes 4    conf:(1)
 2. temperature=cool 4 ==> humidity=normal 4    conf:(1)
 3. outlook=overcast 4 ==> play=yes 4    conf:(1)
```

```
4. temperature=cool play=yes 3 ==> humidity=normal 3    conf:(1)
5. outlook=rainy windy=FALSE 3 ==> play=yes 3    conf:(1)
6. outlook=rainy play=yes 3 ==> windy=FALSE 3    conf:(1)
7. outlook=sunny humidity=high 3 ==> play=no 3    conf:(1)
8. outlook=sunny play=no 3 ==> humidity=high 3    conf:(1)
9. temperature=cool windy=FALSE 2 ==> humidity=normal play=yes 2    conf:(1)
10. temperature=cool humidity=normal windy=FALSE 2 ==> play=yes 2    conf:(1)
```

Exercise 4: **Apriori and FP-Growth** **(to be done at your own time, not in class)**

Giving the following database with 5 transactions and a minimum support threshold of 60% and a minimum confidence threshold of 80%, find all frequent itemsets using (a) Apriori and (b) FP-Growth. (c) Compare the efficiency of both processes. (d) List all strong association rules that contain "A" in the antecedent (Constraint). (e) Can we use this constraint in the frequent itemset generation phase?

TID	Transaction
T1	{A, B, C, D, E, F}
T2	{B, C, D, E, F, G}
T3	{A, D, E, H}
T4	{A, D, F, I, J}
T5	{B, D, E, K}

9

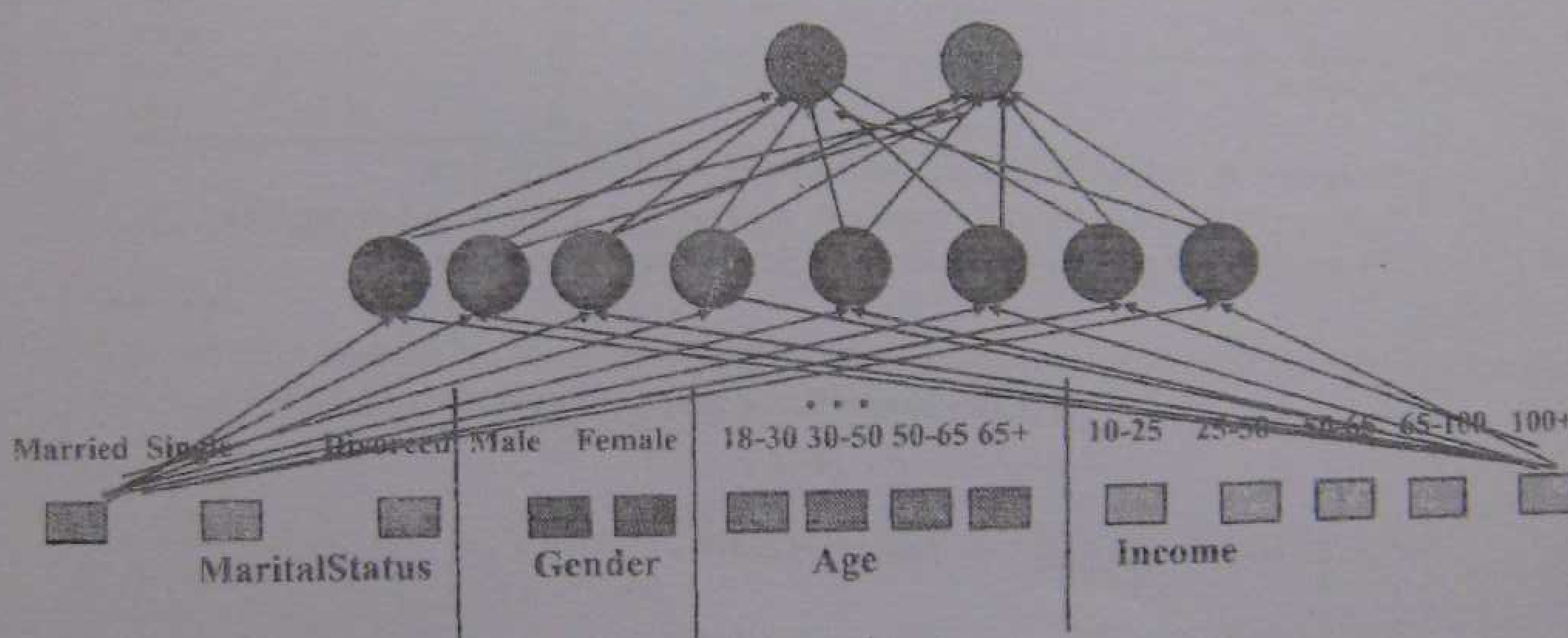
Solutions for Tutorial exercises
Backpropagation neural networks, Naïve Bayes, Decision Trees, k-NN, Associative Classification.

Exercise 1.

Suppose we want to classify potential bank customers as good creditors or bad creditors for loan applications. We have a training dataset describing past customers using the following attributes: Marital status {married, single, divorced}, Gender {male, female}, Age {[18..30[, [30..50[, [50..65[, [65+]}, Income {[10K..25K[, [25K..50K[, [50K..65K[, [65K..100K[, [100K+]}.
Design a neural network that could be trained to predict the credit rating of an applicant.

Solution:

We have 2 classes, good creditor and bad creditor. This means we would need two nodes in the output layer. There are 4 variables: Marital Status, Gender, Age and Income. However, since we have 3 values for Marital status, 2 values for Gender, 4 intervals for Age and 5 intervals for Income, we would have 14 neuron units in the input layer. In the hidden layer we can have $(14+2)/2=8$ neurons. The architecture of the neural networks could look like this.

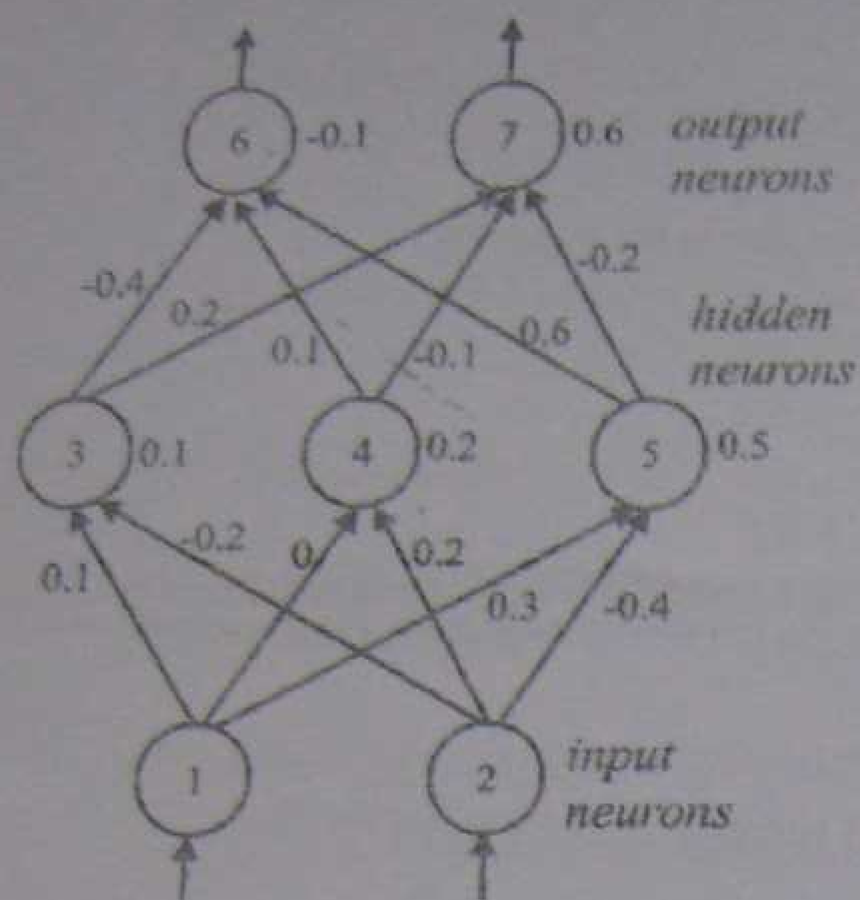


The weights are initialized with random values.

Are there other possible architectures?

Exercise 2.

Given the following neural network with initialized weights as in the picture, explain the network architecture knowing that we are trying to distinguish between nails and screws and an example of training tupples is as follows: T1{0.6, 0.1, nail}, T2{0.2, 0.3, screw}.



Let the learning rate η be 0.1 and the weights be as indicated in the figure above. Do the forward propagation of the signals in the network using T1 as input, then perform the back propagation of the error. Show the changes of the weights.

Solution:

What encoding of the outputs?
10 for class "nail", 01 for class "screw"

Forward pass for T1 - calculate the outputs o6 and o7

o1=0.6, o2=0.1, target output 1 0, i.e. class "nail"

Activations of the hidden units:

net3= o1 *w13+ o2*w23+b3=0.6*0.1+0.1*(-0.2)+0.1=0.14
o3=1/(1+e-net3) =0.53

net4= o1 *w14+ o2*w24+b4=0.6*0+0.1*0.2+0.2=0.22
o4=1/(1+e-net4) =0.55

net5= o1 *w15+ o2*w25+b5=0.6*0.3+0.1*(-0.4)+0.5=0.64
o5=1/(1+e-net5) =0.65

Activations of the output units:

net6= o3 *w36+ o4*w46+ o5*w56 +b6=0.53*(-0.4)+0.55*0.1+0.65*0.6-0.1=0.13
o6=1/(1+e-net6) =0.53

net7= o3 *w37+ o4*w47+ o5*w57 +b7=0.53*0.2+0.55*(-0.1)+0.65*(-0.2)+0.6=0.52
o7=1/(1+e-net7) =0.63

Backward pass for T1 - calculate the output errors δ6 and δ7
(note that d6=1, d7=0 for class "nail")

$\delta 6 = (d6\text{-}o6) * o6 * (1\text{-}o6)=(1\text{-}0.53)*0.53*(1\text{-}0.53)=0.12$
$\delta 7 = (d7\text{-}o7) * o7 * (1\text{-}o7)=(0\text{-}0.63)*0.63*(1\text{-}0.63)=-0.15$

Calculate the new weights between the hidden and output units (η=0.1)
$\Delta w36 = \eta * \delta 6 * o3 = 0.1*0.12*0.53=0.006$
$w36new = w36old + \Delta w36 = -0.4+0.006=-0.394$

$\Delta w37 = \eta * \delta 7 * o3 = 0.1*\text{-}0.15*0.53=-0.008$
$w37new = w37old + \Delta w37 = 0.2\text{-}0.008=-0.19$

Similarly for w46new , w47new , w56new and w57new

For the biases b6 and b7 (remember: biases are weights with input 1):
$\Delta b6 = \eta * \delta 6 * 1 = 0.1*0.12=0.012$
$b6new = b6old + \Delta b6 = -0.1+0.012=-0.012$
Similarly for b7

Calculate the errors of the hidden units δ3, δ4 and δ5
$\delta 3 = o3 * (1\text{-}o3) * (w36 * \delta 6 + w37 * \delta 7) = 0.53*(1\text{-}0.53)(-0.4*0.12+0.2*(-0.15))=-0.019$

Similarly for δ4 and δ5

Calculate the new weights between the input and hidden units (η=0.1)

$\Delta w13 = \eta * \delta 3 * o1 = 0.1*(-0.019)*0.6=-0.0011$
$w13new = w13old + \Delta w13 = 0.1\text{-}0.0011=0.0989$

Similarly for w23new , w14new , w24new , w15new and w25new; b3, b4 and b6

Repeat the same procedure for the other training examples
Forward pass for T2...backward pass for T2...

Exercise 3.
Why is the Naïve Bayesian classification called "naïve"?

Answer: Naïve Bayes assumes that all attributes are: 1) equally important and 2) independent of one another given the class.

Exercise 4. Naïve Bayes for data with nominal attributes
Given the training data in the table below (*Buy Computer* data), predict the class of the following new example using Naive Bayes classification: age<=30, income=medium, student=yes, credit-rating=fair

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31 . . . 40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31 . . . 40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31 . . . 40	medium	no	excellent	yes
13	31 . . . 40	high	yes	fair	yes
14	>40	medium	no	excellent	no

Solution:

E= age<=30, income=medium, student=yes, credit-rating=fair
E_1 is age<=30, E2 is income=medium, student=yes, E4 is credit-rating=fair
We need to compute P(yes|E) and P(no|E) and compare them.

$$P(yes \mid E) = \frac{P(E_1 \mid yes) P(E_2 \mid yes) P(E_3 \mid yes) P(E_4 \mid yes) P(yes)}{P(E)}$$

P(yes)=9/14=0.643 P(no)=5/14=0.357

P(E1|yes)=2/9=0.222 P(E1|no)=3/5=0.6
P(E2|yes)=4/9=0.444 P(E2|no)=2/5=0.4
P(E3|yes)=6/9=0.667 P(E3|no)=1/5=0.2
P(E4|yes)=6/9=0.667 P(E4|no)=2/5=0.4

$$P(yes \mid E) = \frac{0.222\ 0.444\ 0.667\ 0.668\ 0.443}{P(E)} = \frac{0.028}{P(E)} \qquad P(no \mid E) = \frac{0.6\ 0.4\ 0.2\ 0.4\ 0.357}{P(E)} = \frac{0.007}{P(E)}$$

Hence, the Naïve Bayes classifier predicts buys_computer=yes for the new example.

Exercise 5.* Applying Naïve Bayes to data with numerical attributes and using the Laplace correction *(to be done at your own time, not in class)
Given the training data in the table below (*Tennis* data with some numerical attributes), predict the class of the following new example using Naïve Bayes classification:
outlook=overcast, temperature=60, humidity=62, windy=false.

3

Tip. You can use Excel or Matlab for the calculations of logarithm, mean and standard deviation. Matlab is installed on our undergraduate machines. The following Matlab functions can be used: `log2` – logarithm with base 2, `mean` – mean value, `std` – standard deviation. Type `help <function name>` (e.g. `help mean`) for help on how to use the functions and examples.

outlook	temperature	humidity	windy	play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	60	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

Solution:

First, we need to calculate the mean μ and standard deviation σ values for the numerical attributes.
X_i, $i=1..n$ – the i-th measurement, n-number of measurements

$$\mu = \frac{\sum_{i=1}^{n} X_i}{n}$$

$$\sigma^2 = \frac{\sum_{i=1}^{n} (X_i - \mu)^2}{n-1}$$

μ_temp_yes=73, σ_temp_yes=6.2; μ_temp_no=74.6, σ_temp_no=8.0

μ_hum_yes=79.1, σ_temp_yes=10.2; μ_hum_no=86.2, σ_temp_no=9.7

Second, to calculate f(temperature=60|yes), f(temperature=60|no), f(humidity=62|yes) and f(humidity=62|no) using the probability density function for the normal distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$f(temperature = 60 \mid yes) = \frac{1}{6.2\sqrt{2\pi}} e^{-\frac{(60-73)^2}{2\,(6.2)^2}} = 0.071$$

$$f(temperature = 60 \mid no) = \frac{1}{8\sqrt{2\pi}} e^{-\frac{(60-74.6)^2}{2\,8^2}} = 0.0094$$

$$f(humidity = 62 \mid yes) = \frac{1}{10.2\sqrt{2\pi}} e^{-\frac{(62-79.1)^2}{2\,(10.2)^2}} = 0.0096$$

$$f(humidity = 62 \mid no) = \frac{1}{9.7\sqrt{2\pi}} e^{-\frac{(62-86.2)^2}{2\,(9.7)^2}} = 0.0018$$

Third, we can calculate the probabilities for the nominal attributes:

P(yes)=9/14=0.643 P(no)=5/14=0.357

P(outlook=overcast|yes)=4/14=0.286 P(outlook=overcast|no)=0/5=0

P(windy=false|yes)=6/9=0.667 P(windy=false|no)=2/5=0.4

As P(outlook=overcast|no)=0, we need to use a Laplace estimator for the attribute outlook. We assume that the three values (sunny, overcast, rainy) are equally probable and set μ=3:

$$P(outlook = overcast \mid yes) = \frac{4+1}{9+3} = \frac{5}{12} = 0.4167$$

$$P(outlook = overcast \mid no) = \frac{0+1}{5+3} = \frac{1}{8} = 0.125$$

Fourth, we can calculate the final probabilities:

$$P(yes \mid E) = \frac{0.4167 * 0.0071 * 0.0096 * 0.667 * 0.643}{P(E)} = \frac{1.22 * 10^{-5}}{P(E)}$$

$$P(no \mid E) = \frac{0.125 * 0.0094 * 0.0018 * 0.4 * 0.357}{P(E)} = \frac{3.02 * 10^{-7}}{P(E)}$$

Therefore, the Naïve Bayes classifier predicts play=yes for the new example.

Exercise 6. ***Using Weka** (to be done at your own time, not in class)*

Load iris data (iris.arff). Choose 10-fold cross validation. Run the Naïve Bayes and Multi-layer percepton (trained with the backpropagation algorithm) classifiers and compare their performance. Which classifier produced the most accurate classification? Which one learns faster?

Exercise 7. ***k-Nearest neighbours***

Given the training data in Exercise 4 (*Buy Computer* data), predict the class of the following new example using k-Nearest Neighbour for k=5: age<=30, income=medium, student=yes, credit-rating=fair. For similarity measure use a simple match of attribute values: Similarity(A,B)=

(7)

$\sum_{i=1}^{4} w_i * \partial(a_i, b_i) \Big/ 4$ where $\partial(a_i, b_i)$ is 1 if a_i equals b_i and 0 otherwise. a_i and b_i are either *age*, *income*, *student* or *credit_rating*. Weights are all 1 except for income it is 2.

Solution:

RID	age	income	student	credit_rating	Class: buys_computer
1	<=30	high	no	fair	no
2	<=30	high	no	excellent	no
3	31…40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31…40	low	yes	excellent	yes
8	<=30	medium	no	fair	no
9	<=30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	<=30	medium	yes	excellent	yes
12	31…40	medium	no	excellent	yes
13	31…40	high	yes	fair	yes
14	>40	medium	no	excellent	no

RID	Class	Distance to New
1	No	(1+0+0+1)/4=0.5
2	No	(1+0+0+0)/4=0.25
3	Yes	(0+0+0+1)/4=0.25
4	Yes	(0+2+0+1)/4=0.75
5	Yes	(0+0+1+1)/4=0.5
6	No	(0+0+1+0)/4=0.25
7	Yes	(0+0+1+0)/4=0.25
8	No	(1+2+0+1)/4=1
9	Yes	(1+0+1+1)/4=0.75
10	Yes	(0+2+1+1)/4=1
11	Yes	(1+2+1+0)/4=1
12	Yes	(0+2+0+0)/4=0.5
13	Yes	(0+0+1+1)/4=0.5
14	No	(0+2+0+0)/4=0.5

Among the five nearest neighbours four are from class *Yes* and one from class *No*. Hence, the k-NN classifier predicts buys_computer=yes for the new example.

Exercise 8. Decision trees

Given the training data in Exercise 4 (*Buy Computer* data), build a decision tree and predict the class of the following new example: age<=30, income=medium, student=yes, credit-rating=fair.

Solution:

First check which attribute provides the highest Information Gain in order to split the training set based on that attribute. We need to calculate the expected information to classify the set and the entropy of each attribute. The information gain is this mutual information minus the entropy:

The mutual information of the two classes $I(S_{Yes}, S_{no}) = I(9,5) = -9/14 \log_2(9/14) - 5/14 \log_2(5/14) = 0.94$

- For Age we have three values $age_{<=30}$ (2 yes and 3 no), $age_{31..40}$ (4 yes and 0 no) and $age_{>40}$ (3 yes 2 no)

$$
\begin{aligned}
Entropy(age) &= 5/14\,(-2/5 \log(2/5) - 3/5 \log(3/5)) + 4/14\,(0) + 5/14\,(-3/5 \log(3/5) - 2/5 \log(2/5)) \\
&= 5/14(0.9709) + 0 + 5/14(0.9709) \\
&= 0.6935
\end{aligned}
$$

$Gain(age) = 0.94 - 0.6935 = 0.2465$

- For Income we have three values $income_{high}$ (2 yes and 2 no), $income_{medium}$ (4 yes and 2 no) and $income_{low}$ (3 yes 1 no)

$$
\begin{aligned}
Entropy(income) &= 4/14(-2/4 \log(2/4) - 2/4 \log(2/4)) + 6/14\,(-4/6 \log(4/6) - 2/6 \log(2/6)) \\
&\quad + 4/14\,(-3/4 \log(3/4) - 1/4 \log(1/4)) \\
&= 4/14\,(1) + 6/14\,(0.918) + 4/14\,(0.811) \\
&= 0.285714 + 0.393428 + 0.231714 = 0.9108
\end{aligned}
$$

$Gain(income) = 0.94 - 0.9108 = 0.0292$

- For Student we have two values $student_{yes}$ (6 yes and 1 no) and $student_{no}$ (3 yes 4 no)

$$
\begin{aligned}
Entropy(student) &= 7/14(-6/7 \log(6/7)) + 7/14(-3/7 \log(3/7) - 4/7 \log(4/7)) \\
&= 7/14(0.5916) + 7/14(0.9852) \\
&= 0.2958 + 0.4926 = 0.7884
\end{aligned}
$$

$Gain\ (student) = 0.94 - 0.7884 = 0.1516$

- For Credit_Rating we have two values $credit\_rating_{fair}$ (6 yes and 2 no) and $credit\_rating_{excellent}$ (3 yes 3 no)

$$
\begin{aligned}
Entropy(credit\_rating) &= 8/14(-6/8 \log(6/8) - 2/8 \log(2/8)) + 6/14(-3/6 \log(3/6) - 3/6 \log(3/6)) \\
&= 8/14(0.8112) + 6/14(1) \\
&= 0.4635 + 0.4285 = 0.8920
\end{aligned}
$$

$Gain(credit\_rating) = 0.94 - 0.8920 = 0.479$

Since Age has the highest Information Gain we start splitting the dataset using the age attribute

age

<=30

Income	student	credit_rating	Class
high	no	fair	No
high	no	excellent	No
medium	no	fair	No
low	yes	fair	Yes
medium	yes	excellent	Yes

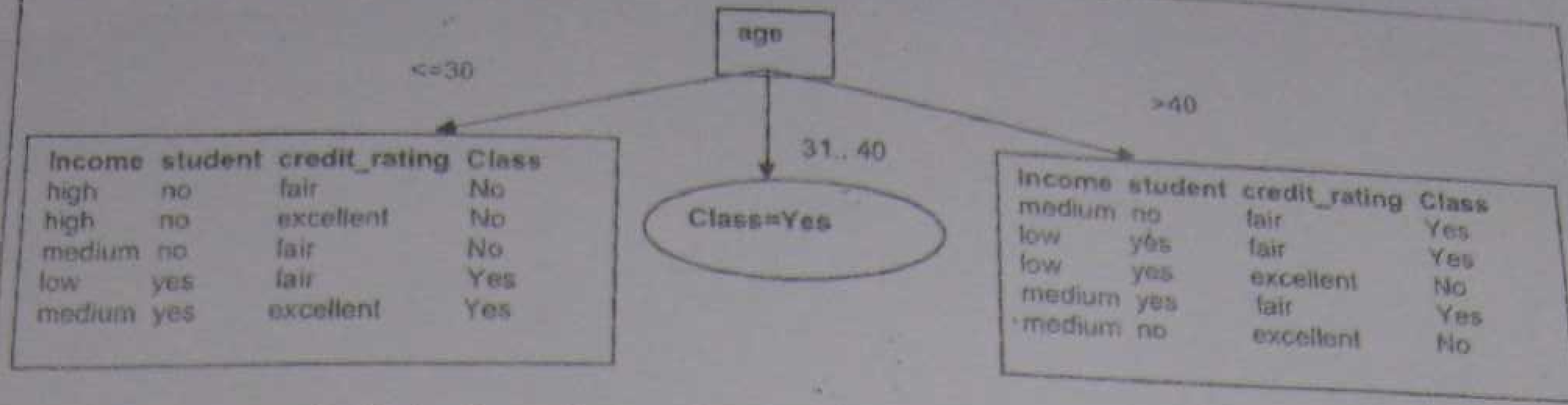
31..40

Income	student	credit_rating	Class
high	no	fair	Yes
low	yes	excellent	Yes
medium	no	excellent	Yes
high	yes	fair	Yes

>40

Income	student	credit_rating	Class
medium	no	fair	Yes
low	yes	fair	Yes
low	yes	excellent	No
medium	yes	fair	Yes
medium	no	excellent	No

Since all records under the branch $age_{31..40}$ are all of class Yes, we can replace the leaf with Class=Yes



The same process of splitting has to happen for the two remaining branches.
For branch $age_{<=30}$ we still have attributes income, student and credit_rating. Which one should be use to split the partition?

The mutual information is $I(S_{Yes}, S_{No}) = I(2,3) = -2/5 \log_2(2/5) - 3/5 \log_2(3/5) = 0.97$

- For Income we have three values $income_{high}$ (0 yes and 2 no), $income_{medium}$ (1 yes and 1 no) and $income_{low}$ (1 yes and 0 no)

$$\text{Entropy(income)} = 2/5(0) + 2/5\,(-1/2\log(1/2) - 1/2\log(1/2)) + 1/5\,(0)$$
$$= 2/5\,(1) = 0.4$$

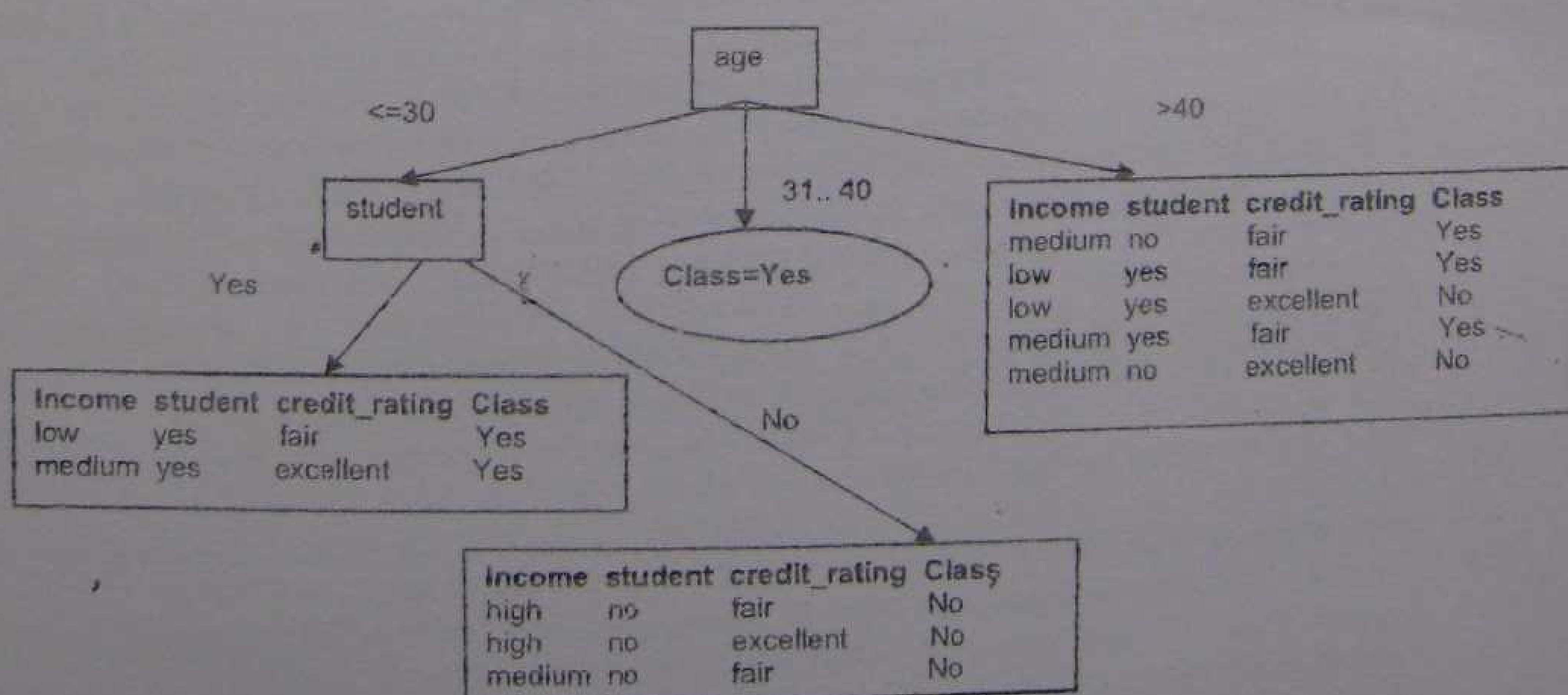
Gain(income) = 0.97 – 0.4 = 0.57

- For Student we have two values $student_{yes}$ (2 yes and 0 no) and $student_{no}$ (0 yes 3 no)

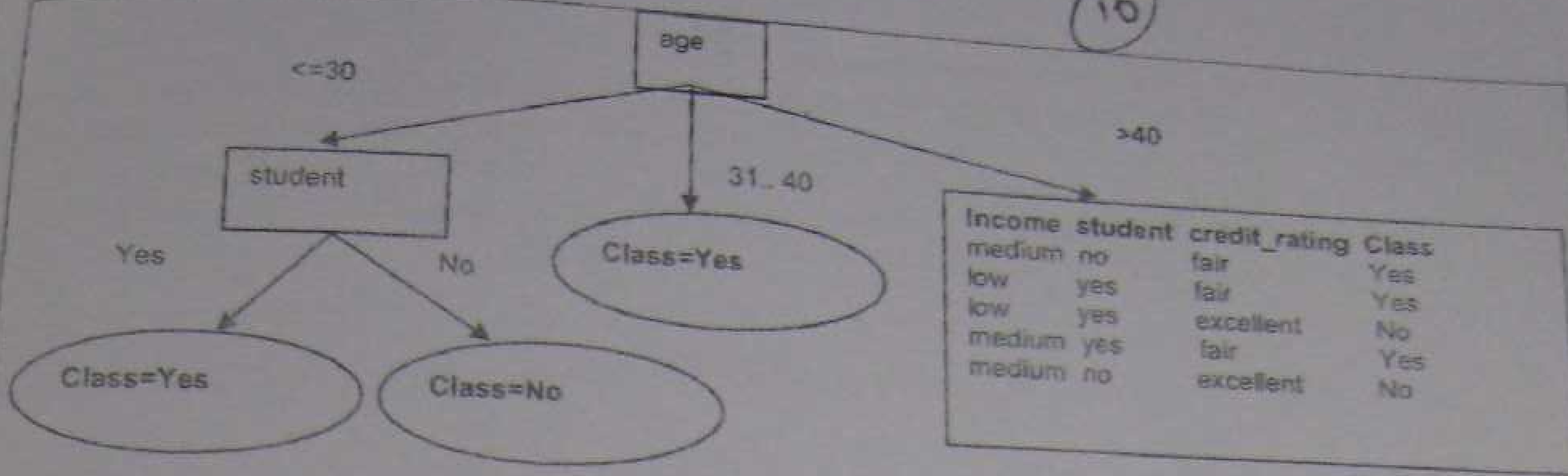
Entropy(student) = 2/5(0) + 3/5(0) = 0

Gain (student) = 0.97 – 0 = 0.97

We can then safely split on attribute student without checking the other attributes since the information gain is maximized.



Since these two new branches are from distinct classes, we make them into leaf nodes with their respective class as label:



Again the same process is needed for the other branch of age.

The mutual information is $I(S_{Yes}, S_{No}) = I(3,2) = -3/5 \log_2(3/5) - 2/5 \log_2(2/5) = 0.97$

- For Income we have two values $income_{medium}$ (2 yes and 1 no) and $income_{low}$ (1 yes and 1 no)

$$Entropy(income) = 3/5(-2/3\log(2/3)-1/3\log(1/3)) + 2/5(-1/2\log(1/2)-1/2\log(1/2))$$
$$= 3/5(0.9182)+2/5(1) = 0.55+0.4 = 0.95$$

Gain(income) = 0.97 – 0.95 = 0.02

- For Student we have two values $student_{yes}$ (2 yes and 1 no) and $student_{no}$ (1 yes and 1 no)

$$Entropy(student) = 3/5(-2/3\log(2/3)-1/3\log(1/3)) + 2/5(-1/2\log(1/2)-1/2\log(1/2)) = 0.95$$

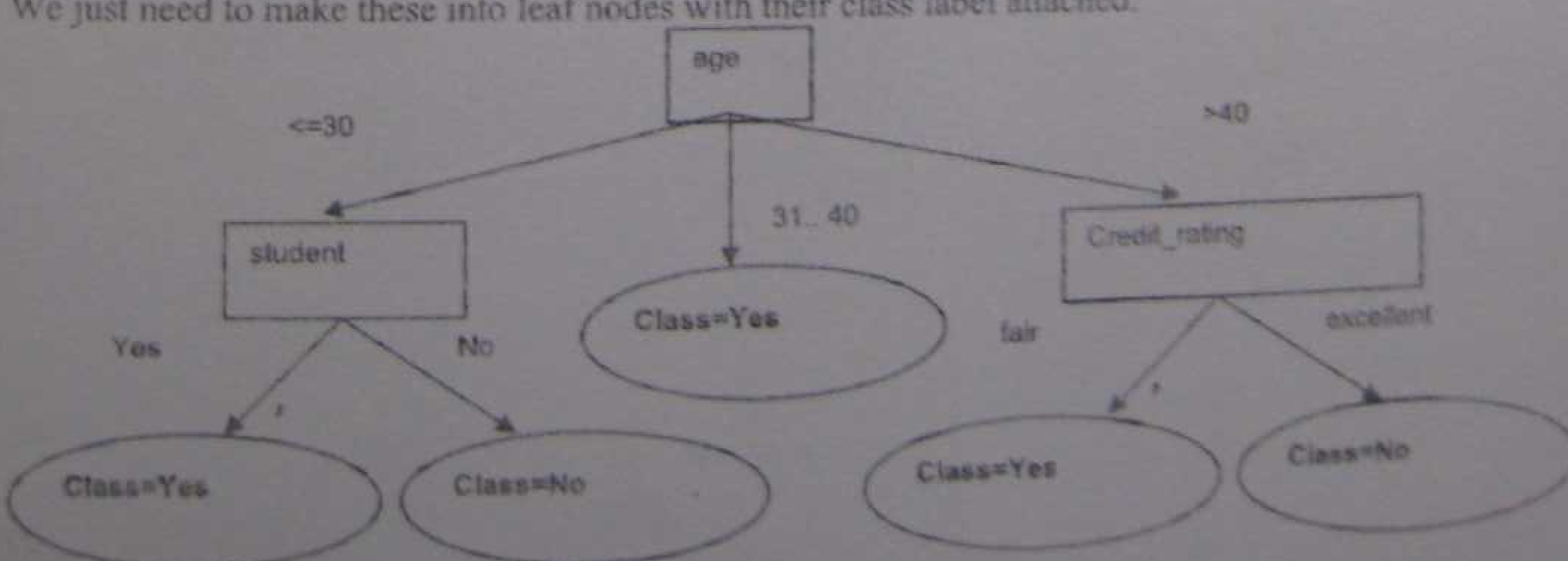
Gain (student) = 0.97 – 0.95 = 0.02

- For Credit_Rating we have two values $credit\_rating_{fair}$ (3 yes and 0 no) and $credit\_rating_{excellent}$ (0 yes and 2 no)

Entropy(credit_rating) = 0

Gain(credit_rating) = 0.97 – 0 = 0.97

We then split based on credit_rating. These splits give partitions each with records from the same class. We just need to make these into leaf nodes with their class label attached:



New example: age<=30, income=medium, student=yes, credit-rating=fair
Follow branch(age<=30) then student=yes we predict Class=yes → Buys_computer = yes

Tutorial exercises

Clustering – K-means, Nearest Neighbor and Hierarchical.

1

Exercise 1. K-means clustering

Use the k-means algorithm and Euclidean distance to cluster the following 8 examples into 3 clusters:
A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).
The distance matrix based on the Euclidean distance is given below:

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	$\sqrt{25}$	$\sqrt{36}$	$\sqrt{13}$	$\sqrt{50}$	$\sqrt{52}$	$\sqrt{65}$	$\sqrt{5}$
A2		0	$\sqrt{37}$	$\sqrt{18}$	$\sqrt{25}$	$\sqrt{17}$	$\sqrt{10}$	$\sqrt{20}$
A3			0	$\sqrt{25}$	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{53}$	$\sqrt{41}$
A4				0	$\sqrt{13}$	$\sqrt{17}$	$\sqrt{52}$	$\sqrt{2}$
A5					0	$\sqrt{2}$	$\sqrt{45}$	$\sqrt{25}$
A6						0	$\sqrt{29}$	$\sqrt{29}$
A7							0	$\sqrt{58}$
A8								0

Suppose that the initial seeds (centers of each cluster) are A1, A4 and A7. Run the k-means algorithm for 1 epoch only. At the end of this epoch show:

a) The new clusters (i.e. the examples belonging to each cluster)

b) The centers of the new clusters

c) Draw a 10 by 10 space with all the 8 points and show the clusters after the first epoch and the new centroids.

d) How many more iterations are needed to converge? Draw the result for each epoch.

Solution:

a)

d(a,b) denotes the Eucledian distance between a and b. It is obtained directly from the distance matrix or calculated as follows: $d(a,b)=sqrt((x_b-x_a)^2+(y_b-y_a)^2))$

seed1=A1=(2,10), seed2=A4=(5,8), seed3=A7=(1,2)

epoch1 – start:

A1:
d(A1, seed1)=0 as A1 is seed1
d(A1, seed2)= $\sqrt{13}$ >0
d(A1, seed3)= $\sqrt{65}$ >0
→ A1 ∈ cluster1

A2:
d(A2,seed1)= $\sqrt{25}$ = 5.
d(A2, seed2)= $\sqrt{18}$ = 4.24
d(A2, seed3)= $\sqrt{10}$ = 3.16 ← smaller
→ A2 ∈ cluster3

A3:
d(A3, seed1)= $\sqrt{36}$ = 6
d(A3, seed2)= $\sqrt{25}$ = 5 ← smaller
d(A3, seed3)= $\sqrt{53}$ = 7.28
→ A3 ∈ cluster2

A4:
d(A4, seed1)= $\sqrt{13}$
d(A4, seed2)=0 as A4 is seed2
d(A4, seed3)= $\sqrt{52}$ >0
→ A4 ∈ cluster2

A5:
d(A5, seed1)= $\sqrt{50}$ = 7.07

A6:
d(A6, seed1)= $\sqrt{52}$ = 7.21

d(A5, seed2)= $\sqrt{13}$ = 3.60 ← smaller
d(A5, seed3)= $\sqrt{45}$ = 6.70
→ A5 ∈ cluster2

d(A6, seed2)= $\sqrt{17}$ = 4.12 ← smaller
d(A6, seed3)= $\sqrt{29}$ = 5.38
→ A6 ∈ cluster2

A7:
d(A7, seed1)= $\sqrt{65}$ >0
d(A7, seed2)= $\sqrt{52}$ >0
d(A7, seed3)=0 as A7 is seed3
→ A7 ∈ cluster3

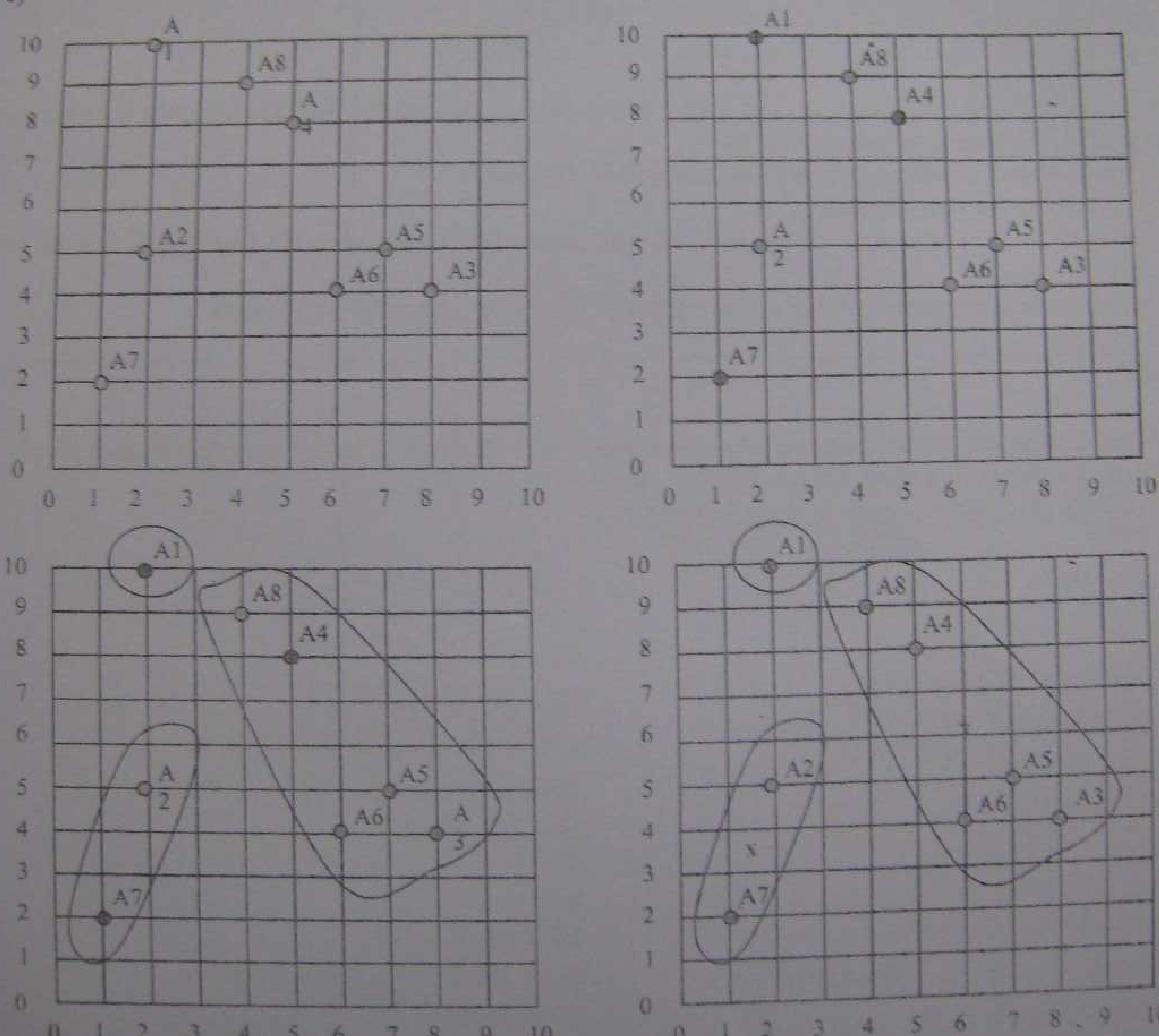
A8:
d(A8, seed1)= $\sqrt{5}$
d(A8, seed2)= $\sqrt{2}$ ← smaller
d(A8, seed3)= $\sqrt{58}$
→ A8 ∈ cluster2

end of epoch1

new clusters: 1: {A1}, 2: {A3, A4, A5, A6, A8}, 3: {A2, A7}

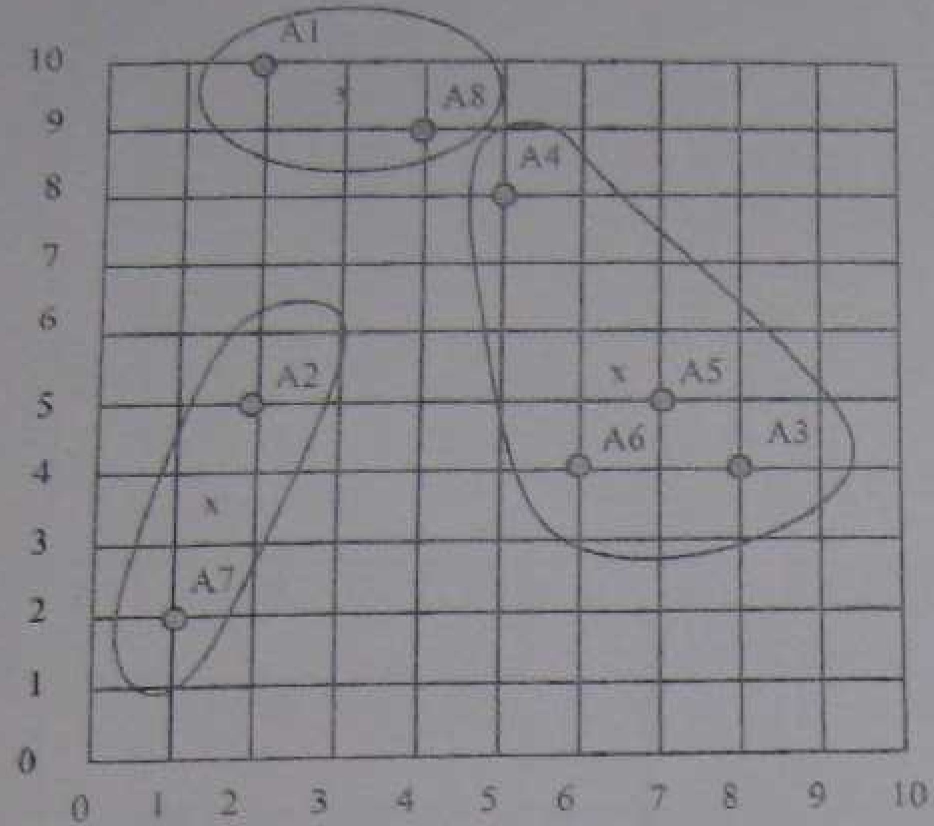
b) centers of the new clusters:
C1=(2, 10), C2= ((8+5+7+6+4)/5, (4+8+5+4+9)/5) = (6, 6), C3= ((2+1)/2, (5+2)/2) = (1.5, 3.5)

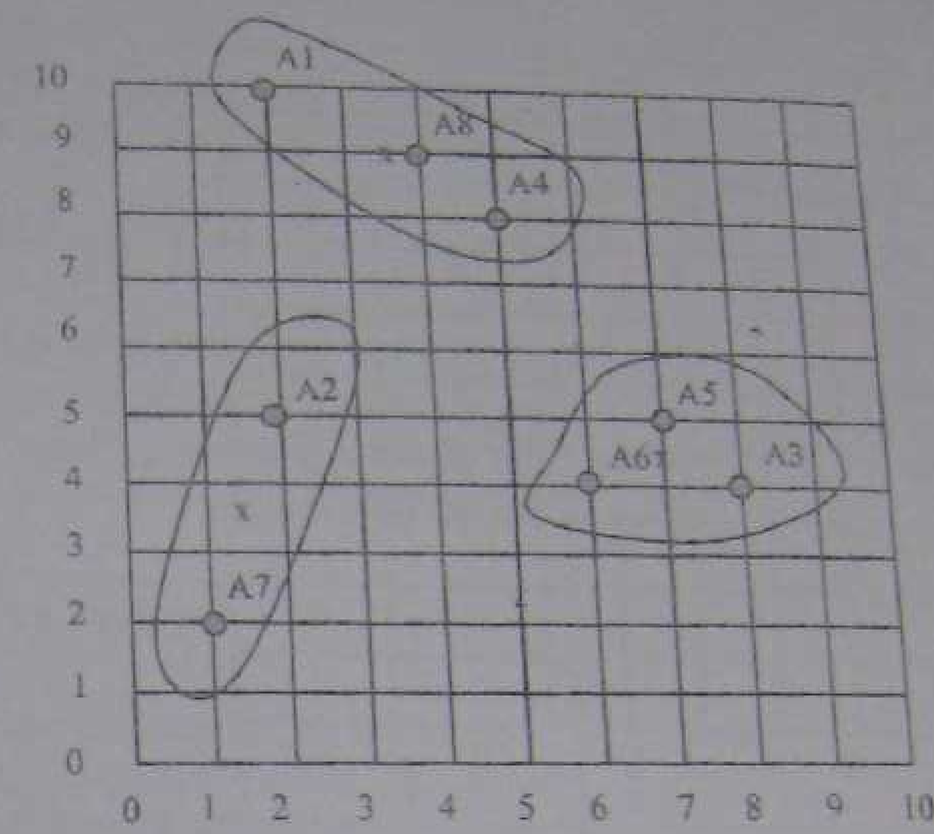
c)



We would need two more epochs. After the 2nd epoch the results would be:
1: {A1, A8}, 2: {A3, A4, A5, A6}, 3: {A2, A7}
with centers C1=(3, 9.5), C2=(6.5, 5.25) and C3=(1.5, 3.5).
After the 3rd epoch, the results would be:
1: {A1, A4, A8}, 2: {A3, A5, A6}, 3: {A2, A7}
with centers C1=(3.66, 9), C2=(7, 4.33) and C3=(1.5, 3.5).

3





Exercise 2. Nearest Neighbor clustering

Use the Nearest Neighbor clustering algorithm and Euclidean distance to cluster the examples from the previous exercise: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9). Suppose that the threshold t is 4.

Solution:
A1 is placed in a cluster by itself, so we have K1={A1}.

We then look at A2 if it should be added to K1 or be placed in a new cluster.
d(A1,A2)= $\sqrt{25}$ = 5 > t → K2={A2}

A3: we compare the distances from A3 to A1 and A2.
A3 is closer to A2 and d(A3,A2)=$\sqrt{36}$ > t → K3={A3}

A4: We compare the distances from A4 to A1, A2 and A3.
A1 is the closest object and d(A4,A1)= $\sqrt{13}$ < t → K1={A1, A4}

A5: We compare the distances from A5 to A1, A2, A3 and A4.
A3 is the closest object and d(A5,A3)= $\sqrt{2}$ < t → K3={A3, A5}

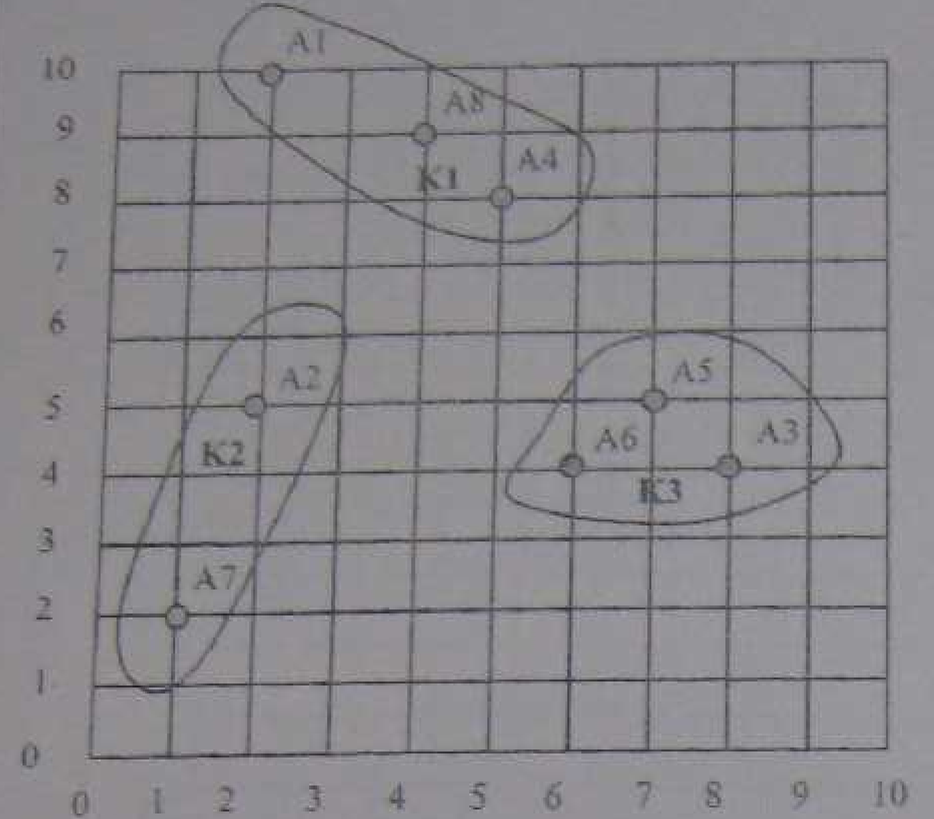
A6: We compare the distances from A6 to A1, A2, A3, A4 and A5.
A3 is the closest object and d(A6,A3)= $\sqrt{2}$ < t → K3={A3, A5, A6}

A7: We compare the distances from A7 to A1, A2, A3, A4, A5, and A6.
A2 is the closest object and d(A7,A2)= $\sqrt{10}$ < t → K2={A2, A7}

A8: We compare the distances from A8 to A1, A2, A3, A4, A5, A6 and A7.
A4 is the closest object and d(A8,A4)= $\sqrt{2}$ < t → K1={A1, A4, A8)

Thus: K1={A1, A4, A8), K2={A2, A7), K3={A3, A5, A6)

Yes, it is the same result as with K-means.



4

Exercise 3. Hierarchical clustering

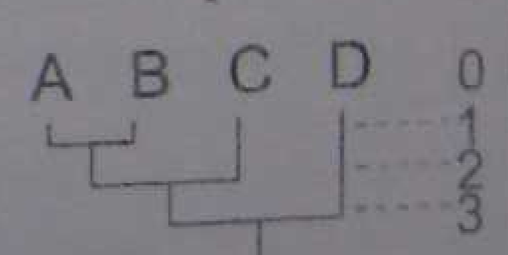
Use single and complete link agglomerative clustering to group the data described by the following distance matrix. Show the dendrograms.

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Solution:
Agglomerative → initially every point is a cluster of its own and we merge cluster until we end-up with one unique cluster containing all points.

a) single link: distance between two clusters is the shortest distance between a pair of elements from the two clusters.

d	k	K	Comments
0	4	{A}, {B}, {C}, {D}	We start with each point = cluster
1	3	{A, B}, {C}, {D}	Merge {A} and {B} since A & B are the closest: d(A, B)=1
2	2	{A, B, C}, {D}	Merge {A, B} and {C} since B & C are the closest: d(B, C)=2
3	1	{A, B, C, D}	Merge D

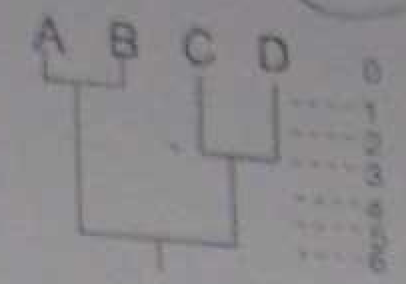


b) complete link: distance between two clusters is the longest distance between a pair of elements from

5

the two clusters.

d	k	K	Comments
0	4	{A}, {B}, {C}, {D}	We start with each point = cluster
1	3	{A, B}, {C}, {D}	d(A,B)=1<=1 → merge {A} and {B}
2	3	{A, B}, {C}, {D}	d(A,C)=4>2 so we can't merge C with {A,B} d(A,D)=5>2 and d(B,D)=6>2 so we can't merge D with {A, B} d(C,D)=3>2 so we can't merge C and D
3	2	{A, B}, {C, D}	- d(A,C)=4>3 so we can't merge C with {A,B} - d(A,D)=5>3 and d(B,D)=6>3 so we can't merge D with {A, B} - d(C,D)=3 <=3 so merge C and D
4	2	{A, B}, {C, D}	{C,D} cannot be merged with {A, B} as d(A,D)= 5 >4 (and also d(B,D)= 6 >4) although d(A,C)= 4 <= 4, d(B,C)= 2<=4)
5	2	{A, B}, {C, D}	{C,D} cannot be merged with {A, B} as d(B,D)= 6 > 5
6	1	{A, B, C, D}	{C, D} can be merged with {A, B} since d(B,D)= 6 <= 6, d(A,D)= 5 <= 6, d(A,C)= 4 <= 6, d(B,C)= 2 <= 6



Exercise 4: **Hierarchical clustering (to be done at your own time, not in class)**

Use single-link, complete-link, average-link agglomerative clustering as well as medoid and centroid to cluster the following 8 examples:

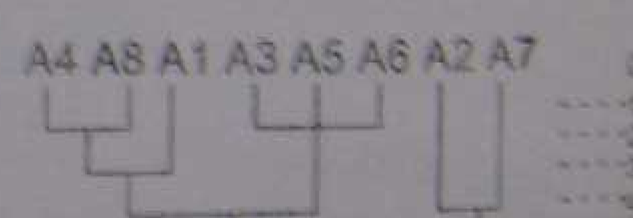
A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9).

The distance matrix is the same as the one in Exercise 1. Show the dendrograms.

Solution:

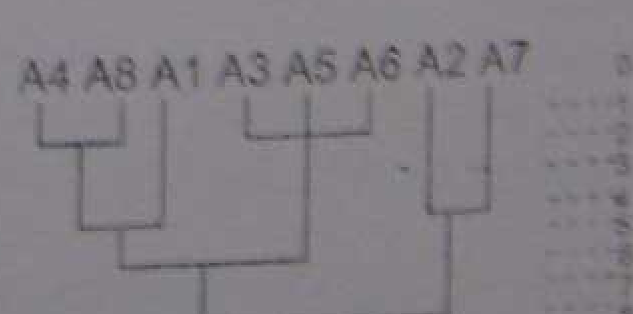
Single Link:

d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	4	{A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}
4	2	{A1, A3, A4, A5, A6, A8}, {A2, A7}
5	1	{A1, A3, A4, A5, A6, A8, A2, A7}



Complete Link

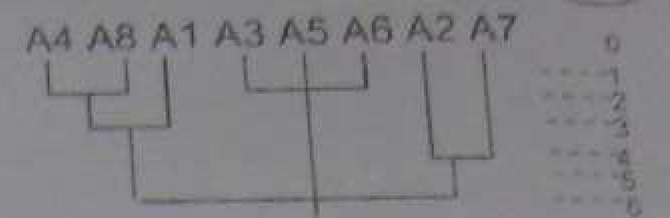
d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
4	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
5	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
6	2	{A4, A8, A1, A3, A5, A6}, {A2, A7}
7	2	{A4, A8, A1, A3, A5, A6}, {A2, A7}
8	1	{A4, A8, A1, A3, A5, A6, A2, A7}



6

Average Link

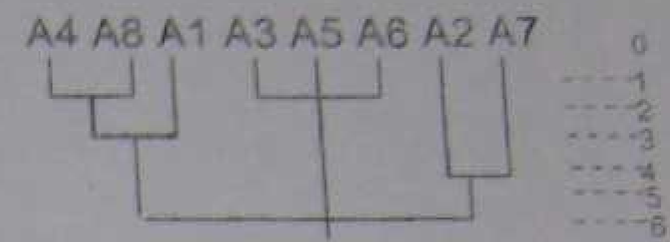
d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	4	{A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}
4	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
5	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
6	1	{A4, A8, A1, A3, A5, A6, A2, A7}



Average distance from {A3, A5, A6} to {A1, A4, A8} is 5.53 and is 5.75 to {A2, A7}

Centroid

D	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
4	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
5	3	{A4, A8, A1}, {A3, A5, A6}, {A2, A7}
6	1	{A4, A8, A1, A3, A5, A6, A2, A7}

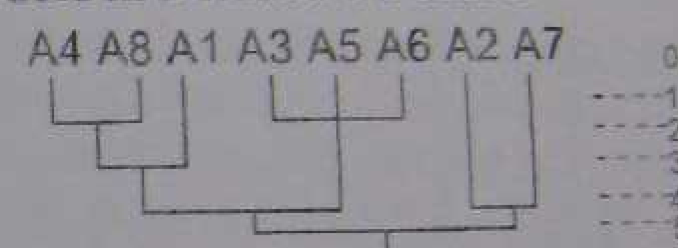


Centroid of {A4, A8} is B=(4.5, 8.5) and centroid of {A3, A5, A6} is C=(7, 4.33)
distance(A1, B) = 2.91 Centroid of {A1, A4, A8} is D=(3.66, 9) and of {A2, A7} is E=(1.5, 3.5)
distance(D,C)= 5.74 distance(D,E)= 5.90

Medoid

This is not deterministic. It can be different depending upon which medoid in a cluster we chose.

d	k	K
0	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
1	8	{A1}, {A2}, {A3}, {A4}, {A5}, {A6}, {A7}, {A8}
2	5	{A4, A8}, {A1}, {A3, A5, A6}, {A2}, {A7}
3	4	{A4, A8, A1}, {A3, A5, A6}, {A2}, {A7}
4	2	{A1, A3, A4, A5, A6, A8}, {A2, A7}
5	1	{A1, A3, A4, A5, A6, A8, A2, A7}



Exercise 5: DBScan

If Epsilon is 2 and minpoint is 2, what are the clusters that DBScan would discover with the following 8 examples: A1=(2,10), A2=(2,5), A3=(8,4), A4=(5,8), A5=(7,5), A6=(6,4), A7=(1,2), A8=(4,9). The distance matrix is the same as the one in Exercise 1. Draw the 10 by 10 space and illustrate the discovered clusters. What if Epsilon is increased to $\sqrt{10}$?

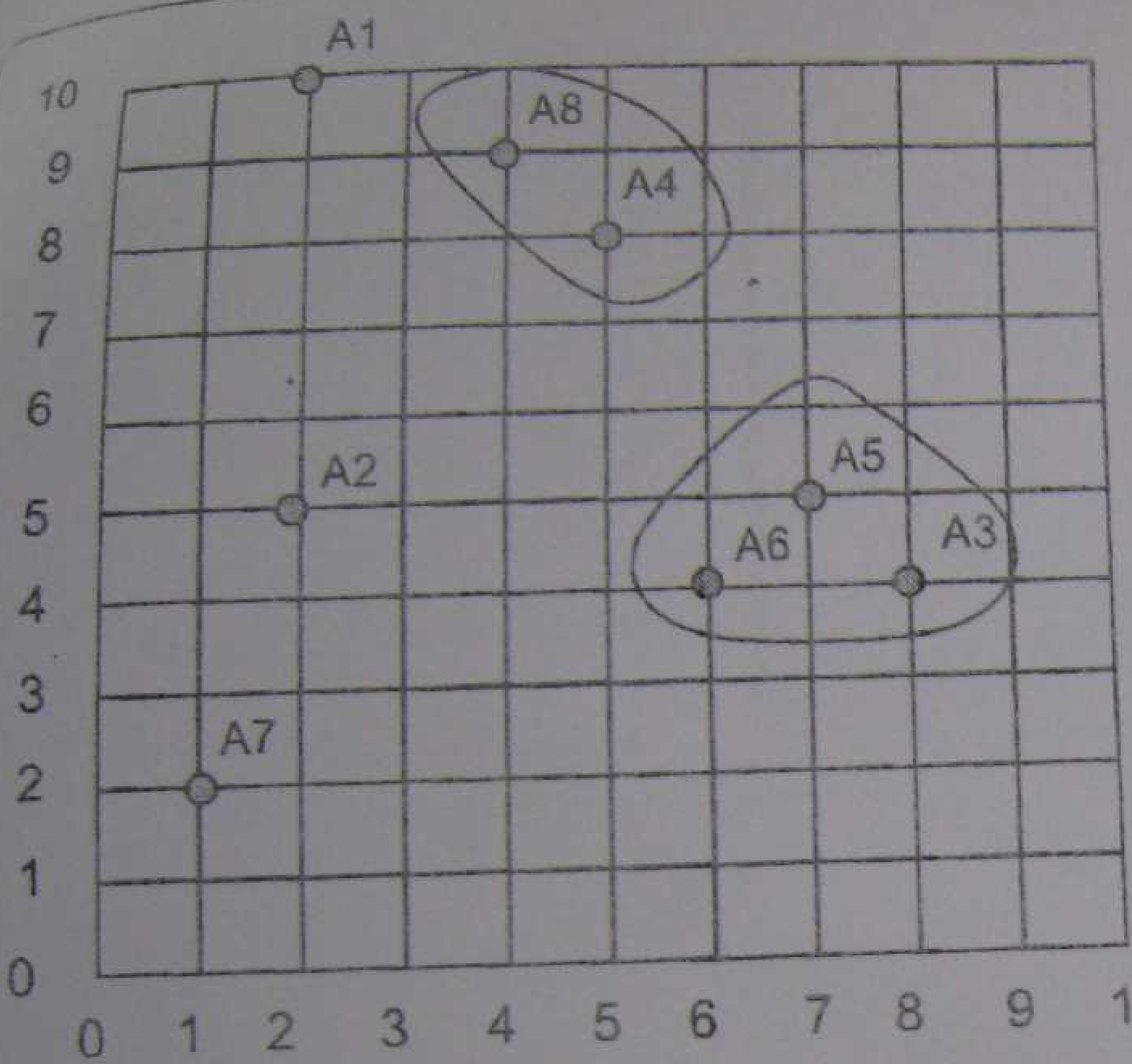
Solution:

What is the Epsilon neighborhood of each point?

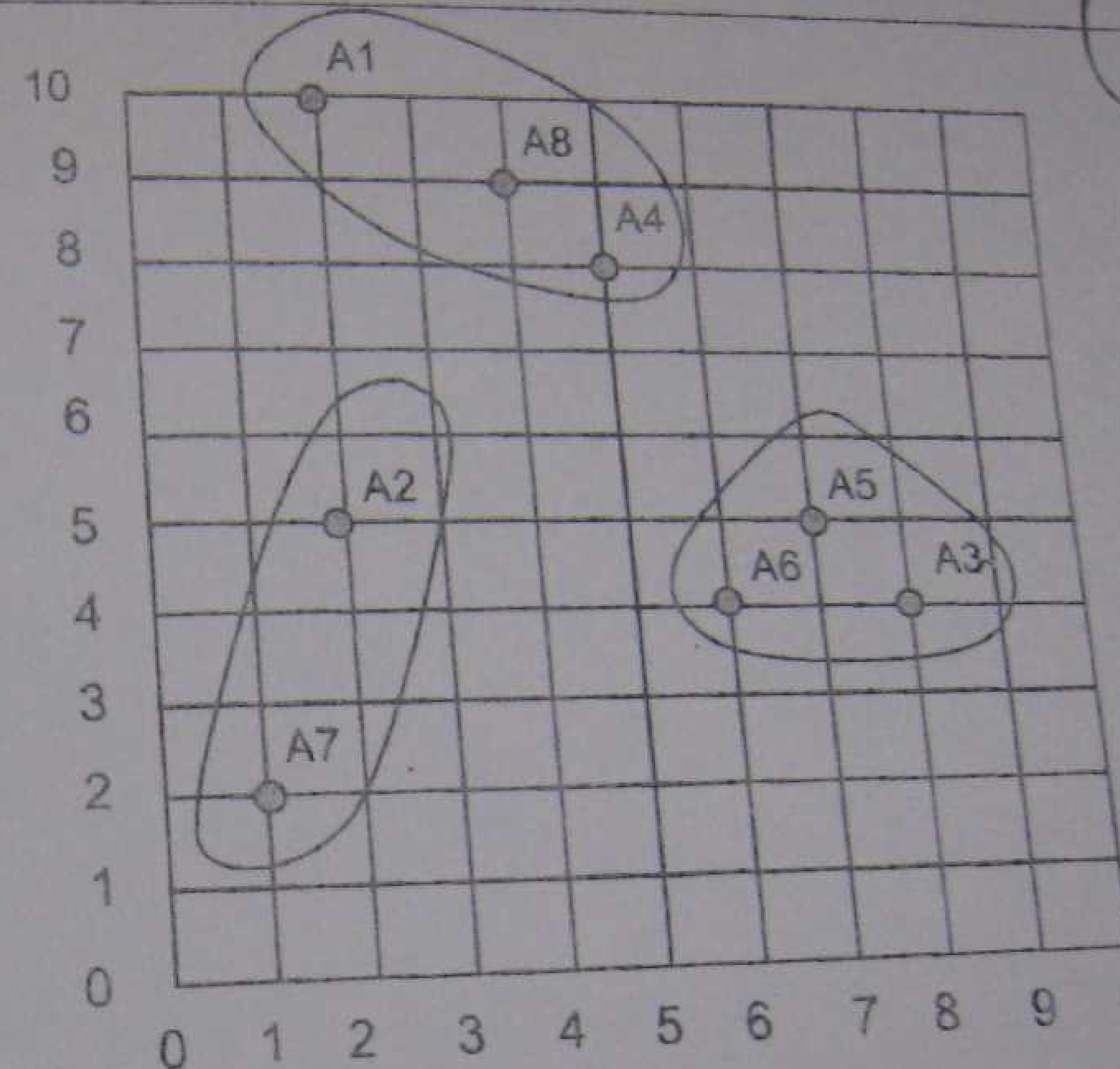
N_2(A1)={}; N_2(A2)={}; N_2(A3)={A5, A6}; N_2(A4)={A8}; N_2(A5)={A3, A6};
N_2(A6)={A3, A5}; N_2(A7)={}; N_2(A8)={A4}

So A1, A2, and A7 are outliers, while we have two clusters C1={A4, A8} and C2={A3, A5, A6}

If Epsilon is $\sqrt{10}$ then the neighborhood of some points will increase:
A1 would join the cluster C1 and A2 would joint with A7 to form cluster C3={A2, A7}.



Epsilon = 2



Epsilon = $\sqrt{10}$