German University in Cairo
Faculty of Media Engineering and Technology
Spring 2019

CSEN1083 – Data Mining

Problem Set #5

Problem 1

Consider the data given below that describes 8 students studying for an exam. Each student has 3 attributes, and each is either focused on their studies or not. The IsFocused column represents the label of the data. Find the corresponding decision tree.

#	AmountOfSleep	IsInterested	HasBreakPlans	IsFocused
1	Not Enough	Yes	No	Yes
2	Too Much	Yes	No	Yes
3	None	Yes	No	No
4	Not Enough	Yes	No	Yes
5	None	No	Yes	No
6	Too Much	No	Yes	No
7	Not Enough	No	No	No
8	Not Enough	Yes	No	Yes

Solution

We start by computing the Gain for each attribute to determine the root node in the decision tree.

Gain(S, AmountOfSleep)

= Entropy(S) – (4/8) Entropy(AmountOfSleep = Not Enough) – (2/8) Entropy(AmountOfSleep = Too Much) - (2/8) Entropy(AmountOfSleep =None)

Entropy(S) = $-(4/8)\log_2(4/8) -(4/8)\log_2(4/8) = 1$

Entropy(AmountOfSleep = Not Enough) = $-(3/4)\log_2(3/4) -(1/4)\log_2(1/4) = 0.8113$

Entropy(AmountOfSleep = Too Much) = 1

Entropy(AmountOfSleep = None) = 0

Gain(S, AmountOfSleep) = 1 – (4/8) x 0.8113 – (2/8) x 1 – 0 = 0.3443

Gain(S, IsInterested)

= Entropy(S) – (5/8) Entropy(IsInterested = Yes) – (3/8) Entropy(IsInterested = No)

Entropy(IsInterested = Yes) = $-(4/5)\log_2(4/5) -(1/5)\log_2(1/5) = 0.7219$

Entropy(IsInterested = No) = 0

CSEN1083 – Data Mining

Problem Set #5

Gain(S, IsInterested) = 1 – (5/8) x 0.7219 – 0 = 0.5488

Gain(S, HasBreakPlans)

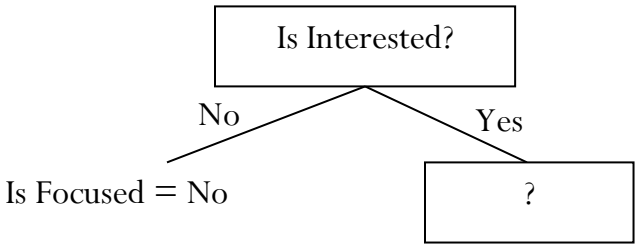
= Entropy(S) – (2/8) Entropy(HasBreakPlans = Yes) – (6/8) Entropy(HasBreakPlans = No)

Entropy(HasBreakPlans = Yes) = 0

Entropy(HasBreakPlans = No) = $-(4/6)\log_2(4/6) -(2/6)\log_2(2/6) = 0.9183$

Gain(S, HasBreakPlans) = 1 – (6/8) x 0.9183 = 0.2640

Since, Gain(S, IsInterested) is the largest, then the root node is IsInterested



We then compute the Gain of possible attributes for the dataset of IsInterested = Yes.
Gain(IsInterested = Yes, AmountOfSleep)

= Entropy(IsInterested = Yes) – (3/5) Entropy(IsInterested = Yes , AmountOfSleep = Not Enough) – (1/5) Entropy(IsInterested = Yes , AmountOfSleep = Too Much) - (1/5) Entropy(IsInterested = Yes , AmountOfSleep =None)

Entropy(IsInterested = Yes , AmountOfSleep = Not Enough) = 0

Entropy(IsInterested = Yes , AmountOfSleep = Too much) = 0

Entropy(IsInterested = Yes , AmountOfSleep = None) = 0

Gain(IsInterested = Yes, AmountOfSleep) = 0.7219 – 0 – 0 – 0 = 0.7219

Gain(IsInterested = Yes, HasBreakPlans)

CSEN1083 – Data Mining

Problem Set #5

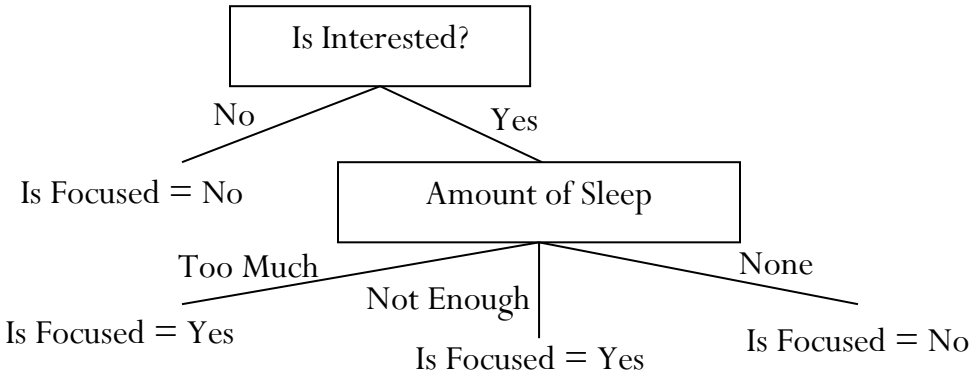
= Entropy(IsInterested = Yes) – (0/5) Entropy(IsInterested = Yes , HasBreakPlans = Yes) – (5/5) Entropy(IsInterested = Yes , HasBreakPlans = No)

Entropy(IsInterested = Yes , HasBreakPlans = No) = $-(4/5)\log_2(4/5) -(1/5)\log_2(1/5) = 0.7219$

Entropy(IsInterested = Yes , HasBreakPlans = Yes) = 0

Gain(IsInterested = Yes, HasBreakPlans) = 0.7219 – 0.7219 = 0

Since, **Gain(IsInterested = Yes, AmountOfSleep)** is the largest, then we use AmountOfSleep

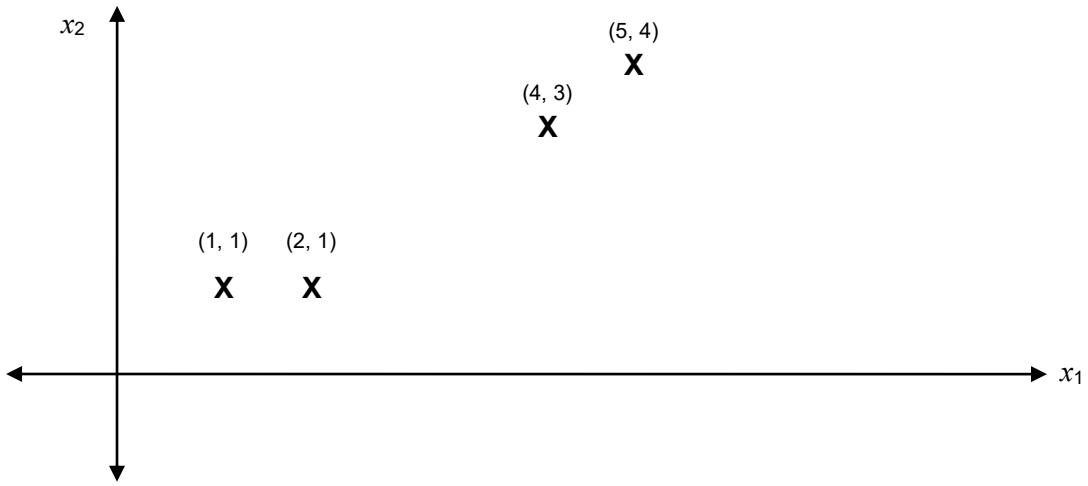


CSEN1083 – Data Mining

Problem Set #5

Problem 2

Consider the data given below. Apply the K-means clustering algorithm using K = 2. Show the result of each iteration till convergence. Use initial centers at $\boldsymbol{\mu}_1 = (1, 1)$ and $\boldsymbol{\mu}_2 = (2, 1)$.



Solution

Iteration 1:

Based on the choice of centers, we estimate r_{nk} for each point based on the distance to the centers:

Point 1: (1, 1) → $r_{11} = 1, r_{12} = 0$

Point 2: (2, 1) → $r_{21} = 0, r_{22} = 1$

Point 3: (4, 3) → $r_{31} = 0, r_{32} = 1$

Point 4: (5, 4) → $r_{41} = 0, r_{42} = 1$

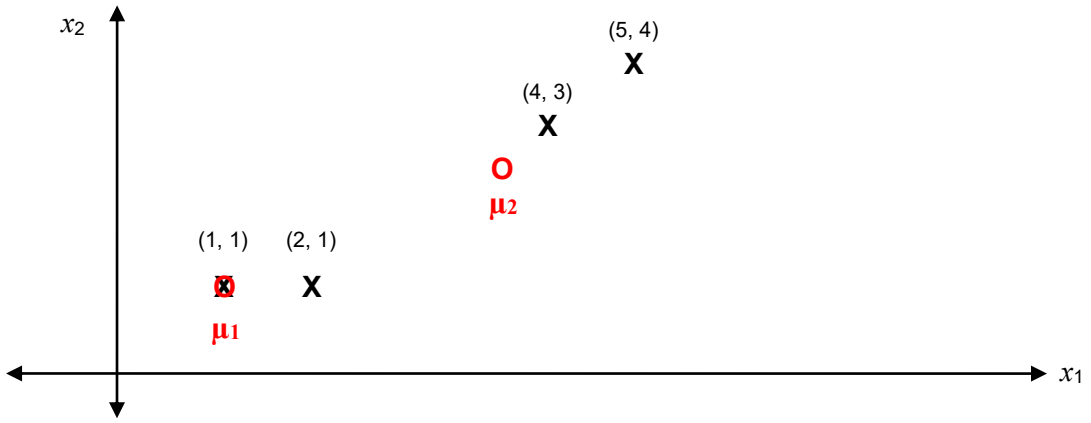
We then re-estimate the centers:

$\boldsymbol{\mu}_1 = (1, 1)$

$\boldsymbol{\mu}_2 = 1/3\ ((2, 1) + (4, 3) + (5, 4)) = (3.67, 2.67)$

CSEN1083 – Data Mining

Problem Set #5



Iteration 2:

Based on the new value of the centers, we estimate r_{nk} for each point based on the distance to the centers:

Point 1: (1, 1) → $r_{11} = 1, r_{12} = 0$

Point 2: (2, 1) → $r_{21} = 1, r_{22} = 0$

Point 3: (4, 3) → $r_{31} = 0, r_{32} = 1$

Point 4: (5, 4) → $r_{41} = 0, r_{42} = 1$

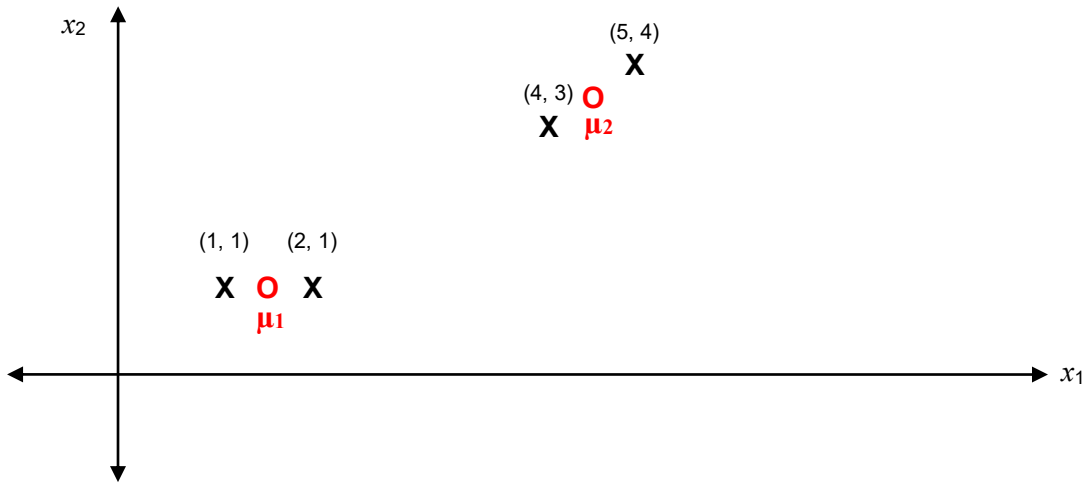
We then re-estimate the centers:

$\boldsymbol{\mu}_1 = 1/2\ ((1, 1) + (2, 1)) = (1.5, 1)$

$\boldsymbol{\mu}_2 = 1/2\ ((4, 3) + (5, 4)) = (4.5, 3.5)$

CSEN1083 – Data Mining

Problem Set #5

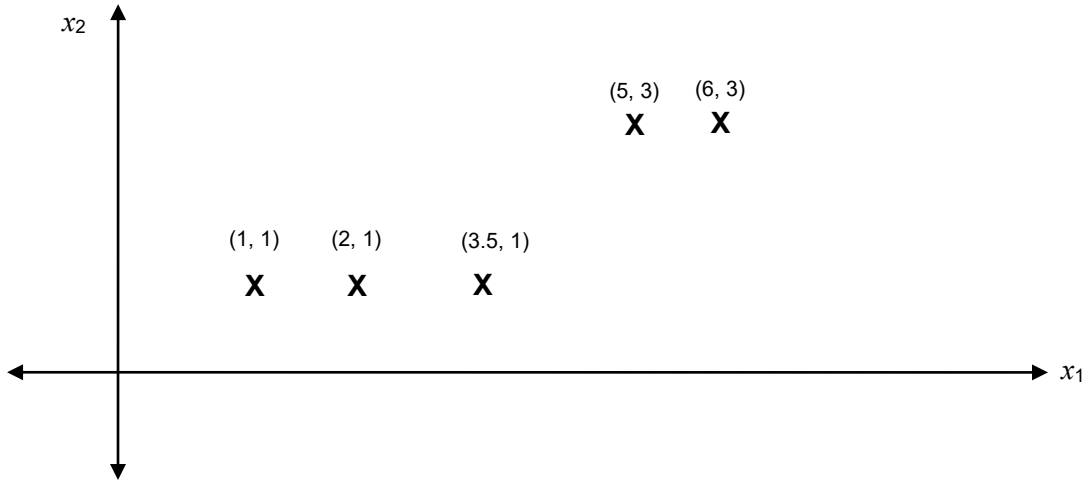


CSEN1083 – Data Mining

Problem Set #5

Problem 3

Consider the data given below. Apply hierarchical clustering using single linkage and using complete linkage. Show your computations and the resulting dendrogram.



Solution

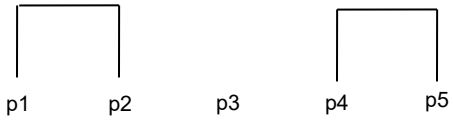
Let the dataset points be denoted by: p1 = (1,1), p2 = (2,1), p3 = (3.5,1), p4 = (4,3) and p5 = (5,4).

We start by forming a distance matrix shown the distance between every pair of points as shown below

	p1	p2	p3	p4	p5
p1	0	1	2.5	4.47	5.83
p2	1	0	1.5	3.6	4.47
p3	2.5	1.5	0	2.5	3.2
p4	4.47	3.6	2.5	0	1
p5	5.83	4.47	3.2	1	0

We find the minimum (non-zero) distance in the distance matrix, and merge the corresponding two points. In this case, the minimum value is 1, so we merge p1 with p2 in one cluster, and p4 and p5 in one cluster. Note that for this first step, it doesn't matter if we are using single or complete linkage since each cluster is one point only.

The resulting dendrogram after the first step is:



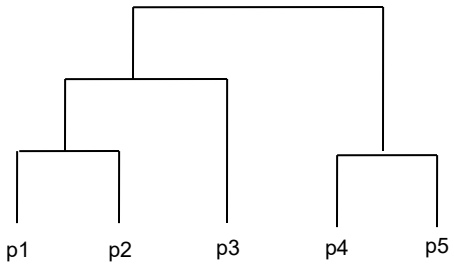
CSEN1083 – Data Mining

Problem Set #5

We next re-form the distance matrix based on the linkage method used. For single linkage, we use the following distance measure: $D(C_i, C_j) = \min_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$

	(p1, p2)	p3	(p4, p5)
(p1, p2)	0	1.5	3.6
p3	1.5	0	2.5
(p4, p5)	3.6	2.5	0

Since the smallest distance is 1.5 corresponding to p3 and (p1, p2), we merge p3 with cluster (p1, p2). The resulting dendrogram is given by



If we use complete linkage, we use the following distance measure: $D(C_i, C_j) = \max_{\mathbf{x}_i \in C_i, \mathbf{x}_j \in C_j} d(\mathbf{x}_i, \mathbf{x}_j)$. The distance matrix is then given by

	(p1, p2)	p3	(p4, p5)
(p1, p2)	0	2.5	5.83
p3	2.5	0	3.2
(p4, p5)	5.83	3.2	0

Since the smallest distance is 2.5 corresponding to p3 and (p1, p2), we merge p3 with cluster (p1, p2). The resulting dendrogram is given by