COMP 472: Artificial Intelligence

Naïve Bayes Classification

Solutions

Question 1 Assume that a fancy food-store sells wild hand-picked mushrooms from a local farmer. In the store, the mushrooms are labelled as *gourmet*, *good*, or *at-your-own-risk*. The store always keeps the following inventory: 25% of its mushrooms are labeled *gourmet*, 50% are labeled *good*, and 25% are labeled *at-your-own-risk*. Mushrooms labeled as *gourmet* have a 5% chance of being poisonous, a *good* mushroom has a 15% chance of poisoning someone, and a *at-your-own-risk* mushroom has a 25% chance.

If Jim bought a mushroom from the store and was poisoned,

(a) What is the probability that the mushroom had been labeled *gourmet*?
Given:
$P(gourmet) = 0.25$
$P(good) = 0.5$
$P(ayor) = 0.25$
$P(poisonous|gourmet) = 0.05$
$P(poisonous|good) = 0.15$
$P(poisonous|ayor) = 0.25$

$$\begin{aligned} C_i &\in \{gourmet,\ good,\ at\text{-}your\text{-}own\text{-}risk\} \\ P(poisonous) &= \sum P(poisonous|C_i) * P(C_i) \\ &= 0.05 * 0.25 + 0.15 * 0.5 + 0.25 * 0.25 \\ &= 0.15 \end{aligned}$$

$$\begin{aligned} P(gourmet|poisonous) &= \frac{P(poisonous|gourmet) * P(gourmet)}{P(poisonous)} \\ &= \frac{0.05 * 0.25}{0.15} \\ &= 0.083 \end{aligned}$$

(b) What is the probability that the mushroom had been labeled *good*?

$$\begin{aligned}P(good|poisonous) &= \frac{P(poisonous|good) * P(good)}{P(poisonous)} \\ &= \frac{0.15 * 0.5}{0.15} \\ &= 0.5\end{aligned}$$

(c) What is the probability that the mushroom had been labeled *at-your-own-risk*?

$$\begin{aligned}P(ayor|poisonous) &= \frac{P(poisonous|ayor) * P(ayor)}{P(poisonous)} \\ &= \frac{0.25 * 0.25}{0.15} \\ &= 0.417\end{aligned}$$

Question 2 Assume that Cecilia receives many e-mails from her home town in Klinga, where people speak Klinish. If you do not know Klinish, don't worry. It is a simple language made up of only 1,000 words that all start with the letter "k". A Klinish document may also contain words that do not start with "k", but these are considered out-of-vocabulary words (like a proper name, for example). Jack is trying to help Cecilia sort her Inbox into 3 mail folders (Personal, Work and Promotion). However, Jack does not speak Klinish, so all he has to work from are old e-mails that Cecilia has already sorted into the right folders. The table below shows a sample of the data that Jack has gathered from Cecilia's previous e-mails. The table indicates the frequency of each Klinish word in each folder (to be complete, the table should contain 1,000 rows, corresponding to each word in Klinish). For example, the word kiki appeared 30 times in e-mails labelled Personal, 50 times in e-mails about Work,...

		Folder		
		Personal	Work	Promotion
Word	kami	45	12	17
	kawa	78	1	67
	keke	0	5	80
	kiki	30	50	9
	koko	6	10	10
	kotuku	5	27	20
	koula	17	56	3
	...			
Total Nb of Words		20,000	25,000	17,000

The table above corresponds to data collected from 50 e-mails labeled *Personal*, 65 e-mails labeled *Work* and 45 e-mails labeled *Promotion*.

Based on the data above, Jack is trying to classify the following two e-mails (note that upper and lower cases should not be distinguished).

Email 1:	Koko kami kawa koula keke
Email 2:	Keke kawa, koko Google koula keke!

(a) Use a Naive Bayes classifier without any smoothing, to classify the two e-mails above. Use the sum of logs (base 10), and show the score of each of the 3 classes (Personal, Work and Promotion) and the most likely class.
Solution:
priors:
$P(Personal) = 50 / 50 + 65 + 45$
$P(Work) = 65 / 50 + 65 + 45$

P(Promotion) = 45 /50 + 65 + 45

Email 1: Koko kami kawa koula keke

$score(Personal) =$
$log(P(personal)) + log(P(koko|personal)) + log(P(kami|personal))+$
$log(P(kawa|personal)) + log(P(koula|personal)) + log(P(keke|personal))$
$= log(50/160)+log(6/20,000)+log(45/20,000)+log(78/20,000)+log(17/20,000)+$
$log(0/20,000)$
$= -\infty$

$score(work) = log(P(work)) + log(P(koko|work)) + log(P(kami|work)) +$
$log(P(kawa|work)) + log(P(koula|work)) + log(P(keke|work))$
$= log(65/160)+log(10/25,000)+log(12/25,000)+log(1/25,000)+log(56/25,000)+$
$log(5/25,000)$
$= -17.8546$

$score(promotion) = log(P(promotion)) + log(P(koko|promotion)) +$
$log(P(kami|promotion))+log(P(kawa|promotion))+log(P(koula|promotion))+$
$log(P(keke|promotion))$
$= log(45/160)+log(10/17,000)+log(17/17,000)+log(67/17,000)+log(3/17,000)+$
$log(80/17,000)$
$= -15.2664$

highest score is -15.2664 $\implies$ *the most likely class is promotion*

Email 2: Keke kawa, koko Google koula keke!

notes:

- ignore the word Google.
- keke counts twice

$score(Personal) = log(P(personal))+log(P(keke|personal))+log(P(kawa|personal))+$
$log(P(koko|personal)) + log(P(koula|personal)) + log(P(keke|personal))$
$= log(50/160)+log(0/2,0000)+log(78/2,0000)+log(6/20,,000)+log(17/20,000)+$
$log(0/2,0000)$
$= -\infty$

$score(work) = log(65/160)+log(5/25,000)+log(1/25,000)+log(10/25,000)+$
$log(56/25,000) + log(5/25,000)$
$= -18.2348$

$score(promotion)$ $==$ $log(45/160) + log(80/17,000) + log(67/17,000) +$
$log(10/17,000) + log(3/17,000) + log(80/17,000)$
$= -14.5938$

highest score is -14.5938 $\implies$ *the most likely class is promotion*

(b) Do the same as part A above, but this time use "add 0.5 smoothing" (i.e. instead of adding the value 1 to each word frequency, add $\frac{1}{2}$ to each word frequency). Adjust the smoothing formula accordingly, and show all your work. Again, use the sum of logs (base 10), and show the score of each of the 3 classes and the most likely class.

Solution:

		Folder		
		Personal	Work	Promotion
Word	kami	45.5	12.5	17.5
	kawa	78.5	1.5	67.5
	keke	0.5	5.5	80.5
	kiki	30.5	50.5	9.5
	koko	6.5	10.5	10.5
	kotuku	5.5	27.5	20.5
	koula	17.5	56.5	3.5
	...			
Total Nb of Words		20,000 +0.5 x 1,000 = 20,500	25,000 +0.5 x 1,000 = 25,500	17,000 +0.5 x 1,000 = 17,500

total nb of words + (smooth * vocab)

Email 1: Koko kami kawa koula keke

$score(personal) = log(P(personal)) + log(P(koko|personal)) + log(P(kami|personal)) + log(P(kawa|personal)) + log(P(koula|personal)) + log(P(keke|personal))$
$= log(50/160) + log(6.5/20,500) + log(45.5/20,500) + log(78.5/20,500) + log(17.5/20,500) + log(0.5/20,500)$
$= -16.7561$

$score(work) = log(P(work)) + log(P(koko|work)) + log(P(kami|work)) + log(P(kawa|work)) + log(P(koula|work)) + log(P(keke|work))$
$= log(65/160) + log(10.5/25,500) + log(12.5/25,500) + log(1.5/25,500) + log(56.5/25,500) + log(5.5/25,500)$
$= -17.6373$

$score(promotion) = log(P(promotion)) + log(P(koko|promotion)) + log(P(kami|promotion)) + log(P(kawa|promotion)) + log(P(koula|promotion)) + log(P(keke|promotion))$
$= log(45/160) + log(10.5/17,500) + log(17.5/17,500) + log(67.5/17,500) + log(3.5/17,500) + log(80.5/17,500)$
$= -15.2227$

highest score is -15.2227 $\Longrightarrow$ *the most likely class is promotion*

		Folder		
		Personal	Work	Promotion
Word	kami	45.5	12.5	17.5
	kawa	78.5	1.5	67.5
	keke	0.5	5.5	80.5
	kiki	30.5	50.5	9.5
	koko	6.5	10.5	10.5
	kotuku	5.5	27.5	20.5
	koula	17.5	56.5	3.5
	...			
Total Nb of Words		20,500	25,500	17,500

Email 2: Keke kawa, koko Google koula keke!

notes:

- *ignore the word Google.*
- *keke counts twice*

$score(Personal) = log(P(personal)) + log(P(keke|personal)) + log(P(kawa|personal)) + log(P(koko|personal)) + log(P(koula|personal)) + log(P(keke|personal))$
$= log(50/160) + log(0.5/20,500) + log(78.5/20,500) + log(6.5/20,500) + log(17.5/20,500) + log(0.5/20,500)$
$= -18.7152$

$score(work) = log(65/160) + log(5.5/25500) + log(1.5/25500) + log(10.5/25500) + log(56.5/25500) + log(5.5/25500)$
$= -17.9939$

$score(promotion) = log(45/160) + log(80.5/17,500) + log(67.5/17,500) + log(10.5/17,500) + log(3.5/17,500) + log(80.5/17,500)$
$= -14.5599$

highest score is -14.5599 $\Longrightarrow$ *the most likely class is promotion*

COMP 472: Artificial Intelligence
Decision Trees
Solutions

1 Question

(a) Given the training instances below, use information gain to determine whether 'Outlook' or 'Windy' is the best feature to decide when to play a game of golf.

Outlook	**Temperature**	**Humidity**	**Windy**	**Golf**
sunny	hot	high	false	don't play
sunny	hot	high	true	don't play
overcast	hot	high	false	play
rain	mild	high	false	play
rain	cold	normal	false	play
rain	cold	normal	true	don't play
overcast	cold	normal	true	play
sunny	mild	high	false	don't play
sunny	cold	normal	false	play
rain	mild	normal	false	play
sunny	mild	normal	true	play
overcast	mild	high	true	play
overcast	hot	normal	false	play
rain	mild	high	true	don't play

Solution

$H(\text{Golf}) = H\left(\frac{5}{14}, \frac{9}{14}\right) = -\left(\frac{5}{14}\log_2\frac{5}{14} + \frac{9}{14}\log_2\frac{9}{14}\right) = 0.94$

$H(\text{Golf}|\text{Outlook=sunny}) = H\left(\frac{3}{5}, \frac{2}{5}\right) = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = \underline{0.97}$
$H(\text{Golf}|\text{Outlook=overcast}) = H(0, 1) = -(0\log_2 0 + 1\log_2 1) = \underline{0}$
$H(\text{Golf}|\text{Outlook=rain}) = H\left(\frac{2}{5}, \frac{3}{5}\right) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = \underline{0.97}$

$H(\text{Golf}|\text{Outlook}) = \frac{5}{14}\underline{0.97} + \frac{4}{14}\underline{0} + \frac{5}{14}\underline{0.97} = 0.69$

$\text{Gain(Golf, Outlook)} = H(\text{Golf}) - H(\text{Golf}|\text{Outlook}) = 0.94 - 0.69 = 0.25$

$H(\text{Golf}|\text{Windy=true}) = H\left(\frac{1}{2}, \frac{1}{2}\right) = 1$
$H(\text{Golf}|\text{Windy=false}) = H\left(\frac{1}{4}, \frac{3}{4}\right) = 0.81$

$H(\text{Golf}|\text{Windy}) = \frac{6}{14}1 + \frac{8}{14}0.81 = 0.89$

$\text{Gain(Golf, Windy)} = H(\text{Golf}) - H(\text{Golf}|\text{Windy}) = 0.94 - 0.89 = 0.05$

'Outlook' is a better feature because it has a higher information gain.

(b) Assume that we build a decision tree with the feature 'Outlook' as root. What would be the best feature to use as root of the sub-tree for the branch 'Outlook=sunny'. Again, use information gain.

Solution

To simplify the notation, let $A = (\text{Golf}|\text{Outlook=sunny})$

$H(A) = H\left(\frac{3}{5}, \frac{2}{5}\right) = -\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right) = 0.97$

$H(A|\text{Temperature=hot}) = H\left(\frac{2}{2}, \frac{0}{2}\right) = -\left(1\log_2 1 + 0\log_2 0\right) = 0$
$H(A|\text{Temperature=mild}) = H\left(\frac{1}{2}, \frac{1}{2}\right) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1$
$H(A|\text{Temperature=cold}) = H\left(\frac{0}{1}, \frac{1}{1}\right) = -\left(0\log_2 0 + 1\log_2 1\right) = 0$
$H(A|\text{Temperature}) = \frac{2}{5}0 + \frac{2}{5}1 + \frac{1}{5}0 = 0.4$
$\text{Gain}(A, \text{Temperature}) = H(A) - H(A|\text{Temperature}) = 0.97 - 0.4 = 0.57$

$H(A|\text{Humidity=high}) = H\left(\frac{3}{3}, \frac{0}{3}\right) = 0$
$H(A|\text{Humidity=normal}) = H\left(\frac{0}{2}, \frac{2}{2}\right) = 0$
$H(A|\text{Humidity}) = \frac{3}{5}0 + \frac{2}{5}0 = 0$
$\text{Gain}(A, \text{Humidity}) = H(A) - H(A|\text{Humidity}) = 0.97 - 0 = 0.97$

$H(A|\text{Windy=true}) = H\left(\frac{1}{2}, \frac{1}{2}\right) = 1$
$H(A|\text{Windy=false}) = H\left(\frac{2}{3}, \frac{1}{3}\right) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) = 0.27$
$H(A|\text{Windy}) = \frac{2}{5}1 + \frac{3}{5}0.27 = 0.56$
$\text{Gain}(A, \text{Windy}) = H(A) - H(A|\text{Windy}) = 0.97 - 0.56 = 0.41$

'Humidity' is a better feature for this branch since it has a higher information gain.

COMP 472: Artificial Intelligence

Classification Metrics

Solutions

1 Question

(a) Assume that you have the following datasets:

training set:

	a1	a2	a3	Expected Output
Data1	1.0	1.0	1.0	cat
Data2	1.5	1.0	2.0	dog
Data3	3.0	1.0	4.0	elephant
Data4	5.0	1.0	7.0	cat
Data5	3.5	1.0	5.0	elephant
Data6	4.5	1.0	5.0	cat
Data7	3.5	1.0	4.5	cat

validation set:

	a1	a2	a3	Expected Output
Data1	1.0	1.0	1.0	cat
Data2	1.5	1.0	2.0	dog
Data3	3.0	1.0	4.0	elephant
Data4	5.0	1.0	7.0	cat
Data5	3.5	1.0	5.0	elephant
Data6	4.5	1.0	5.0	cat
Data7	3.5	1.0	4.5	cat

test set:

	a1	a2	a3	Expected Output
Data1	1.0	1.0	1.0	cat
Data2	1.5	1.0	2.0	dog
Data3	3.0	1.0	4.0	elephant
Data4	5.0	1.0	7.0	cat
Data5	3.5	1.0	5.0	elephant
Data6	4.5	1.0	5.0	cat
Data7	3.5	1.0	4.5	cat

and you have built 2 models that give you the following outputs with the above data sets:

with the training set:

	model 1	model 2
Data1	cat	cat
Data2	dog	dog
Data3	dog	elephant
Data4	dog	cat
Data5	elephant	cat
Data6	dog	cat
Data7	cat	cat

with the validation set:

	model 1	model 2
Data1	dog	cat
Data2	dog	cat
Data3	dog	elephant
Data4	cat	cat
Data5	elephant	cat
Data6	dog	cat
Data7	cat	cat

with the test set:

	model 1	**model 2**
Data1	dog	cat
Data2	dog	cat
Data3	elephant	cat
Data4	dog	cat
Data5	dog	elephant
Data6	cat	cat
Data7	cat	cat

(a) What is the classification accuracy of model 1? What is the classification accuracy of model 2?

Solution

The classification accuracy is equal to the percentage of instances of the test set that the model correctly classifies. Let's calculate the accuracy for model 1 and model 2.
model 1:

$$Accuracy = \frac{4}{7} \times 100 = 57\%$$

model 2:

$$Accuracy = \frac{5}{7} \times 100 = 71\%$$

(b) Show the Confusion Matrix of both models with the test set.

Solution

Confusion matrix for model 1:

	Expected		
	Dog	**Cat**	**Elephant**
Predicted Dog	1	2	1
Predicted Cat	0	2	0
Predicted Elephant	0	0	1

Confusion matrix for model 2:

	Expected		
	Dog	**Cat**	**Elephant**
Predicted Dog	0	0	0
Predicted Cat	1	4	1
Predicted Elephant	0	0	1

(c) Calculate the Precision, Recall, and F1-measure of both models.

Solution

Let's evaluate **Model 1**:
True positive: diagonal position, which is equal to 1.
False positive: sum of row (without main diagonal), which is 3.
False negative: sum of column (without main diagonal), which is 0.
Precision of class Dog:

$$\frac{TP}{TP + FP} = \frac{1}{4} = 0.25$$

Recall of class Dog:

$$\frac{TP}{TP + FN} = \frac{1}{1} = 1$$

F1-measure of class Dog:

$$\frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2 \times 0.25 \times 1}{0.25 + 1} = 0.4$$

Using the same principle, we can compute P, R and F1 for the classes Cat and Elephant:

	Precision	**Recall**	**F1-measure**
Class Dog	0.25	1	0.4
Class Cat	1	0.5	0.66
Class Elephant	1	0.5	0.66

For **Model 2** we have:
Precision of class Cat:

$$\frac{TP}{TP + FP} = \frac{4}{6} = 0.67$$

Recall of class Cat:

$$\frac{TP}{TP + FN} = \frac{4}{4} = 1$$

F1-measure of class Cat:

$$\frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{0.67 \times 2}{1.67} = 0.80$$

	Precision	**Recall**	**F1-measure**
Class Dog	0	0	0
Class Cat	0.67	1	0.8
Class Elephant	1	0.5	0.67

COMP 472: Artificial Intelligence

k-means Clustering

Solutions

Python code associated with question 2 is available on Moodle.

Question 1 Consider the following data set with two attributes **a1** and **a2**.

	a1	**a2**
Data1	1.0	1.0
Data2	1.5	2.0
Data3	3.0	4.0
Data4	5.0	7.0
Data5	3.5	5.0
Data6	4.5	5.0
Data7	3.5	4.5

(a) Assume that we initialize the clusters using Data1 and Data4 as initial centroids. Using the Euclidean distance, in which cluster will each individuals be initially assigned? Do not perform the entire clustering - only do an initial assignment of points.

	Individuals	Centroid
Cluster 1	1	(1.0, 1.0)
Cluster 2	4	(5.0, 7.0)

For Data2:
Distance to Centroid 1: $\sqrt{(1.5-1.0)^2+(2.0-1.0)^2} \approx 1.1$
Distance to Centroid 2: $\sqrt{(1.5-5.0)^2+(2.0-7.0)^2} \approx 6.1$

	Distance to Centroid 1	Distance to Centroid 2
Data1	0.0	7.2
Data2	1.1	6.1
Data3	3.6	3.6
Data4	7.2	0.0
Data5	4.7	2.5
Data6	5.3	2.1
Data7	4.3	2.9

(b) Recalculate the centroids based on the current partition, reassign the individuals based on the new centroids. Which individuals (if any) changed clusters as a result?

For cluster 1:

$$\frac{1.0 + 1.5 + 3.0}{3} = 1.83$$

$$\frac{1.0 + 2.0 + 4.0}{3} \approx 2.33$$

	Individuals	Centroid
Cluster 1	1, 2, 3	(1.83, 2.33)
Cluster 2	4, 5, 6, 7	(4.13, 5.38)

	Distance to Centroid 1	Distance to Centroid 2
Data1	1.6	5.4
Data2	0.5	4.3
Data3	2.0	1.8
Data4	5.6	1.9
Data5	3.1	0.7
Data6	3.8	0.5
Data7	2.7	1.1

Data3 changed from cluster 1 to cluster 2. All other data points remained in the same cluster.

Question 2 Consider the following data set of points with two attributes **x** and **y**.

	x	**y**
Data1	0.0	1.0
Data2	1.0	0.0
Data3	-1.0	0.0
Data4	0.0	-1.0
Data5	0.5	0.5
Data6	-0.5	-0.5
Data7	-0.5	0.5
Data8	0.5	-0.5
Data9	4.0	4.0
Data10	-4.0	-4.0
Data11	-4.0	4.0
Data12	4.0	-4.0
Data13	4.0	0.0
Data14	-4.0	0.0
Data15	0.0	4.0
Data16	0.0	-4.0

(a) Apply k-means on the data set above given the 2 configurations of initial centroids indicated in the table below.

	Initial Centroid 1	Initial Centroid 2
Configuration 1	(0.0, 0.0)	(4.0, 4.0)
Configuration 2	(-5.0, 0.0)	(2.0, 0.0)

You need to apply k-means separately for each configuration and report the two clusters you have found for each setup.

Let's start with configuration 1:

Config. 1	Distance to Centroid 1	Distance to Centroid 2	Cluster
Data1	1.0	5.0	1
Data2	1.0	5.0	1
Data3	1.0	6.4	1
Data4	1.0	6.4	1
Data5	0.7	4.9	1
Data6	0.7	6.3	1
Data7	0.7	5.7	1
Data8	0.7	5.7	1
Data9	5.6	0.0	2
Data10	5.6	11.3	1
Data11	5.6	8.0	1
Data12	5.6	8.0	1
Data13	4.0	4.0	1
Data14	4.0	8.9	1
Data15	4.0	4.0	1
Data16	4.0	8.9	1

Now that we have assigned each data to their closest centroid, we have to re-compute the new clusters centroid and repeat the process until none of the data instances change cluster.

For cluster 1:

$$x_1 = \frac{0.0 + 1.0 - 1.0 + 0.0 + 0.5 - 0.5 + 0.5 - 0.5 - 4.0 + 4.0 - 4.0 + 4.0 - 4.0 + 0.0 + 0.0}{15} \approx -0.266$$

$$y_1 = \frac{0.0 + 1.0 - 1.0 + 0.0 + 0.5 - 0.5 + 0.5 - 0.5 - 4.0 + 4.0 - 4.0 + 0.0 + 0.0 + 4.0 - 4.0}{15} \approx -0.266$$

For cluster 2:

$$x_1 = \frac{4.0}{1} \approx 4.0$$
$$y_1 = \frac{4.0}{1} \approx 4.0$$

	Individuals	Centroid
Cluster 1	1, 2, ..., 8, 10, 11, 12, 13, 14, 15, 16	(-0.266, -0.266)
Cluster 2	9	(4.0, 4.0)

Let's calculate the distances to the new centroids for the second time.

Config. 1	Distance to Centroid 1	Distance to Centroid 2	Cluster
Data1	1.29	5.0	1
Data2	1.29	5.0	1
Data3	0.78	6.4	1
Data4	0.78	6.4	1
Data5	1.08	4.9	1
Data6	0.80	6.3	1
Data7	0.80	5.7	1
Data8	0.32	5.7	1
Data9	6.03	0.00	2
Data10	5.27	11.3	1
Data11	5.66	8.0	1
Data12	5.66	8.0	1
Data13	4.27	4.0	2
Data14	3.74	8.9	1
Data15	4.27	4.0	2
Data16	3.74	8.9	1

Let's calculate the centroids again:
For cluster 1:

$$x_1 = \frac{0.0 + 1.0 - 1.0 + 0.0 + 0.5 - 0.5 + 0.5 - 0.5 - 4.0 - 4.0 + 4.0 - 4.0 + 0.0}{13} \approx -0.615$$

$$y_1 = \frac{0.0 + 1.0 - 1.0 + 0.0 + 0.5 - 0.5 + 0.5 - 0.5 - 4.0 - 4.0 + 4.0 - 4.0 + 0.0}{13} \approx -0.615$$

For cluster 2:

$$x_1 = \frac{4.0 + 4.0}{3} \approx 2.667$$
$$y_1 = \frac{4.0 + 4.0}{3} \approx 2.667$$

Let's calculate the distances to the new centroids for the third time. But since one of the centroids is the same we can just copy the distance values for that centroid.

Config. 1	Distance to Centroid 1	Distance to Centroid 2	Cluster
Data1	1.72	3.14	1
Data2	1.72	3.14	1
Data3	0.72	3.66	1
Data4	0.72	3.66	1
Data5	1.57	3.06	1
Data6	0.02	4.48	1
Data7	1.11	3.84	1
Data8	1.11	3.84	1
Data9	6.52	1.88	2
Data10	4.78	9.42	1
Data11	5.72	6.79	1
Data12	5.72	6.79	1
Data13	4.65	2.98	2
Data14	3.44	7.18	1
Data15	4.65	2.98	2
Data16	3.44	7.18	1

Since none of the instances changed cluster, we can stop here and report the clusters we have found.

Let's apply k-means on our data set again using the second configuration:

Config. 2	Distance to Centroid 1	Distance to Centroid 2	Cluster
Data1	5.1	2.2	2
Data2	6.0	1.0	2
Data3	4.0	3.0	2
Data4	5.1	2.2	2
Data5	5.5	1.5	2
Data6	4.5	2.5	2
Data7	4.5	2.5	2
Data8	5.5	1.5	2
Data9	9.8	4.4	2
Data10	4.1	7.2	1
Data11	4.1	7.2	1
Data12	9.8	4.4	2
Data13	9.0	2.0	2
Data14	1.0	6.0	1
Data15	6.4	4.4	2
Data16	6.4	4.4	2

Now, we have to find the clusters centroid and repeat the process until none of the data instances change cluster.

For cluster 1:

$$x_1 = \frac{-4.0 - 4.0 - 4.0}{3} = -4.0$$

$$y_1 = \frac{-4.0 + 4.0 + 0.0}{3} = 0.0$$

For cluster 2:

$$x_1 = \frac{12.0}{13} \approx 0.92$$

$$y_1 = \frac{0.0 + 4.0}{13} = 0.0$$

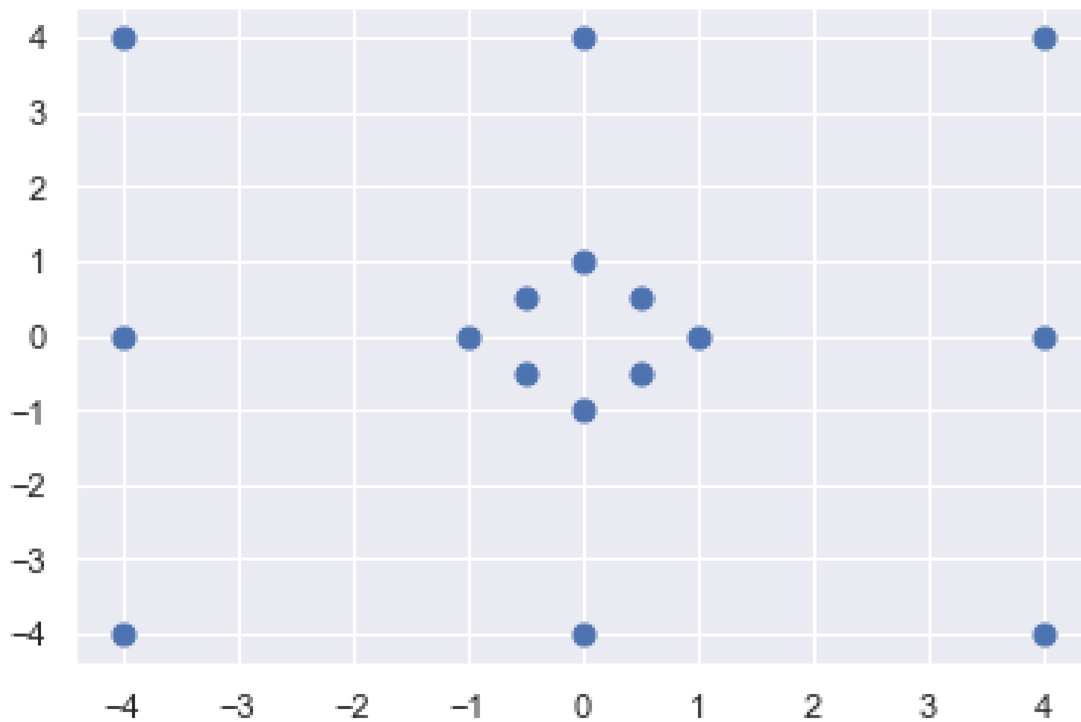
	Individuals	Centroid
Cluster 1	10, 11, 14	(-4.0, 0.0)
Cluster 2	1, 2, ..., 8, 9, 12, 13, 15, 16	(0.92, 0.0)

Let's calculate the distances to our new cluster centroids.

Config. 2	Distance to Centroid 1	Distance to Centroid 2	Cluster
Data1	4.1	1.3	2
Data2	5.0	0.1	2
Data3	3.0	1.9	2
Data4	4.1	1.3	2
Data5	4.5	0.6	2
Data6	3.5	1.5	2
Data7	3.5	1.5	2
Data8	4.5	0.6	2
Data9	8.9	5.1	2
Data10	4.0	6.3	1
Data11	4.0	6.3	1
Data12	8.9	5.1	2
Data13	8.0	3.1	2
Data14	0.0	4.9	1
Data15	5.6	4.1	2
Data16	5.6	4.1	2

None of the data instances changed cluster, so the algorithm will stop here and final centroids are the same as in the previous step.

(b) Plot the data and analyze your results from **part a**. Do the clusters you have found seem reasonable?
The figure below shows the distribution of the data. As the figure shows, the data cannot be naturally separated into two clear clusters and so k-means failed to find representative clusters. k-means is limited to linear cluster boundaries which means that it will fail to find more complicated boundaries.



Question 2-b Visualization of the data set

(c) Having **part a** in mind, were you surprised that the two runs of k-means with the 2 different initial configurations gave different resulting clusters? *No, k-means is very sensitive to the initial choice of centroids and usually, if you run k-means with two random sets of centroids you won't get the same results.*

COMP 472: Artificial Intelligence
Perceptrons
Solutions

Question 1 Assume that your local bank is trying to determine if a client should be considered to be a low-risk borrower or a high-risk borrower. Jim, the bank director, compiled a few data from previous experience based on the following criteria:

- The client has a mortgage or not.
- The client has an income inferior to 40,000$ or not.
- The client has a university degree or not.

Client	Has a ? mortgage?	Income $< 40,000\$$?	Has a University Degree?	Type of client?
A	No	Yes	Yes	low-risk
B	Yes	Yes	Yes	low-risk
C	No	Yes	Yes	low-risk
D	No	Yes	No	high-risk
E	Yes	No	No	high-risk

Assume that the weights of a perceptron are initialized this way:

- -0.6 for the Mortgage feature
- -0.2 for the Income feature
- +0.3 for the University degree feature

Show how the weights will be modified after each observation (each client) has been taken into account. Do only one iteration over the training data (1 epoch). Assume that a sign function is used and that all weights are always adjusted by a constant value of 0.1. Show all your work.

Let's assume that if $f(net) \geq 0$, we classify as low-risk; and otherwise, we classify as high-risk. Note that the opposite could have been assumed but the results would be different.

for A: $f(0 \times -.6 + 1 \times -.2 + 1 \times .3) = f(+.1) \geq 0$
So we conclude low-risk, which is correct. We don't change the weights at this point.

for B: $f(1 \times -.6 + 1 \times -.2 + 1 \times .3) = f(-.5) < 0$
So we conclude high-risk, which is incorrect. $f(net)$ is too low, we need to increase the weights of the active inputs (they are all active here).

$$w1 = -.6 + .1 = -.5$$
$$w2 = -.2 + .1 = -.1$$
$$w3 = +.3 + .1 = +.4$$

for C: $f(0 \times -.5 + 1 \times -.1 + 1 \times .4) = f(.3) \geq 0$
So we conclude low-risk, which is correct. We don't change the weights at this point.

for D: $f(0 \times -.5 + 1 \times -.1 + 0 \times .4) = f(-.1) < 0$
So we conclude high-risk, which is correct. We don't change the weights at this point.

for E: $f(1 \times -.5 + 0 \times -.1 + 0 \times .4) = f(-.5) < 0$
So we conclude high-risk, which is correct. We don't change the weights at this point.

The final weights are:

$$w1 = -.5$$
$$w2 = -.1$$
$$w3 = +.4$$

Question 2 Can a perceptron learn the SAME function of three binary inputs, defined to be 1 if all inputs are the same value and 0 otherwise? Either argue/show that this is impossible or construct a perceptron that correctly represents this function.

No. SAME is the complement of XOR. It is not linearly separable and therefore cannot be represented by a perceptron. Proof is by a figure showing the 3D space.

Question 3 Can a perceptron learn to correctly classify the following data, where each consists of three binary input values and a binary classification value: (111,1), (110,1), (011,1), (010,0), (000,0)? Either show that this is impossible or construct such a perceptron.

Yes. Output is 1 if at least 2 of the 3 inputs are 1. Therefore a perceptron with all three weights equal to 0.5 and a threshold value of 0.8 will work.

$$0.5i_1 + 0.5i_2 + 0.5i_3 \geq 0.8 \textit{ then output } = 1$$
$$0.5i_1 + 0.5i_2 + 0.5i_3 < 0.8 \textit{ then output } = 0$$