CS-540 Homework Assignment #3: Introduction to Machine Learning Concepts

Assigned: Saturday, March 4th
Due: Saturday, March 11th

Hand-In Instructions

This assignment includes only written problems. Hand in all parts electronically by uploading them in *a single zipped file* to the assignment page on Canvas.

Your answers to each written problem should be turned in as separate pdf files called <wisc NetID>-HW3-P1.pdf, <wisc NetID>-HW3-P2.pdf, and <wisc NetID>-HW3-P3.pdf. Use your username, not your ID number.

You can write up your answers in a medium of your choice – pencil and paper, word processor, LaTeX – as long as you show your work fluidly and neatly. If you handwrite your solutions to written problems, make sure to scan them in. *No photographed assignments will be accepted.*

Once you are finished, put your three PDF files into a single directory. Zip it, name it <wisc NetID>-HW3, and upload it to the assignment Canvas page.

Late Policy

All assignments are due **at 11:59 p.m.** on the due date. One (1) day late, defined as a 24-hour period from the deadline (weekday or weekend), will result in 10% of the total points for the assignment deducted. So, for example, if a 100-point assignment is due on a Wednesday and it is handed in between any time on Thursday, 10 points will be deducted. Two (2) days late, 25% off; three (3) days late, 50% off. No homework can be turned in more than three (3) days late. Written questions and program submission have the same deadline. A total of three (3) free late days may be used throughout the semester without penalty. Assignment grading questions must be raised with the instructor or a TA within one week after the assignment is returned.

Collaboration Policy

You are to complete this assignment individually. However, you are encouraged to discuss the general algorithms and ideas with classmates, TAs, and instructor to help you answer the questions.

You are also welcome to give each other examples that are not on the assignment in order to demonstrate how to solve problems.

But we require you to:

- not explicitly tell each other the answers
- not to copy answers or code fragments from anyone or anywhere
- not to allow your answers to be copied
- not to get any code on the Web

In those cases where you work with one or more other people on the general discussion of the assignment and surrounding topics, we suggest that you specifically record on the assignment the names of the people you were in discussion with.

Problem 1: Hierarchical Agglomerative Clustering [12]

Consider the following information about distances between pairs of U.S. cities:

	BOS	NY	DC	MIA	CHI	SEA	SF
BOS	0	206	429	1504	963	2976	3095
NY	206	0	233	1308	802	2815	2934
DC	429	233	0	1075	671	2684	2799
MIA	1504	1308	1075	0	1329	3273	3053
CHI	963	802	671	1329	0	2013	2142
SEA	2976	2815	2684	3273	2013	0	808
SF	3095	2934	2799	3053	2142	808	0

a) [9] Run hierarchical clustering using **complete-linkage** on the above data. Show the intermediate input in the form of a table at each step and draw the final dendrogram.

b) [3] What clusters of cities are created if you want 4 clusters?

Problem 2: K-Means Clustering [18]

Training Instance	X	Y
A	1	4
B	4	8
C	7	2
D	0	3
E	5	6
F	2	1
G	6	0
H	5	3
I	2	5

a) [12] Perform **one iteration** (once through the body of the algorithm while-loop) of K-Means clustering on the following data points. Assume k=2, and that your initial cluster centers (2,5) and (6,6). You can refer to them as clusters 1 and 2 respectively.

For the distance metric between points, use Manhattan distance, which for points (x_1, y_1) and (x_2, y_2) is equal to $|x_1 - x_2| + |y_1 - y_2|$. Break any ties in favor of the 1st cluster.

This means for all training instances you should list

- the nearest cluster for each training instance
- the distance between each training instance and the nearest cluster

b) [4] Now that you have computed which points belong to each cluster, you need to compute two new cluster centers.

What are the new centers for each cluster after the previous iteration? **Hint:** the formula for the centroid of two-dimensional points is (average of x-coordinates, average of y-coordinates)

c) [2] At the point, would the K-Means algorithm perform another iteration, or would it meet the conditions for terminating? Justify briefly.

Problem 3: Decision Trees [42]

Below is some (very fake) data that represents scraping information from a web page. One might be interested in automatically downloading relevant photographs to accompany the text of news articles but not advertisements.

The features represent characteristics of an image, and the labels on the far right column represent the image's classification. The labels available in this dataset are {"ad", "not ad"}.

hasCaption	sizeInPixels	humanFaceLikelihood	paidMembership	label
0	300	0.2	1	Ad
1	500	0.1	1	Ad
1	712	0.9	0	Not ad
1	600	0.5	1	Ad
0	400	0.6	1	Ad
1	1200	0.3	0	Not ad
1	1540	0.8	1	Not ad
1	800	0.7	0	Ad
0	720	0.2	0	Ad
1	900	0.3	1	Not ad
0	1000	0.4	1	Not ad

a) [2] Is this a concept learning problem? Why or why not? (Be brief.)

b) [5] For each feature (and label) in the dataset, classify the variable as categorical, ordinal, or real/continuous.

c) [25] Compute the information gain of each feature at the root of the decision tree. Show the computation of entropy, conditional entropy, and specific conditional entropy values.

For real-valued features, compute candidate **binary** splits using the technique described in the lecture slides (which is also described on 707 in the textbook). Assume that the right side of those hypothetical splits is > and the left is ≤.

d) [5] Assume that you are using the ID3 algorithm from the slides (aka the buildTree algorithm in the textbook.) Given your computations in part c, which feature is chosen to split on at the root? Break ties in favor of leftmost attributes. Which examples are passed down to each child node in the recursive calls that generate subtrees?

e) [5] Imagine you are working with a scientist interested in using machine learning to diagnose a rare disease. Of the 1000 patient samples available to you in the form of feature vectors, only 11 have the condition. The scientist has used *the ID3 decision tree learning* algorithm to build a decision tree on 700 of the samples, and the resulting tree has an error rate of less than 1% on the 300 samples held aside as test data. She is very excited about the results and wants to use the model in her practice.

Is this test set accuracy a guarantee that the decision tree will be able to correctly identify sick patients in the real world? Justify your answer using properties of the ID3 algorithm, entropy, and typical machine learning workflow.