Homework 1: Classification

Release Date: September 19, 2017

Due Date: October 3, 2017 (upon the start of class)

1. (30 points). The following table consists of the training data from an employee database. Let *salary* be the class attribute. In C4.5 decision tree algorithm, a tree is built by multiple-branch splitting based on **Gain Ratio** as the uncertainty measure. Which attribute should be selected to split the records in the first iteration? (**Illustrate all necessary steps about how you get the answer**. **Source code is not required but acceptable as necessary steps for the calculations**.)

department	*status*	*age*	*salary*
sales	senior	31-40	Medium
sales	junior	21-30	Low
sales	junior	31-40	Low
systems	junior	21-30	Medium
systems	senior	31-40	High
systems	junior	21-30	Medium
systems	senior	41-50	High
marketing	senior	31-40	Medium
marketing	junior	31-40	Medium
secretary	senior	41-50	Medium
secretary	junior	21-30	Low

2. (40 points). Consider the training data set below. The class label Play (whether to play tennis or not) is determined by the values of four attributes for weather conditions.

Outlook	Temperature	Humidity	Windy	Play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

(a) Build a Naïve Bayesian Classifier (NBC) based on the training data (show all probabilities used by the classifier). (10 points)

(b) Use the classifier built in part (a) to predict whether to play tennis or not when Outlook = sunny, Temperature = cool, Humidity = high, and Windy = True. (**Show all necessary calculation steps**. **Source code is NOT acceptable**) (10 points)

(c) According to the training data table, are attributes Humidity and Windy independent or not? Justify your answer. (10 points)
(d) According to the training data table, are attributes Humidity and Windy conditionally independent or not knowing the class label Play (yes or not)? Justify your answer. (10 points)

3. (15 points). Given a set of data points in the following table where X and Y are two numeric attributes and C is the class label. Use kNN algorithm to find the class label for data point (2, 2) when K=1, and when K=3. Use Euclidean distance as the distance measure, and use majority vote to determine the class label based on multiple neighbors.

X	Y	C
0	0	1
1	0	1
1	1	1
4	3	2
4	2	2
2	4	2

4. (5 points). Assume the data table in Question 1 is generalized to the following table. For a given row entry, the additional attribute "c*ount*" represents the number of data records having the values for "department", "status", "age", and "salary" given in that row. How would you modify the basic decision tree algorithm to take into consideration of each generalized data record in an efficient way? Briefly explain your solution based on the example training data.

department	*status*	*age*	*salary*	*count*
sales	senior	31-40	Medium	*30*
sales	junior	21-30	Low	*40*
sales	junior	31-40	Low	*40*
systems	junior	21-30	Medium	*20*
systems	senior	31-40	High	*5*
systems	junior	21-30	Medium	*3*
systems	senior	41-50	High	*3*
marketing	senior	31-40	Medium	*10*
marketing	junior	31-40	Medium	*4*
secretary	senior	41-50	Medium	*4*
secretary	junior	21-30	Low	*6*

5. (10 points). An eager learner (e.g., decision tree) builds a fixed and global model based on the training data and uses the same global model for each test instance, while a lazy learner (e.g., k-nearest neighbor) builds a flexible and local model based on the training data for each test instance. First, suggest a lazy version of the eager decision tree algorithm by describing the main idea of the new algorithm, and then discuss the *advantages* and *disadvantages* of the new algorithm compared with the eager decision tree algorithm, and the *advantages* and *disadvantages* of the new algorithm compared with the lazy kNN algorithm.