CS 419 Homework Assignment 1

Due: 7:00pm, Wednesday, September 30th

This assignment is scored out of 70. It consists of 4 questions. When you submit, you are required to create a folder with your name (Last name first, then First name), CS419, HW1, e.g., LastName_ FirstName _CS419_HW1. Type your answers into a text file (only .txt, .doc., and .pdf file formats are accepted) and save it in this folder. Put all your java programs as well as your output files (as .txt files preferably) in a folder titled with ***prog***. Your ***prog*** folder should appear as a subfolder of your homework folder, i.e., LastName_ FirstName _CS419_HW1. Zip this folder, and submit it as one file to Desire2Learn. Do not hand in any printouts.

Short Answers

P1. (10pts) The nearest-neighbor algorithm described in Lecture 2 can be extended to handle nominal attributes. A variant of the algorithm called PEBLS (Parallel Examplar-Based Learning System) by Cost and Salzberg [2] measures the distance between two values of a nominal attribute using the modified value difference metric (MVDM). Given a pair of nominal attribute values, V_1 and V_2, the distance between them is defined as follows:

$$d(V_1, V_2) = \sum_{i=1}^{k} \left| \frac{n_{i1}}{n_1} - \frac{n_{i2}}{n_2} \right|,$$

where n_{ij} is the number of examples from class i with attribute value V_j and n_j is the number of examples with attribute value V_j .

Consider the training set for the loan classification problem **shown in the figure below**. Use the MVDM measure to compute the distance between every pair of attribute values for the **Home Owner** and **Marital Status** attributes.

Tid	Home Owner	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	Yes
5	No	Divorced	95K	Yes
6	No	Single	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

P2. (10pts) Why classifying unknown records with the nearest-neighbor algorithm is relatively expensive? Compared to this algorithm, what would you think about the support vector machine (SVM) with respect to the efficiency? *Hint: you can discuss the time cost for each phase of SVM.*

P3. (5pts) A report shows that 13% of the undergraduate students smoke while this number is 25% for the graduate students. If in a university, 3/4 of the students are undergraduate students and the rest are graduate students, can you derive the probability that a student who smokes is an undergraduate student?

Programming Questions

P4. (45pts) A high-level summary of the nearest-neighbor classification method is given in the following algorithm. It computes the distance (or similarity) between each test example $z = (\boldsymbol{x}', y')$ and all the training examples $(\boldsymbol{x}, y) \in D$ to determine its nearest-neighbor list, D_z.

Algorithm: The $\boldsymbol{k}$-nearest neighbor classification algorithm

---

1. Let k be the number of nearest neighbors and D be the set of training examples.
2. **for** each test example $z = (\boldsymbol{x}', y')$ **do**
3. Compute $d(\boldsymbol{x}', \boldsymbol{x})$, the distance between z and every example, $(\boldsymbol{x}, y) \in D$.
4. Select $D_z \subseteq D$, the set of k closest training examples to z.
5. $y' = \underset{v}{\operatorname{argmax}} \sum_{(\boldsymbol{x}_i, y_i) \in D} I(v = y_i)$
6. **end for**

Once the nearest-neighbor list is obtained, the test example is classified based on the majority class of its nearest neighbors:

$$\text{Majority Voting: } y' = \underset{v}{\operatorname{argmax}} \sum_{(\boldsymbol{x}_i, y_i) \in D} I(v = y_i)$$

where v is a class label, y_i is the class label for one of the nearest neighbors, and $I(\cdot)$ is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

As we have discussed in Lecture 2, there is also a distance-weighted voting scheme that weights the votes with the distances between examples:

$$\text{Distance} - \text{Weighted Voting: } y' = \underset{v}{\operatorname{argmax}} \sum_{(\boldsymbol{x}_i, y_i) \in D} w_i \times I(v = y_i)$$

where $w_i = 1/\, d(\boldsymbol{x}', \boldsymbol{x_i})^2$ is the weight.

Your tasks:

1. Implement the $\boldsymbol{k}$-NN algorithm that is illustrated above, using ***Euclidean distance*** and the ***majority voting scheme***.
2. Read the specification of Glass Identification Data Set, which can be obtained here: https://archive.ics.uci.edu/ml/datasets/Glass+Identification
3. You should write code to randomly pick up 75% of the records as the training set and use the rest of them as the test set. Note that in the original dataset, the last column is

the class label. You should make sure that in your test set, you have records that are from different classes.

4. Run your program with different k's (k = {1, 5, 10}) on the training set and test set that you have generated. Record the classification accuracies. You can use the number of correctly classified records in your test set divided by the number of total records in your test set as the accuracy percentage.
5. Implement the ***k***-NN algorithm that is illustrated above, using *Euclidean distance* and the *distance-weighted voting scheme*.
6. Redo step 4 and record the classification accuracies.
7. Report and briefly discuss your results.

Note: You are required to put your code as well as a report document (.txt, .doc, or .pdf) in the ***prog*** folder.