CMPT 741 Fall 2019
Data Mining
Martin Ester
TA: Zahra Zohrevand

Assignment 2

Total marks: 100
Due date: October 15, 2019

Assignment 2.1 (30 marks)

Given the following dataset of bank clients with categorical attributes Age, Salary, City, and Creditworthiness:

Age	Salary	City	Creditworthiness
young	high	Vancouver	bad
medium	medium	Burnaby	bad
young	low	Vancouver	bad
old	medium	Coquitlam	good
old	high	Richmond	good
medium	low	Richmond	bad
young	medium	Vancouver	bad
old	low	Burnaby	bad
young	medium	Coquitlam	good
young	medium	Vancouver	good

Given the above dataset of bank clients with categorical attributes Age, Salary, City, and Creditworthiness. We have trained a Naïve Bayes classifier to predict the creditworthiness of a bank client, i.e. we are using Creditworthiness as the class label. Here are the parameters of the classifier (computed without using Laplacian smoothing, and using C as abbreviation for Creditworthiness):

P(Creditworthiness=good)=0.4 P(Creditworthiness=bad)=0.6

P(Age=young|C=good) = 0.5 P(Age=medium|C=good) = 0.0 P(Age=old|C=good) = 0.5
P(Age=young|C=bad) = 0.5 P(Age=medium|C=bad) = 0.34 P(Age=old|C=bad) = 0.16

P(Salary=low|C=good) = 0.0 P(Salary=medium|C=good) = 0.75
P(Salary=high|C=good) = 0.25
P(Salary=low|C=bad) = 0.5 P(Salary=medium|C=bad) = 0.34
P(Salary=high|C=bad) = 0.16

P(City=Vancouver|C=good) = 0.25 P(City=Burnaby|C=good) = 0.0
P(City=Coquitlam|C=good) = 0.5 P(City=Richmond|C=good) = 0.25
P(City=Vancouver|C=bad) = 0.5 P(City=Burnaby|C=bad) = 0.34
P(City=Coquitlam|C=bad) = 0.0 P(City=Richmond|C=bad) = 0.16

a) What class label (Creditworthiness) does the Naïve Bayes classifier predict for an unseen client with Age = young, Salary = low and City = Vancouver? Show the necessary computations.

b) Given the Naïve Bayes assumption and the parameters of the above Naïve Bayes classifier, what is the probability of observing a client with Age = old, Salary = medium and City = Vancouver? Show the necessary computations.

Assignment 2.2 (30 marks)

Labeled data is typically much harder to obtain than unlabelled data, because labelling requires a domain expert and is therefore expensive and time-consuming. For example, you may have a large dataset of patients with their demographic and clinical attributes, but only a small dataset of patients with all of these attributes and a diagnosis, i.e. a class label. Consider a situation where you want to learn a classifier and have a small labeled dataset L (with features and class labels) and a large unlabeled dataset U (only features). The goal is to predict the class labels of all examples in U. How can you employ the datasets L and U to train a classifier that is more accurate than a classifier trained using only dataset L?

Hint: learn a sequence of classifiers with increasingly better accuracy.

a) Explain your idea in plain English.

b) Present the pseudo-code of your solution, i.e. a function
Classification (labeled dataset L, unlabeled dataset U), which returns the class labels for all examples in U.
Your pseudo-code can use the following functions (which you do not have to implement):
TrainClassifier (training dataset T),
which returns a classifier (trained on labelled dataset T)
Classifier (test case t), which returns the class label of the test case and the confidence of the prediction.

Assignment 2.3 (40 marks)

Design a decision tree classifier that is scalable to large secondary storage data sets. This decision tree classifier needs to assume that training data are not in main memory, but on secondary storage. Training data can be accessed (read / written) only block-wise, and the cost of accessing a disk block is several orders of magnitude larger than for memory-resident data. Consequently, I/O cost (measured by the number of blocks read or written) becomes the dominating cost factor.

The most expensive operations of a decision tree classifier are as follows:

- Evaluation of all potential splits and selection of the best one.
- Partitioning of the training data according to the chosen split.

To efficiently support these operations on secondary storage data sets, CH (Class Histogram) sets can be used. The CH set for decision tree node N and attribute A contains one class histogram (i.e., frequency values for each class) for each value a of A, representing the class distribution over all training data records belonging to N and having an A value of a. For example, a CH set for some attribute a with values a1, a2, a3, a4 and classes c1, c2, c3 may look as follows:

a1: c1: 70, c2: 10, c3: 20

a2: c1: 45, c2: 15, c3: 40

a3: c1: 45, c2: 15, c3: 40

a4: c1: 45, c2: 15, c3: 40

The CHgroup of a node N consists of the set of CH sets for all attributes of node N. We assume that the whole CH group of the root node fits into main memory. Thus, for each node of the decision tree the corresponding CH group can be kept memory resident.

The cost of the growth-phase clearly dominates the cost of the pruning phase and, therefore, we ignore the pruning phase here.

a) Design an algorithm for decision tree construction from secondary storage training data based on CH sets and CH groups. The algorithm needs to build (memory-resident) CH groups / sets from the (secondary-storage) training data and then build a decision tree based on the information contained in the CH groups / sets. Explain your design and provide the pseudo- code for your algorithm.

b) Analyse the runtime complexity of your algorithm using the number of disk block accesses (reads and writes) as the cost measure. How often do you have to read and / or write the whole training data set?