CPSC 532P — Spring 2017

Assignment 1 - Part A

Due: 12:00 pm, Wednesday 11 January 2017

The aim of this assignment is to play with some learning and probabilistic models. This assignment is in two parts; in the first part we will get some preliminary results that we will discuss in class. In the second part (due the following week) we will try to get better solutions. You can do this in groups any size.

Question 1

The simplest learning problem is where there is a single Boolean target feature (with range $\{0,1\}$) and no input features and the examples are independent and identically distributed (i.i.d.).

The training data can be characterized by two numbers n_0, the number of 0's in the training data and n_1 the number of 1's in the training data. You are to design a predictor that works best on **test examples**. Given n_0 and n_1 you can make a prediction $\widehat{p}$ that is any real number. You need to test your predictions on the test data according to the following criteria:

- sum of squares error: $\sum_{e \in test\ examples} (\widehat{p} - e)^2$
- sum of absolute error: $\sum_{e \in test\ examples} |\widehat{p} - e|^2$
- log loss, the negative of log likelihood: $\sum_{e \in test\ examples} e * -\log_2 \widehat{p} + (1-e) * -\log_2 (1-\widehat{p})$
  which is undefined if either of the logarithms is given a number less than or equal to 0.

To evaluate your predictor, you should run the following 1000 times, to get a reasonable average:

- Generate a random number p in range [0,1], which is the ground truth. In Python this can be done using `p=random.random()`
- Generate n training examples samples according to p (for each example make it true with probability p). In Python this can be done using `e=1 if random.random()<p else 0`.
- Using the training examples generated above, implement $\widehat{p}$.
- Generate a number of test examples (say 100) from the same p. Evaluate your predictor on the test examples for each of the criteria.

Do this for a number of values of n, say 1, 2, 3, 4, 5, 10, 20, 100, 1000. Report your results on the Wiki (see the course web page). We will aim for a table of diverse predictors.

Question 2

MovieLens (`https://movielens.org/`) is a movie recommendation system that acquires movie ratings from users. The rating is from 1 to 5 stars, where 5 stars is better. A rating dataset for such a problem is available from `http://grouplens.org/datasets/movielens/` The ratings data is of of the form of Figure 1, where each user is given a unique number, each item (movie) is given a unique number, and the timestamp is the Unix standard of seconds since 1970-01-01 UTC. The gender of the users (and other info) is also provided in a separate file. We will be using a subset of the 100K dataset.

User	*Item*	*Rating*	*Timestamp*
196	242	3	881250949
186	302	3	891717742
22	377	1	878887116
...	...	...	...

Figure 1: Ratings Data from the MovieLens dataset

Please download both: `http://files.grouplens.org/datasets/movielens/ml-100k.zip` and `http://cs.ubc.ca/~poole/cs532/2017/as1/As1_starter.py` (requires Python 3)
The Python file creates a subset of the ratings we will use, a training set for the gender and a test set.

Write a predictor that makes predictions for the gender for each user in the test set (based on the ratings they have given). Post a description of your algorithm, and the accuracy on the Wiki. We want as many different algorithms as you can think up.

Question 3

Suppose we want to diagnose school student's subtraction of multi-digit binary numbers. Consider subtracting a two-digit number from a three digit number.

That is, problems of the form:

$$\begin{array}{ccc} X_2 & X_1 & X_0 \\ - & Y_1 & Y_0 \\ \hline Z_2 & Z_1 & Z_0 \end{array} \qquad e.g. \qquad \begin{array}{ccc} 1 & 0 & 1 \\ - & 1 & 1 \\ \hline & 1 & 0 \end{array}$$

Here X_i, Y_i and Z_i are all binary digits.

In this question you will represent this problem as a belief network, and test it with a belief network implementation.

(a) Describe how to do multi-digit binary subtraction. [This procedure that you are assuming students are carrying out will affect the network produced.] What errors would you expect students to make?

(b) What variables are needed to model subtraction of this form and the errors students could make? Give a DAG that specifies the dependence of these variables. [Hint: think about what will be observed, what will be queried, and hidden variables that you may have used in your description.]

(c) What are reasonable conditional probabilities for this domain?

(d) Implement this, for example using the belief network tool at: `http://www.aispace.org/bayes/` Or the Python Bayes net code at `http://artint.info/code/python/`. Test the implementation (don't forget to save your graph).

(e) Explain how to extend your graph to allow for two different subtraction problems to help us better test a student. [Hint: What nodes should be shared and what should be replicated?] It is up to you whether you implement this, but it's probably instructive to do so.