Homework 4
Statistical methods in AI/ML

Instructor: Vibhav Gogate
Vibhav.Gogate@utdallas.edu

Due: May 1, 2021 via elearning

Learning algorithms [80 points]

Download the zip file containing the data sets from the class webpage.

The zip file contains 3 directories. Each directory contains a Bayesian network in the UAI 2008 format (this format is slightly different from the UAI 2014 format you used for earlier homeworks and its description is linked on the class webpage) and 9 data files. The Bayesian network is the ground truth (the data sets are constructed by iid sampling of the Bayesian network). Data sets in files: train-f-1.txt to train-f-4.txt are fully observed. Data sets in files train-p-1.txt to train-p-4.txt are partially observed (some values are missing). The data set in the file test.txt is the test data. All variables are binary, they take a value from the set $\{0, 1\}$.

Data file format

The first line has two integers. The first integer gives the number of variables and the second integer gives the number of examples (or samples). Let us denote the number of examples by M. The second line through line $M + 1$ is the data itself, one example or sample per line. Missing values are denoted by the symbol "?".

For example, the following represents a data set of size 3 over 5 variables

```
5 3
0 1 0 1 ?
1 0 1 0 ?
0 1 1 ? 0
```

The first example represents the assignment of values 0, 1, 0, 1 and ? to variables indexed by 0, 1, 2, 3, and 4 respectively. The second example represents the assignment of values 1, 0, 1, 0 and ? to variables indexed by 0, 1, 2, 3, and 4 respectively, and so on.

1. Task 1: Implement the Bayesian network parameter learning algorithm assuming fully observed data and known structure. (Use the maximum likelihood approach.) Let us call this algorithm FOD-learn.

2. Task 2: Implement the EM algorithm for learning the parameters of a Bayesian network assuming partially observed data and known structure. For each example, assume that the number of missing values is bounded by 8. This will enable you to perform exact inference without implementing the junction tree algorithm. Namely, for each example, given m missing values, construct 2^m weighted completions. Run the EM algorithm for 20 iterations only and repeat for 5 random initializations. Report the mean and standard deviation of LLDiff (see Eq. 1) on the test set for the 5 initializations. Let us call this algorithm POD-EM-Learn.

3. Task 3: Let $\mathbf{X} = \{X_1, \ldots, X_n\}$ be a set of variables. Construct k random DAG structures over $\mathbf{X}$ such that the number of parents of each node is less than or equal to 3. Let Θ_i where $1 \leq i \leq k$ denote the parameters of the i-th Bayesian network associated with the i-th DAG structure. Implement the EM algorithm for learning the parameters of the following mixture model:

$$P(\mathbf{X}) = \sum_{i=1}^{k} p_i P_i(\mathbf{X}; \Theta_i)$$

where $\sum_{i=1}^{k} p_i = 1$, $p_i \geq 0 \; \forall i$ and $P_i(\mathbf{X}; \Theta_i)$ is the probability distribution associated with the i-th Bayesian network.

Note that the parameters of the mxiture model are $p_1, \ldots, p_k$ and $\Theta_1, \ldots, \Theta_k$. Run the EM algorithm for 20 iterations only and repeat for 5 random initializations. Vary k from 2 to 6 in increments of 2 (namely try $k = 2, 4, 6$). Report the mean and standard deviation of LLDiff (see Eq. 1) on the test set for the 5 initializations (and for each value of k). Let us call this algorithm Mixture-Random-Bayes.

How to test your algorithms?

- Task 1 and Task 3: Train on data sets: train-f-1.txt to train-f-4.txt. Compute the log-likelihood of the test data for each of your 4 learned models.
- Task 2: Train on data sets: train-p-1.txt to train-p-4.txt. Compute the log-likelihood of the test data for each of your 4 learned models.

Deliverables:

1. Your code. For each task, the input to your program should be a UAI file, training data file and test data file and it should output the cumulative, pointwise difference between the log-likelihoods on the test data computed using the input Bayesian network (the ground truth) and the learned model. Formally, let $\mathcal{B}_o$ and $\mathcal{B}_l$ denote the original Bayesian network and the learned model respectively. Let $\mathcal{D} = (\mathbf{x}[1], \ldots, \mathbf{x}[M])$ denote the test data set. Let $LL(\mathcal{B}, \mathbf{x}[i])$ denote the log-likelihood of $\mathbf{x}[i]$ w.r.t. $\mathcal{B}$. Then,

$$\text{LLDiff} = \sum_{i=1}^{M} |LL(\mathcal{B}_o, \mathbf{x}[i]) - LL(\mathcal{B}_l, \mathbf{x}[i])| \tag{1}$$

For example, when I run your program, I should see the following output:

```
./program <input-uai-file> <task-id> <training-data> <test-data>
----------------------------

log likelihood difference = 1245.2892
----------------------------
```

where *task-id* can be either 1, 2 or 3. To ensure that the log-likelihood is not undefined (when likelihood is zero, log is undefined), replace all zeros in the network by a small constant (e.g., 10^{-5}). Please make sure that each CPT is valid when you make this change (namely, ensure that $\sum_x P(X = x|\mathbf{U} = \mathbf{u}) = 1$).

2. For each of the three Bayesian networks, the following table filled with the log-likelihood difference for the test data given the Bayesian network learned using data sets train-*-1.txt to train-*-4.txt.

Algorithm	LLDiff Train-1	LLDiff Train-2	LLDiff Train-3	LLDiff Train-4
FOD-learn				
POD-EM-learn				
Mixture-Random-Bayes (k=2)				
Mixture-Random-Bayes (k=4)				
Mixture-Random-Bayes (k=6)				

3. Based on your experimental results, compare the FOD-learn algorithm with the POD-EM-learn algorithm. What is the impact of missing data on LLDiff (Think of LLDiff as an error measure).

4. Based on your experimental results, compare the FOD-learn algorithm with the Mixture-Random-Bayes algorithm. What is the impact of not knowing the precise Bayesian network structure on LLDiff. How does k affect the accuracy of the learned model?

EM algorithm (20 points) KF: Koller & Friedman book

- (10 points) KF Exercise 19.4
- (5 points) Consider the dataset given below. All variables are Binary and take values from the domain $\{0, 1\}$. "?" denotes a missing value.

A	B	C	D
0	1	1	?
1	0	0	?
0	1	1	?
1	1	0	?
0	0	1	?
1	1	1	?

Assume that you are learning the parameters of a Bayesian network having the following structure: D is the root node (having no parents). The parent of A is D, B is D and C is D. (Thus the only edges in the Bayesian network are $D \rightarrow A$, $D \rightarrow B$ and $D \rightarrow C$). Starting with probabilities that are initialized uniformly (i.e., all probabilities are initialized to 0.5), calculate the parameters of this Naive Bayes model using the EM algorithm. Stop at convergence or after 3 iterations, whichever is earlier. Does the EM algorithm converge and after how many iterations?

- (5 points) Let us generalize our experience with such datasets, the above Bayesian network and EM with uniform initialization. Assume that you are given a dataset such that D is (always) missing but A, B and C are observed in all examples in the dataset. Assume that you will learn the parameters of the Bayesian network given above using the EM algorithm with uniform initialization. Answer the following questions based on these assumptions.
  - At convergence, what will be the parameters of the Naive Bayes model?
  - After how many iterations will EM converge?