Homework 5
Statistical methods in AI/ML

Instructor: Vibhav Gogate
Vibhav.Gogate@utdallas.edu

Due: May 1, 2013 via elearning

1. Implement Learning algorithms [70 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 format and 11 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-5.txt are fully observed. Data sets in files train-p-1.txt to train-p-5.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-param. (15 points)

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. (15 points)

3. Task 3: Implement the Chow-Liu algorithm for learning a tree Bayesian network (fully observable case). Use the implementation in Task 1 to learn the parameters (CPTs) for this tree. (15 points)

How to test your algorithms?

- Task 1 and Task 3: Train on data sets: train-f-1.txt to train-f-5.txt. Compute the log-likelihood of the test data for each of your 5 learned models.
- Task 2: Train on data sets: train-p-1.txt to train-p-5.txt. Compute the log-likelihood of the test data for each of your 5 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 learned network in the UAI format. It should also 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,

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

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

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

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

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-*-5.txt. (25 points)

Algorithm	LL-diff Train-1	LL-diff Train-2	LL-diff Train-3	LL-diff Train-4	LL-diff Train-5
FOD-param					
EM					
Chow-Liu					

2. Sampling Algorithms (20 points)

1. (Koller and Friedman, Exercise 12.19) Show hat the Gibbs sampling algorithm is a special case of the Metropolis-Hastings algorithm. More precisely, provide a particular proposal distribution Q, that induces precisely the same distribution over the transitions taken as the associated Gibbs transition distribution. [10 points]

2. Consider a Markov network having zeros (namely there exists one or more assignments $\mathbf{x}$ that contribute nothing to the partition function, namely $\prod_i f_i(\mathbf{x}) = 0$). Is the network ergodic? Justify your answer. [10 points]

3. MPE (10 points) AD: Adnan Darwiche's book.

- (4 points) AD problem 10.7
- (4 points) AD problem 10.8
- (2 points) AD problem 10.9