Homework 4

CS 6347: Statistical methods in AI/ML

Instructor: Vibhav Gogate

Vibhav.Gogate@utdallas.edu

Student: Xiaodi Li

Net ID: XXL170011

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 2m 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 $X = \{X1, ..., Xn\}$ be a set of variables. Construct k random DAG structures over X such that the number of parents of each node is less than or equal to 3. Let Θi where $1 \leq i \leq k$ denote the parameters of the ith Bayesian network associated with the i-th DAG structure. Implement the EM algorithm for learning the parameters of the following mixture model:
$$P(X) = \sum_{i=1}^{k} p_i P_i(X; \Theta_i)$$
where $\sum_{i=1}^{k} p_i = 1$, $p_i \geq 0\ \forall i$ and $P_i(X; \Theta_i)$ is the probability distribution associated with the i-th Bayesian network.
Note that the parameters of the mxiture model are $p_1, ..., p_k$ and $\Theta_1, ..., \Theta_k$. 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 B_o and B_l denote the original Bayesian network and the learned model respectively. Let $D = (x[1], ..., x[M])$ denote the test data set. Let $LL(B, x[i])$ denote the log-likelihood of $x[i]$ w.r.t. B. Then,
$$LLDiff = \sum_{i=1}^{M} |LL(B_o, x[i]) - LL(B_l, x[i])| \quad (1)$$
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 $\sum_x P(X = x|U = u) = 1$).

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.

Answer:

Note: All the logs are log10.

dataset1

Algorithm	LLDiff Train-1	LLDiff Train-2	LLDiff Train-3	LLDiff Train-4
FOD-learn	23069682.00578299	2322996.074447176	120196.79625885532	18822.41578725506
POD-EM-learn	mean: 37098367.16303779 standard deviation: 3857341.5426118365	mean: 8699633.008247854 standard deviation: 770802.5558264649	mean: 379027.9363915672 standard deviation: 7243.377191790823	mean: standard deviation:
Mixture-Random-Bayes (k=2)	mean: 86039740.95001993 standard deviation: 0.0	mean: 24518096.04314206 standard deviation: 1.5043125931271424e-05	mean: 60904816.50416414 standard deviation: 205562.7860439612	
Mixture-Random-Bayes (k=4)	mean: 290458746.9114776 standard deviation: 18175306.20497149	mean: 91412822.4241651 standard deviation: 3.1923796551019783	mean: 234407362.07411736 standard deviation: 8649521.178862296	
Mixture-Random-Bayes (k=6)	mean: 364386300.58930314 standard deviation: 16819409.607994303			

dataset2

Algorithm	LLDiff Train-1	LLDiff Train-2	LLDiff Train-3	LLDiff Train-4
FOD-learn	14061419.274846999	7803085.87207597	2078172.3654583923	284486.77927069925

POD-EM-learn	mean: 12829523.522411305 standard deviation: 8447.36789209283	mean: 5455494.980263961 standard deviation: 92987.28168346871	mean: standard deviation:	mean: standard deviation:
Mixture-Random-Bayes (k=2)	mean: 104339388.10217151 standard deviation: 8044994.355028519			
Mixture-Random-Bayes (k=4)				
Mixture-Random-Bayes (k=6)				

dataset3

Algorithm	LLDiff Train-1	LLDiff Train-2	LLDiff Train-3	LLDiff Train-4
FOD-learn	35661370.486773856	1160057.8778899526	60938.212338804726	17527.212980524022
POD-EM-learn	mean: 45197451.46271171 standard deviation: 1257633.3738678917	mean: 4912657.381900059 standard deviation: 280496.12639460404	mean: 267506.8963200842 standard deviation: 12395.575964540953	mean: standard deviation:
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).

   Answer:

The missing data will make LLDiff higher, which means the missing data will increase the error.

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?

   Answer:

   Not knowing the precise Bayesian network structure will make LLDiff higher, which means it will increase the error. When k increases, the accuracy of the learned model will decrease as the variance becomes higher.

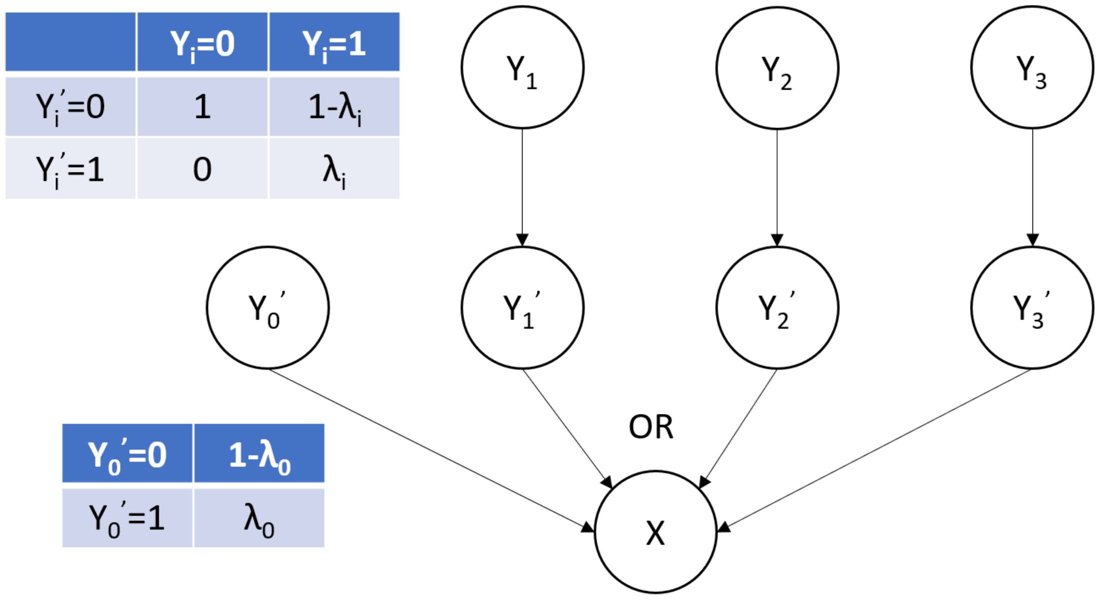
EM algorithm (20 points) KF: Koller & Friedman book

- (10 points) KF Exercise 19.4

Consider the problem of applying EM to parameter estimation for a variable X whose local probabilistic model is a noisy-or. Assume that X has parents $Y_1, \ldots, Y_k$, so that our task for X is to estimate the noise parameters $\lambda_0, \ldots, \lambda_k$. Explain how we can use the EM algorithm to accomplish this task. (Hint: Utilize the structural decomposition of the noisy-or node.)

Answer:

First, we can draw the noisy-or node in the figure (table CPT) as follows:



	Y_i=0	Y_i=1
Y_i'=0	1	1-λ_i
Y_i'=1	0	λ_i

Y_0'=0	1-λ_0
Y_0'=1	λ_0

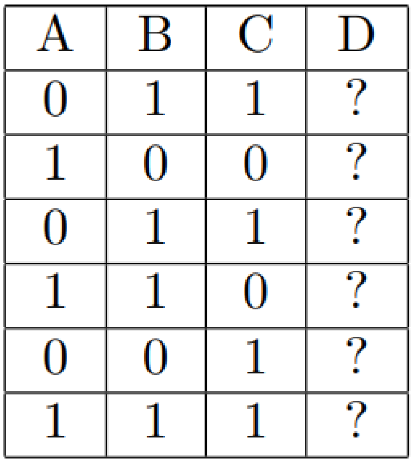
X's CPT is fixed in this case because X's CPT becomes a deterministic OR. Here, Y_i' can be considered as unobserved variables. We only need to estimate the noise parameters λ_i in the CPTs for Y_i' and λ_0 in the CPTs for Y_0'. $\lambda_i = \theta_{Y_i'=1|Y_i=1} = P(Y_i' = 1|Y_i = 1)$ and $\lambda_0 = \theta_{Y_0'=1} = P(Y_0' = 1)$. $\theta_{Y_i'=0|Y_i=0} = 1$, $\theta_{Y_i'=1|Y_i=0} = 0$.

Thus, the parameters we update when we apply EM algorithm are as follows:

$$\lambda_i = \frac{\sum_{j=1}^{M} P(Y_i' = 1, Y_i = 1 | d[j], \theta_{old})}{\sum_{j=1}^{M} P(Y_i = 1 | d[j], \theta_{old})}$$

$$\lambda_0 = \frac{\sum_{j=1}^{M} P(Y_0' = 1 | d[j], \theta_{old})}{M}$$

- (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 $D \to A$, $D \to B$ and $D \to C$). 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?

Answer:

Initialization:

$$\theta_D = 0.5, \theta_{A|D} = 0.5, \theta_{A|\bar{D}} = 0.5, \theta_{B|D} = 0.5,\ \theta_{B|\bar{D}} = 0.5, \theta_{C|D} = 0.5, \theta_{C|\bar{D}} = 0.5$$

Iteration 1:

E-step:

A	B	C	D	weight	normalized weight
0	1	1	0	0.0625	0.5
0	1	1	1	0.0625	0.5
1	0	0	0	0.0625	0.5
1	0	0	1	0.0625	0.5
0	1	1	0	0.0625	0.5
0	1	1	1	0.0625	0.5
1	1	0	0	0.0625	0.5
1	1	0	1	0.0625	0.5
0	0	1	0	0.0625	0.5
0	0	1	1	0.0625	0.5

1	1	1	0	0.0625	0.5
1	1	1	1	0.0625	0.5

M-step:

$$\theta_D = 0.5, \theta_{A|D} = 0.5, \theta_{A|\bar{D}} = 0.5, \theta_{B|D} = 0.6667,\ \theta_{B|\bar{D}} = 0.6667, \theta_{C|D} = 0.6667, \theta_{C|\bar{D}} = 0.6667$$

Iteration 2:

E-step:

A	B	C	D	weight	normalized weight
0	1	1	0	0.1111	0.5
0	1	1	1	0.1111	0.5
1	0	0	0	0.0278	0.5
1	0	0	1	0.0278	0.5
0	1	1	0	0.1111	0.5
0	1	1	1	0.1111	0.5
1	1	0	0	0.0556	0.5
1	1	0	1	0.0556	0.5
0	0	1	0	0.0556	0.5
0	0	1	1	0.0556	0.5
1	1	1	0	0.1111	0.5
1	1	1	1	0.1111	0.5

M-step:

$$\theta_D = 0.5, \theta_{A|D} = 0.5, \theta_{A|\bar{D}} = 0.5, \theta_{B|D} = 0.6667,\ \theta_{B|\bar{D}} = 0.6667, \theta_{C|D} = 0.6667, \theta_{C|\bar{D}} = 0.6667$$

Converge, stop.

The EM algorithm converges and after 2 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?

Answer:

- At convergence, what will be the parameters of the Naive Bayes model?

$$\theta_D = 0.5, \theta_{A|D} = \frac{\#(A,D)}{\#(D)}, \theta_{A|\bar{D}} = \frac{\#(A,\bar{D})}{\#(\bar{D})}, \theta_{B|D} = \frac{\#(B,D)}{\#(D)},\ \theta_{B|\bar{D}} = \frac{\#(B,\bar{D})}{\#(\bar{D})}, \theta_{C|D} = \frac{\#(C,D)}{\#(D)}, \theta_{C|\bar{D}} = \frac{\#(C,\bar{D})}{\#(\bar{D})}$$

- After how many iterations will EM converge?

  2 iterations.