COL 776: Assignment 3

Due: Tuesday Nov 15, 11:50 pm. Total Points: 65

Notes:

- You should submit all your code as well as any graphs that you might plot (see below).
- Include a **single write-up (pdf) file** which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- You can use any programming language from the set C++, Java, Python, Matlab. If you would like to use any other language, please check with us before you start.
- Your code should have appropriate documentation for readability.
- You will be graded based on what you submit as well as your ability to explain your code.
- Refer to the course website for assignment submission instructions.
- This assignment is supposed to be done individually. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. Any cheating will result in a 0 on the assignment. Additional penalties will be incurreed depending on the scale of cheating (going all the way up to a penalty of **-10**). More serious offenses will be referred to the Department's internal committee for disciplinary actions.

1. **Gibbs Sampling for Graphical Models: (20 points)**
   The problem setting below is the same as in Assignment 2. The datasets are also identical to the ones used in that assignment. You will now implement Gibbs sampling to perform inference over the OCR graphical model.

   - **Implementation (12 points)**: Implement the Gibbs sampler over the Markov network structure using all the four factors as described in Assignment 2. Recall that a Gibbs sampler visits each node in turn, sampling it based on its conditional probability given other variables in the network. Do not forget to ignore the intial burn-in samples. You should decide an appropriate convergence threshold. **Note: Make sure that you implement your Gibbs sampler in a generic manner. For the remaining problems in this assignment, we will use the same sampler for inference/learning in a completely different graphical model.**
   - **Experiments: (8 points)** Run your Gibbs sampler for each of the four datasets (tree,treeWS,loopy,loopyWS) and report Ch-Acc, Wd-Acc, LL and Time in the following tabular format (one table for each dataset). Comment on your results. Also, in the table below reproduce the numbers that you obtained for loopy BP from Assignment 2 (for the case of marginal inference). How do the two algorithms compare with each other?

Algorithm	Ch-Acc	Wd-Acc	LL	Time (seconds)
Gibbs				
Loopy BP				

2. **(45 points) Parameter Learning in Probabilistic Graphical Models**
   In the previous assignments, we have studied the problem of inference in graphical models with the parameters of the model pre-specified i.e., CPTs for Bayesian networks or potentials for Markov networks. In this problem, we will look at the task of parameter learning in graphical models. Given some i.i.d. data sampled from the underlying distribution, we will first learn the parameters of the model by maximizing

(regularized) log-likelihood and then perform inference over the test examples to estimate the efficacy of the learned parameters.

We will experiment with three different datasets publicly available from the online Bayesian network repository [1]: (1) Andes: On-Line Student Modeling for Coached Problem Solving (2) Hepar II: Diagnosis of liver disorders (3) Insurance: Car Insurance. All the datasets can be downloaded from the course website. For each of the datasets, you have been provided with a Bayesian network structure, a training file and a test file containing a set of evidence variables and the remaining (query) variables which need to be inferred given the evidence. There is also an associated readme file which gives the details of the network format, datasets, query format and the output format that your program should produce.

Bayesian Network Learning (10 points)

(a) (**5 points**) Learn the parameters of the Bayesian network given the training data for each of the dataests. Output your learned parameters in the tabular format (see readme file for details). Make sure to perform Laplace smoothing with $\alpha = 1$.

(b) (**5 points**) Given the parameters learned in the previous part, perform the inference on the test data using Gibbs sampling. Note that you need to compute the marginal probability of each of the query variables (individually) given the values of the evidence variables. Think about how you would modify your Gibbs sampler to compute the conditional probabilities efficiently (i.e., think whether you need to sample all the variables in the network). Output the probabilities inferred by your model to a file as specified in the readme.

Markov Network Learning (20 points)

(a) (**3 points**) Moralize each of the networks given in the dataset.

(b) (**12 points**) Learn the parameters of the Markov network resulting from moralization for each of the datasets. Specifically, you need to perform gradient ascent to maximize the (regularized) log-likelihood. Use a zero mean Gaussian regularizer with weight $C = 1$. Decide an appropriate learning rate η and a convergence threshold. Note that the optimal value of η may vary based on the dataset.

(c) (**5 points**) Given the parameters learned in the previous part, perform the inference on the test data using the Gibbs sampler as done previously for the Bayesian network setting. Once again, make sure to write a sampler that computes the probabilities of the query variables given the evidence efficiently. Write the inferred probabilities to a file as specified in the readme.

Evaluation Metrics: For each of the datasets, you have been provided another file which contains the true values (gold data) for the query variables. You will evaluate the performance of your learned model by examining how correctly your model can predict the true values of the query variables. Specifially, we will look at the following three metrics:

- Average Accuracy: For each of the query variables, choose the predicted value as the value having the highest marginal probability output by your Gibbs sampler. Average accuracy is defined as the fraction of the query variables whose predicted value mataches the true value.
- Average (Test) Log-likelihood: For each of the query variables, consider the probability of generating the true value given the marginal distribution output by the Gibbs sampler. Average log-likelihood (of the test data) is defined as the log of the average of the likelihoods for each of the query variables.
- AUC (Area Under Precision-Recall Curve): Read about precision-recall curves through the link given on the website. Area under this curve is another metric for evaluating the correctness of the learned model. To compute AUC, you have been provided with two scripts: (1) appendTrueVal.pl which appends the true value to the corresponding probability generated by your model (2) computeAUC.pl which takes the output of appendTrueVal.pl and computes the AUC.

Evaluation (15 points) Next, you will evaluate your models using these metrics.

(a) (**10 points**) Evaluate your models for each of the datasets using each of the metrics. You can output your results in the following tabular format.

[1] http://bnlearn.com/

Dataset	Avg-Acc	Avg-LL (test)	AUC
Andes			
Hepan II			
Insurance			

You should generate one table each for the Bayesian network and the Markov network learning. How do the results compare with other? Comment on your observations.

(b) (**5 points**) Vary the C parameter in the set $\{0.001, 0.01, 0.1, 1, 10, 100, 1000\}$ for the Markov network learning setting. Plot the average (test) log-likelihood (y-axis) as you vary the C parameter (x-axis). You may want to plot the x-axis on the log-scale. What do you observe?