COL 776: Assignment 3 (Part A)

Due: Sunday November 1, 11:50 pm

Max Points: 25 (+ 5 extra credit)

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 penalty of **-10** points on your total course score (in addition to a 0 on the assignment). Stricter penalties (**including a fail grade**) may follow.
- Note: The problem below has been borrowed (with minor changes) from the Graphical Models course offered by Andrew McCallum at UMass Amherst.

Gibbs Sampling for Graphical Models:
The problem setting below is the same as in Assignment 2, Part A. 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, Part A. Recall that a Gibbs sampler covers one node in turn, sampling it based on its conditional probability given other variables in the network. Further, recall that our convention, one full iteration of Gibbs sampling corresponds to sampling each variable (in turn) in the network once. Do not forget to ignore the initial burn-in samples. You should decide an appropriate convergence threshold.
- **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 (Part A). How do the two algorithms compare with each other?

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

- **Rate of Convergence (5 points)**: For the loopyWS dataset, plot the Wd-Acc score obtained as you vary the number of sampling iterations (you should ignore the burn-in samples while counting the number of iterations). Again, recall that one iteration of Gibbs sampling corresponds to one complete sweep through all the variables. What do you observe? Comment on your results.

- **Extra Credit (5 points)**: In the class, we discussed that there are alternate ways to implement a Gibbs sampler. In this part, we will implement a Gibbs sampler in the following way. Instead of cycling through each variable in turn, at each step, we will instead uniformly at random pick one variable to be flipped. After the variable is picked, we will sample its value given other variables (as in the original sampler). Then again, we pick another variable uniformly at random and the algorithm continues.

  Implement the Gibbs sampler as described above. Plot the Wd-Acc score obtained as you vary the number of Gibbs sampling iterations. How does it compare with the graph obtained for the original Gibbs sampler where we loop around each variable in turn (you may want to plot both the curves on the same graph). As before, in both the above cases, one iteration corresponds to n individual samples, n being the number of variables in the network. Comment on your observations.