CS6730 : Assignment 2

Instructor and TAs

Release: 27th Mar 2019; **Due: Apr 5th, 11.59pm**

- Submit to **GradeScope a single LaTeX-generated pdf file** containing your solutions. Please type your answers in the solutions blocks in the source LaTeX file of this assignment.
- You are encouraged to collaborate/discuss with other students on this assignment, but write your solutions/code in your own words.
- We tried our best to keep coding questions simple so that each coding question should require no more than a page of code (in R say).

---

0. [MANDATORY: TIME TO RE-SEARCH] The research paper assignment will be done collaboratively in teams of 4 students each - its deadline will be later in the course, and its evaluation will be based on a critique written by each member in his/her own words (70%) and a team presentation (30%). For now, please provide your

   (a) team members and team name, and

   (b) the single research paper your team has chosen.

   Some example papers are in the course moodle, but feel free to choose any other research paper with some component of probabilistic graphical modeling.

1. (5 points) [JUNCTION TREE GETS A TUNE-UP] Let $H = (\mathcal{X}, E)$ be a chordal Markov network, and let $T = (C, F)$ be a junction tree for H. Let β_c for $c \in C$ be a set of (locally) calibrated potentials, i.e. for all $(i, j) \in F$, we have that

$$\sum_{C_i \setminus S_{i,j}} \beta_i(C_i) = \sum_{C_j \setminus S_{i,j}} \beta_j(C_j),$$

   where $S_{i,j} = C_i \cap C_j$. Show that for all $i, k \in C$ (not necessarily neighbors in T), we have that if $a \in C_i$ and $a \in C_k$,

$$\sum_{C_i \setminus \{a\}} \beta_i(C_i) = \sum_{C_k \setminus \{a\}} \beta_k(C_k)$$

Solution: We have that

$$\sum_{C_i \setminus S_{i,j}} \beta_i(C_i) = \sum_{C_j \setminus S_{i,j}} \beta_j(C_j),$$

where $S_{i,j} = C_i \cap C_j$. If $a \in C_i$,and $a \in C_j$, we have $a \in S_{i,j}$ using Running intersection property. We observe that the both sides of the above equation only depend on $S_{i,j}$, so we can do as given below

$$\sum_{S_{i,j}\setminus\{a\}} \sum_{C_i\setminus S_{i,j}} \beta_i(C_i) = \sum_{S_{i,j}\setminus\{a\}} \sum_{C_j\setminus S_{i,j}} \beta_j(C_j),$$

which reduces to the given equation below

$$\sum_{C_i\setminus\{a\}} \beta_i(C_i) = \sum_{C_j\setminus\{a\}} \beta_j(C_j),$$

As given in question that, if $a \in C_i$ and $a \in C_k$, where i & k need not be adjacent to each other, so let n_1,n_2,n_3,...n_m be the clique nodes along the path of i, k such that each edge joining $n_t, n_{t+1} \in \mathcal{F}$. Thus we have $a \in C_{n_t} \forall t$ by **RIP** and also as we have shown above,

$$\sum_{C_{n_t}\setminus\{a\}} \beta_{n_t}(C_{n_t}) = \sum_{C_{n_{t+1}}\setminus\{a\}} \beta_{n_{t+1}}(C_{n_{t+1}})$$

Therefore we have chain of equalities for the nodes along the path joining node i and k and thus we arrive at this,

$$\sum_{C_i\setminus\{a\}} \beta_i(C_i) = \sum_{C_k\setminus\{a\}} \beta_k(C_k)$$

(Note: After a junction tree algorithm is run, we can compute $P(X_i)$ by choosing **any** clique c containing X_i and marginalizing out all other variables from β_c – this works as the β_c of different cliques c are each valid (unnormalized) marginals of the same (original) joint distribution. This question shows that this behavior also holds true for any set of general factors/potentials β_c as long as they are calibrated (including *pseudo-marginal* potentials β_c that are not marginals for any valid joint distribution over $\mathcal{X}$, and which have applications in variational inference).)

2. (10 points) [TOYING AROUND WITH MCMC::MH] Use normal distribution centered at the current state of the chain as a proposal distribution in the Metropolis-Hastings algorithm to sample from the $Gamma(\theta, 1)$ distribution when θ is a non-integer that is at least 2.

   (a) (3 points) Write down the acceptance probability (after all simplifications). What properties do you need to verify to confirm your method reaches the right stationary distribution after running for a sufficiently long time? Show your verification.

Solution: With the given proposal distribution Q as normal distribution with mean as the current variable and σ^2, P as gamma we get the given acceptance probability after substituting gamma and normal distribution for P&Q respectively.

$$A(x'|x) = \min(1, (\frac{x'}{x})^{(\theta-1)} exp(x - x')),$$

And since normal distribution is symmetric, both term cancels each other reducing to the above given equation.

We verify the detailed balance condition as follows

$$T(x'|x)P(x) = A(x'|x)Q(x'|x)P(x)$$

substituting the relevant values we get

$$\implies A(x'|x)(1/\sqrt{2\pi}\sigma)exp(-(x'-x)^2/(2\sigma^2))((x)^{\theta-1}/\Gamma(\theta))exp(-x)$$

Simplyfing both sides and leaving out the constant variables, we get

$$T(x'|x)P(x) = (x)^{\theta-1}exp(-(x'-x)^2/(2\sigma^2)exp(-x)).A(x'|x)$$

and for reverse transition as given below

$$T(x|x')P(x') = (x')^{\theta-1}exp(-(x-x')^2/(2\sigma^2)exp(-x')).A(x|x')$$

If $A(x'|x) = 1$

$$\implies (\frac{x}{x'})^{(\theta-1)}exp(x'-x)) \leqslant 1$$

$$\implies A(x|x') = (\frac{x}{x'})^{(\theta-1)}exp(x'-x))$$

We also need to consider the reverse transition for verifying the detailed balance property. So, it is sufficient if only $A(x'|x)$ considered as $A(x|x')$ is implied. So we took $A(x'|x) = 1$ in the above equation and got,

$$A(x|x') = (\frac{x}{x'})^{(\theta-1)}exp(x'-x))$$

We get equation given below when $A(x'|x) = 1$

$$T(x'|x)P(x) = (x)^{\theta-1}exp(-(x'-x)^2/(2\sigma^2)exp(-x)),$$

and,

$$T(x|x')P(x') = (x')^{\theta-1}exp(-(x-x')^2/(2\sigma^2)exp(-x'))A(x|x'),$$

$$\implies (x')^{\theta-1}exp(-(x-x')^2/(2\sigma^2)exp(-x'))\ (\frac{x}{x'})^{(\theta-1)}exp(x'-x))$$

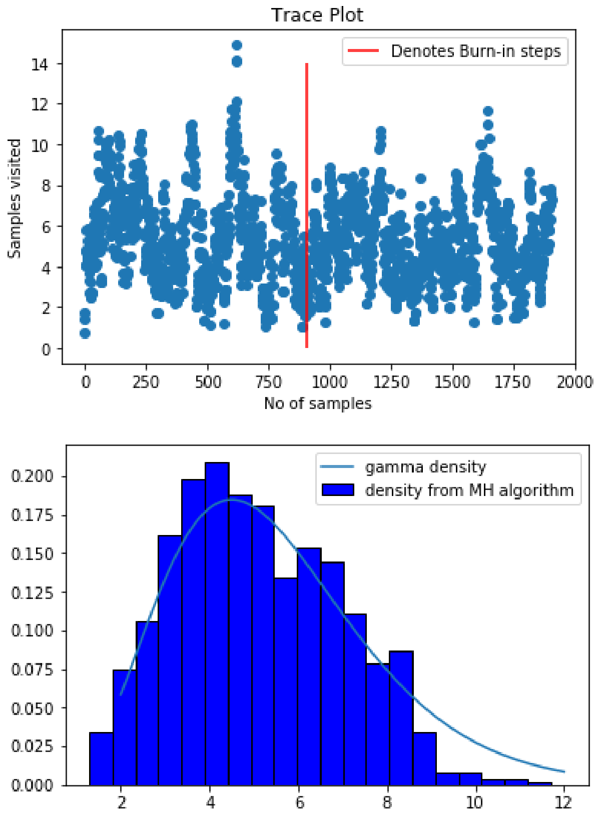
$$\implies (x)^{\theta-1}exp(-(x'-x)^2/(2\sigma^2)exp(-x)),$$

$$\implies T(x'|x)P(x)$$

Therefore, detailed balance is verified, hence the method reaches the right stationary distribution P.

(b) (4 points) Provide your code that simulates 1000 values from $\mathsf{Gamma}(5.5, 1)$, and show the trace plots, and the histogram of X_n (with the gamma density overlaid).

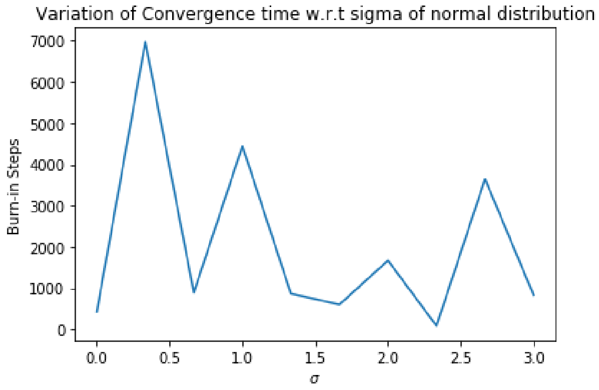
Solution: Please find my code at my github page given in the link **https://github.com/Rohithram/CS6730_Assignments/blob/master/Assignment-2/q2.ipynb**. Because I wrote my code in jupyter notebook, so couldn't attach code here for readability issues. Trace plots and histogram plots attached below.



(c) (3 points) How did you choose your burn-in time? Report it along with your acceptance rate during the burn-in vs. sample collection periods. How did these values change as a function of the variance parameter of your normal distribution, and what do you think is the optimal variance for fast convergence?

Solution: Choose burn in time by observing the convergence by plotting samples from multiple runs of the algorithm which is Trace plots. If the chains are well-mixed, they are probably converged, we check convergence of sample mean to true mean of the distribution here since it's known which is 5.5 So we find the burn-in steps till the sample mean is with tolerance range near true mean where tolerance is a hyperparameter. If true mean is unknown, we can check the successive difference between relative difference between means compared with tolerance to choose it.

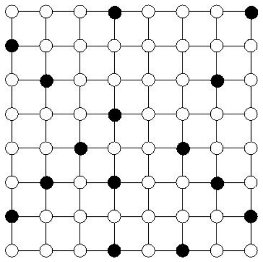
Acceptance rate during Burn-in : 0.837 and Sample Collection period: 0.853. As it is inuitive that acceptance rate once the stationary distribution is reached should be closer to 1. So acceptance rate in burn-in should be lesser than sample collection period.
As we vary the variance parameter of normal distribution, the burn-in decreases as $\sigma \uparrow$, but acceptance rate $\downarrow$. Whereas when σ is very low, the burn-in steps is high as it takes lot of time to mix well.



Thus optimal variance parameter from this graph is $\sigma = 2.33$

3. (10 points) [TOYING AROUND WITH MCMC: Consider a (non-complete) connected graph $G = (V, E)$ such as the one shown with $n = |V|$. Now each vertex in V gets either mapped to 0 or 1, where we only consider the following set $C \subset \{0, 1\}^n$ of admissible configurations characterized by the property that pairs of adjacent vertices **cannot both** take the value 1 (see figure where black denotes 1).

Now, we want to pick one of the admissible configurations $\mathbf{x} \in C$ "at random". That is, we consider the (discrete) uniform distribution π on C, i.e. $\pi_{\mathbf{x}} = \frac{1}{|C|} \ \forall \, \mathbf{x} \in C$.



(a) (3 points) Write down the edge potentials of the undirected graphical model for this problem, and a Gibbs sampling algorithm for sampling from this model (including how you derived the associated conditional $P(X_i|X_{-i})$). Does your algorithm need to know the partition function $|C|$?

Solution:
Let $\phi(X_i, X_j)$ be the edge potentials $\phi(X_i, X_j) \;=\; \{0 \text{ if } X_i = X_j = 1 \;\; \& \; 1 \;\text{ for all other values}\}$
Gibbs Sampling Algorithm:

- Initialize X_0
- for i=0 to i=T-1: including burn-in steps
  - choose a random node in grid denoted by j such that $0 < j < n^2$:
  - check only the neighbours of node j, no of neighbors vary depending on the spatial location of the node j
  - if even one of them is 1, $X_i[j] = 0$ with certainty.
  - else $X_i[j] = 0$ or $X_i[j] = 1$ with probability 0.5 each.

We can observe that at a given vertex in a cycle, the algorithm depends only on its neighbours and not on all the other nodes. Therefore, it is independent of the partition function $|C|$.

(b) (3 points) Is the Markov chain you set up irreducible and aperiodic? Does your chain admit a distribution that satisfies detailed balance? If it simplifies your proof, assume here that your Gibbs sampling routine employs "random scan" (pick a random i from $1, \ldots, n$ and then make a move based on $P(X_i|X_{-i})$ in each epoch) instead of "systematic scan" (cycle through all i from 1 to n in each epoch).

Solution: We consider the detailed balance equation, $T(x'|x)P(x) = T(x|x')P(x')$ if x is feasible and x' is infeasible and then $P(x^{'}) = 0$. Consider node value changes from 0 to 1 in a vertex x such that atleast one of its neighbours is 1. By our CPD, such an update has a probability of 0. Hence, LHS and RHS both take value 0.

Also if both x and x' are infeasible, then $P(x) = P(x') = 0$ such that detailed balance is satisfied.

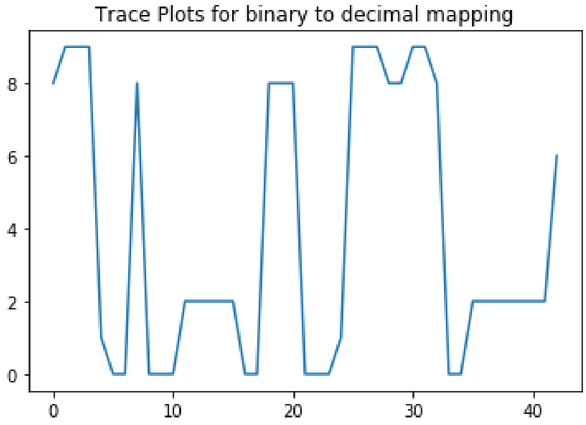
We have $P(x') = P(x)$ in the case of both x and x' are feasible. Consider a vertex which has one of its neighbours taking 1, then $x' = x$ and $T(x'|x) = 1$. Also, the reverse update occurs too with probability 1if the same vertex is considered ie. $T(x|x') = 1$. Thus detailed balance equation is satisfied. Let us now consider a vertex where its neighbours are all 0. Then T(x' | x)=0.5. Similarly as discussed above we have T(x | x')=0. Again, here we have both left and right sides of detailed balance equation are equal.

Hence, detailed balance is verified and thus, irreducibility and aperiodicity of the MC has been verified.

(c) (4 points) Provide your code and trace plots (of some functions that each map a configuration to a real value that helps visualize how well the chain is mixing). Plot the burn-in time you chose as a function of the size of the grid graphs you used. We didn't ask about the empirical acceptance rate here as it is always 100% for Gibbs sampling - prove that it is so under the same "random scan" epoch assumption above.

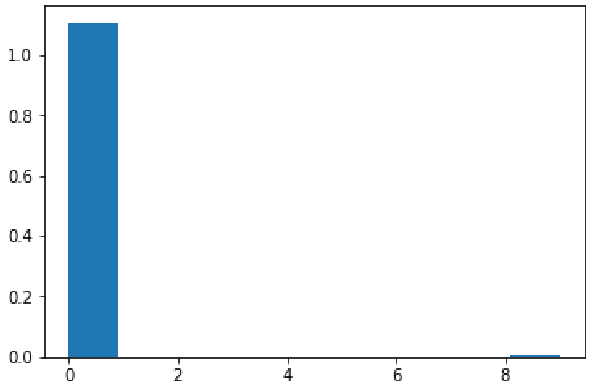
Solution: Please find my code at my github page given in the https://github.com/Rohithram/CS6730_Assignments/blob/master/Assignment-2/q3.ipynb. Because I wrote my code in jupyter notebook, so couldnâĂŹt attach code here for readability issues.

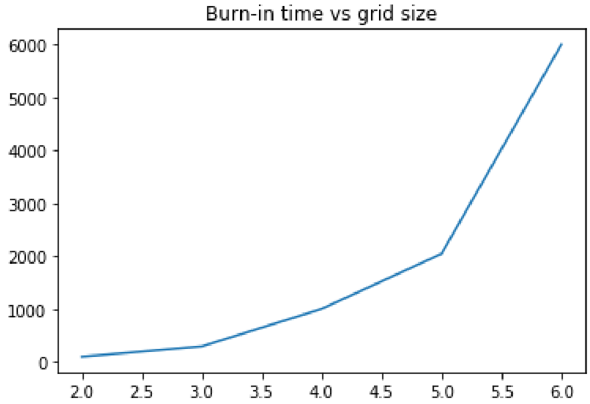
Binary sequence to decimal value mapping function is used to assign a value to each unique configuration. Decimal equivalent of sequence of 0 or $1'$s. Trace plot obtained



We observe that for $n = 2$ there total 7 valid configurations, so when we plot histogram it should give a uniform distribution over these configurations.

Which is illustrated as given below: is a Histogram of mapped values of sampled configurations





We observe that the burn-in period increases with the grid-size as expected.

4. (10 points) [BLOCK NOW, COLLAPSE LATER] Under some conditions, it is useful to partition the random variables into "blocks", and consider these blocks as "super" random variables to do Gibbs sampling on. The same logic for Gibbs sampling applies here as well to show that this also has the stationary distribution equal to P. Sometimes implementing a block Gibbs sampling step takes strictly more computation than standard Gibbs sampling, but it helps by aiding a faster convergence to the stationary distribution.

   (a) (3 points) For Model (a) given below, what is the computational complexity (give an upper bound) of one complete cycle of Gibbs sampling, and of one complete cycle of block Gibbs sampling with $\frac{n}{k}$ blocks of k variables each? Assume "systematic scan" Gibbs sampling (see previous question) is done in each epoch.

   Model (a): Let Φ be a set of factors over $X = (X_1, \ldots, X_n)$. Let
   $P(X) = \frac{1}{Z} \prod_{\phi \in \Phi} \phi(D_\phi)$,
   where D_ϕ is the scope of factor ϕ. Let each random variable X_i take values in $\{0, 1, \ldots, c-1\}$. Let each variable X_i occur in at most b factors. Let cardinality of D_ϕ be upper bounded by a for any factor $\phi \in \Phi$.

   (b) (7 points) Try both Gibbs and block Gibbs sampling approaches for Model (b) given below (group the strongly coupled random variables X_1, X_2 into a block and X_3, X_4 into another block). Provide code, and show how long your code took to reach stationary distribution with or without blocking? Provide intuition on why sampling from Model (b) with or without blocking resulted in different convergence rates.

   Model (b): Consider four random variables with distribution
   $P(X_1, X_2, X_3, X_4) = \frac{1}{Z}\psi(X_1, X_2)\psi(X_3, X_4)\phi(X_2, X_3)$,

where $\psi = \begin{bmatrix} 100 & 1 \\ 1 & 100 \end{bmatrix}$ and $\phi = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$.

(Note: Block Gibbs sampling is different from collapsed Gibbs sampling, where certain variables are integrated out. We may see collapsed Gibbs sampling in a later assignment/tutorial.)

Solution: Please find the code at my github page given in the link **https://github.com/Rohithram/CS6730_Assignments/blob/master/Assignment-2/q4.ipynb**.

Since there is high correlation between variables X_1, X_2 and X_3, X_4 as implied by the factors given. So block gibbs sampling converges faster than normal gibbs sampling in this case, the code took around 25 burn-in steps when normal gibbs sampling used and 12 burn-in steps when block gibbs sampling used, which confirms our reasoning.