ECSE 6810 Project-2 Report
Shiuli Subhra Ghosh

1 Introduction

In this project, we are given a medical application of Bayesian Networks. The CHILD network is designed for diagnosing congenital heart disease in a new born "blue" baby. Clinical expertise and available data, which is our evidence nodes here, are combined to diagnose the diseases. Fig. 1 is taken from [3] and shows the Bayesian Network represantation of our problem. CPT's are taken from the provided link [1].

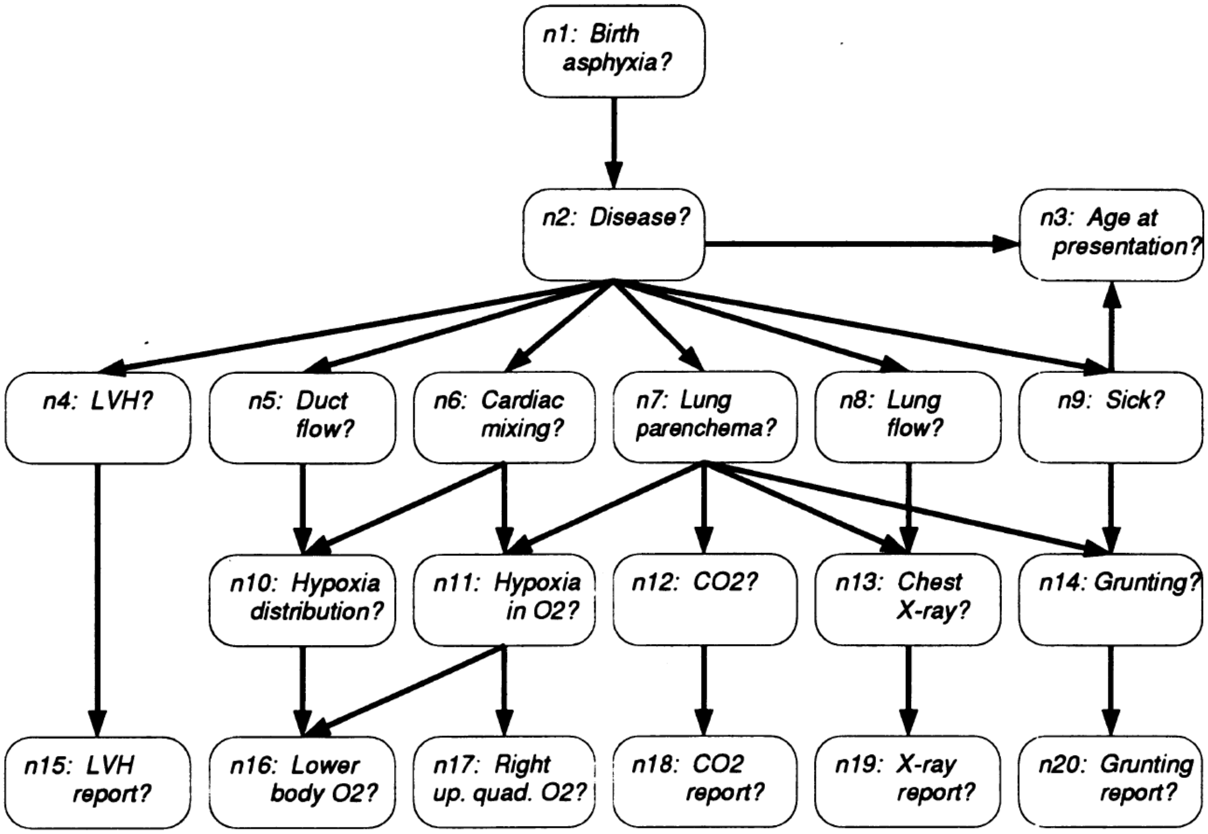

Figure 1: DAG representing the incidence and presentation of 6 possible diseases that would lead to a "blue" baby.

Exact inference methods such as variable elimination, belief propogation or junction trees are not feasible in terms of computational complexity. Therefore we use approximate inference methods. Here in this project I have used, Likelihood Weighted Sampling, Gibbs Sampling and Variational Mean Field method.

2 Problem Statement

3 Likelihood Weighted Sampling

By following the topological order of the BN, the sampler always first visits the parents of a node before visiting the node itself. In the Weighted Likelihood Sampling, maintaining the topological ordering, the sampling is done for all the non evidence nodes and the weights of the samples are updated based on the evidence nodes given the parents of the evidence nodes sampled earlier in the topological ordering. Given infinite samples, the method converges.

3.1 Algorithm

Algorithm 1 Weighted Likelihood Sampling

```
E : Evidence Nodes
Ordering BN variables $X_1, X_2, ..., X_N$ according to their topological ordering from the root nodes ultil leaf nodes
Initiliaze weights $w_1, w_2, ..., w_T$ to 1
for t = 1 to T do
    t : index to the number of samples
    for n = 1 to N do
        n : index to the node number
        if $X_n^t \notin E$ then
            sample $x_n^t$ from $p(X_n^t|\pi(X_n^t))$
        else
            $x_n^t = e_n$
            $w_t = w_t \times p(X_n^t = e_n|\pi(X_n^t))$
    From sample $x^t = \{x_1^t, x_2^t, ..., x_N^t\}$ compute its weight $w_t$
return $(x^1, w_1), (x^2, w_2), ..., (x^T, w_T)$
```

3.2 Limitations

1. Although likelihood weighted sampling helps in the case with evidences, it introduces bias in estimation due to the weights.

2. In some case, such as when the evidence is on leaves, it is inefficient.

3. If P(e) is small, we need many samples to get a decent estimate.

3.3 Results

run.py file is the main driver file for this class project. By varying the values of T we have run inference for different methods. For this Likelihood Weighted Sampling method, we have taken T = 50000.

3.3.1 Posterior Probability Inference

The result is as follows,
$P(BirthAsphyxia = yes|CO2Report =< 7.5, LVHreport = yes, XrayReport = Plethoric) = 0.08266$
$p(X_0 = 0|X_{17} = 0, X_{14} = 0, X_{18} = 2) = 0.08266$
$P(BirthAsphyxia = no|CO2Report =< 7.5, LVHreport = yes, XrayReport = Plethoric) = 0.9177$
$p(X_0 = 1|X_{17} = 0, X_{14} = 0, X_{18} = 2) = 0.9177$

3.3.2 MAP Inference

$argmaxP(Disease|CO2Report =< 7.5, LVHreport = yes, XrayReport = Plethoric) = PAIVS$
$argmaxp(X_1|X_{17} = 0, X_{14} = 0, X_{18} = 2)$

3.4 Observations

1. Sampling order matters. Because, child nodes are sampled based on their parents. So, if the parents are not sampled earlier, then the sampling method will not be successful.

2. As this method is asymptotically exact. But the result is slightly different from Gibbs Sampling. Probably, this bias happened due to weights.

4 Gibbs Sampling

Gibbs sampling is the simplest Monte Carlo Markov Chain (MCMC) method. Its simplicity comes from Markov chain assumption. At each step, starting from the initial state, one non-evidence node X_i is randomly chosen, and CPT of that node is evaluated using Markov Blanket principle. To do so we use the fact that it can be computed using only CPT's that involve X_i and its children, the equation given below. Thanks to this property, it can be applied on a large network with different node sizes, like our case. Also, given that the chain is run long enough, it's guaranteed to converge.

$$x_i^t \sim p(X_i|MB(X_i^{t-1})) = \frac{p(X_i|\pi(X_i)\prod_{k=1}^{K} p(Y_k|\pi(Y_k)))}{\sum_{x_i} p(X_i|\pi(X_i)\prod_{k=1}^{K} p(Y_k|\pi(Y_k)))}$$

4.1 Algorithm

Algorithm 2 Single Chain Gibbs Sampling

```
Initiliaze $\mathbf{X} = \{X_1, \dots, X_N\}$ to $\mathbf{x}^0 = \{x_1, \dots, x_N\}$
Fix evidence nodes $\mathbf{E}$
t: index of sample out of total T
i : index of node out of total N
b : burn-in period
k : step size to sample after burn-in period
while $t \leq T$ do
    Randomly select an i
    Obtain $x_i^{t+1} \sim p(X_i|MB(X_i^t))$
    Form a new sample $\mathbf{x}^{t+1} = \{x_1^t x_2^t, \dots, x_i^{t+1}, \dots, x_N^t\}$
return $\mathbf{x}^{b+1}, \mathbf{x}^{b+k+1}, \mathbf{x}^{b+2k+1}, \dots, \mathbf{x}^T$
```

4.2 Limitations

1. There is no theoretical guidance as to how many samples we need burn before we can assume later samples are from the true distribution.

2. Can be slow to converge and hard to diagnose their convergence when there are extreme probabilities or when the evidence is very unlikely.

3. Several tuning parameters (burn-in time, sample skip to avoid sample correlation)

4. Difficult to tell if the chain works.

4.3 Results

If we have enough samples, initilization will not have an effect on the inference. Moreover, the effect of burn-in period becomes little with increasing sample sizes as well. Hence, the chain is iterated till T = 10000000 Samples. Then using this long chain, experiments with different burn-in periods, skip sizes and initializations are realized.

4.3.1 Posterior Probability Inference

When the chain is run for $T = 10.000.000$ steps with a random initial state, skip step is taken as $k = 50$, and burn-in period $b = 10000$ we get posterior probability inference
$P(BirthAsphyxia = yes|CO2Report =< 7.5, LVHreport = yes, XrayReport = Plethoric)$

$$p(X_0 = 0|X_{17} = 0, X_{14} = 0, X_{18} = 2) = 0.08112$$

$$p(X_0 = 1|X_{17} = 0, X_{14} = 0, X_{18} = 2) = 0.9188$$

4.3.2 MAP Inference

Disease	p%
PFC	0.0237%
TGA	0.1086%
Fallot	0.1418%
PAIVS	0.6840%
TAPVD	0.0107%
Lung	0.0313%

Table 1: Infered CPT for $p(X_1|X_{17} = 0, X_{14} = 0, X_{18} = 2)$

4.4 Analysis

In this section we will discuss about the effect of initialization, burn-in-time and skip time on the inference result.

4.4.1 Effect of Burn-in Period

Let's take a chain of length $T = 10^5$, $k = 50$, and sweep the burn-in period b at range $[0, 50000]$ with 100 steps. Notice that we will have only $(T - b)/k = 2000 - b/50$ samples to infer from. This explains the wild changes and contradicting results at Fig. 2. For larger burning time, the inference is getting worst.

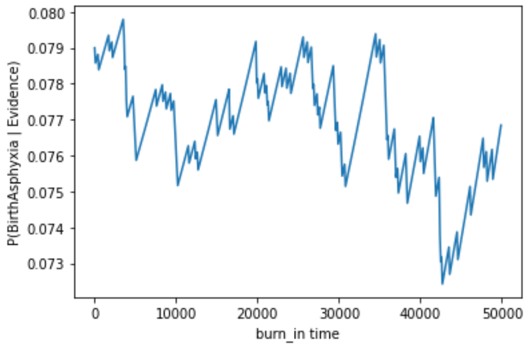

Figure 2: Posterior probability inference: Effect of burn-in period with $T = 100000$

When we repeat the same thing with $T = 1000000$, we will have $20000 - b/50$ samples to infer from. As can be seen at Fig. 3, the effect of burn-in period is negligible when we have enough sample. And results match with what we have found for Posterior probability inference.
The same effect can be seen at MAP inference as well. The Disease value doesn't change with varying burn-in for above setup, however CPT changes and probability changes wildly with different burn-ins when we use less number of samples and remains almost same when we are using enough samples. It is found that probability of Disease = PAIVS is still too big for each case and Disease $p(X_1|X_{17} = 0, X_{14} = 0, X_{18} = 2) = 3$ does not change.

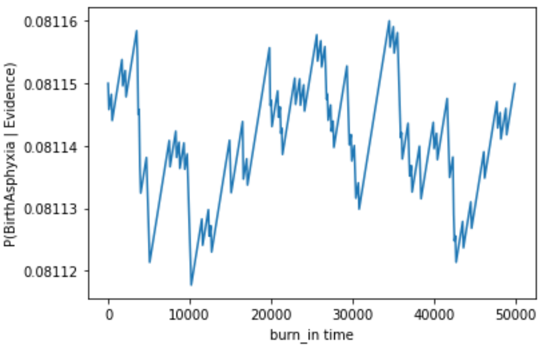

Figure 3: Posterior probability inference:Effect of burn-in period with $T = 1000000$

4.4.2 Effect of Skip-time (step_size)

Let's take a chain of length $T = 10^5$, burn_in $= 10000$, and step_size at range $[0, 100]$ with 5 steps. With lower number of samples, the inference is varying a lot with step_size. 4

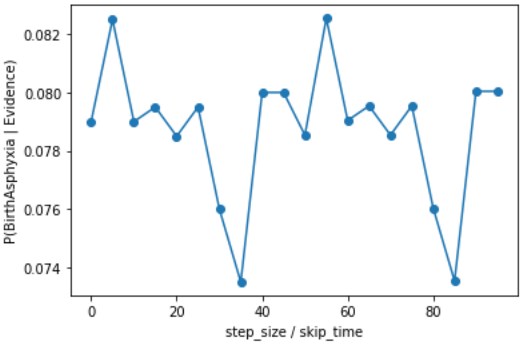

Figure 4: Posterior probability inference:Effect of step size with changing T up to 10000

When we repeat the experiment with $T = 10^7$ samples, The variation is relatively lower. 5

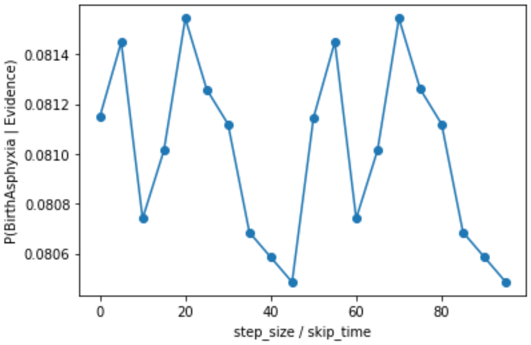

Figure 5: Posterior probability inference:Effect of step size with changing T up to 1000000

4.4.3 Effect of Initialization

We have stated that when we have enough number of samples, initialization does not matter where it does when we have small number of samples. I did run the model multiple times with random initialization. The posterior probability inference is more or less similar with 1% variation. But if we try inference with smaller number of samples, then there is significant effect on initialization. 6

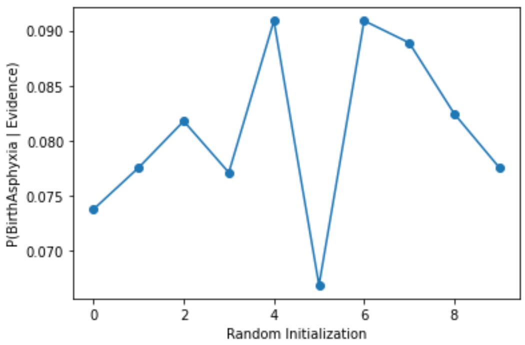

Figure 6: Posterior probability inference:Effect of initialization with changing T up to 100000

5 Variational Mean Field Inference

Variational inference is another very famous approach for approximate inference. While the sampling and other exact inference approach is really computationally expensive, this variational mean field approach can be useful for reducing the computational complexity. The main idea of this approach is to find a simple surrogate distribution $q(x|\beta)$ as an approximate of the original distribution. If we can find such q which can approximate the true distribution then we can perform inference on the surrogate distribution. The surrogate distribution often assumes independence among the target variables for tractable inference. For constructing the approximate distribution, one strategy is finding the variational parameters β to minimize the KL Divergence $KL(q||p)$. [2]

$$\beta^* = \arg\min_{\beta} KL(q(X|\beta)||p(X|e))$$

It can be expanded as,

$$\begin{aligned}\beta^* &= \arg\min_{\beta} KL(q(X|\beta)||p(X|e)) \\ &= \sum_x q(X|\beta)\log\frac{q(X|\beta)}{p(X|e)} \\ &= \sum_x q(X|\beta)\log q(X|\beta) - \sum_x q(X|\beta)\log p(X,e) + \log p(e)\end{aligned} \tag{1}$$

Since, $p(e)$ is constant, minimizing $KL(q||p)$ is equivalent to minimizing,

$$F(\beta) = \sum_x q(X|\beta)\log q(X|\beta) - \sum_x q(X|\beta)\log p(X,e)$$

$-F(\beta)$ is called the variational evidence lower bound (ELBO).
Different q needs to be calculated for each evidence instant set e. The choice of the function q is the key to the variational approach. As we are considering mean field method, we will choose the q to be fully factorized. Hence,

$$q(X) = \prod_{n=1}^{N} q(X_n|\beta)$$

Putting $q(X)$ on $F(\beta)$ we get,

$$
\begin{aligned}
F(\beta) &= \sum_{x_n \in X} \sum_{x_n} q(x_n|\beta_n) \log q(x_n|\beta_n) - \sum_{X} [\prod_{x_n \in X} q(x_n|\beta_n)] \log p(x,e) \\
&= \sum_{x_n} q(x_n,\beta_n) \log q(x_n,\beta_n) + \sum_{x_j \in X \backslash x_n} \sum_{x_j} q(x_j,\beta_j) \log q(x_j,\beta_j) \\
&- \sum_{x_n} q(x_n,\beta_n) \sum_{x \backslash x_n} \prod_{x_{j \neq n} \in X} q(x_j,\beta_j) \log p(x,e) \\
&= \sum_{x_n} q(x_n,\beta_n) \log q(x_n,\beta_n) + K_{x \backslash x_n} - \sum_{x_n} q(x_n,\beta_n) \mathbb{E}_{x \backslash x_n}[\log p(x,e)] \\
&= \sum_{x_n} q(x_n,\beta_n) \log \frac{q(x_n,\beta_n)}{\exp \mathbb{E}_{x \backslash x_n}[\log p(x,e)]} + K_{x \backslash x_n}
\end{aligned} \tag{2}
$$

$K_{x\backslash x_n}$ is not a function of β_n, $F(\beta)$ is minimized when $q(x_n, \beta_n) = \exp \mathbb{E}_{x\backslash x_n}[\log p(x, e)]$. Assuming x_n has k states,

$$
\beta_{nk} = \frac{\exp \mathbb{E}_{x\backslash x_n}[\log p(x \setminus x_n, x_n = k, e)]}{\sum_{j=1}^{k} \exp \mathbb{E}_{x\backslash x_n}[\log p(x \setminus x_n, x_n = j, e)]} \tag{3}
$$

Applying BN chain rule, $\mathbb{E}_{x\backslash x_n}[\log p(x \setminus x_n, x_n = k, e)]$ can be written as,

$$
\begin{aligned}
\mathbb{E}_{x\backslash x_n}[\log p(x \setminus x_n, x_n = k, e)] &= \mathbb{E}_{x\backslash x_n}[\sum_{l=1}^{N} \log p(x_l|\pi(x_l))] \\
&= \sum_{l=1}^{N} \mathbb{E}_{x\backslash x_n}[\log p(x_l|\pi(x_l))]
\end{aligned} \tag{4}
$$

5.1 Algorithm

Algorithm 3 Mean Field Algorithm

```
Input : a BN with evidence e and query variables X
Output : mean field parameters $\beta = \{\beta_{nk}\}$ for $k = 1, 2, .., k$
Initialize the parameters $\beta_{nk}$ such that $\sum_k \beta_{nk} = 1$
while $\beta_{nk}$ not converging do
    for n = 1 to N do
        for k = 1 to $k_i$ do
            Compute $\beta_{ik}$ using 3
```

6 Limitations

1. One of the main problem with the variational approach is characterizinf their accuracy.

6.1 Results

I expect this method might give a different result than other methods because, this is a relatively different approach, where we are trying to approximate the true distribution with mean field approximation. So, this method assume certain independencies among the target variables, which might not capture the actual dependencies among the target variables. The accuracy of its solution is dependent on the closeness between p and q.

7 Conclusion

Although variational mean field approach is computationally efficient, sometimes it really fail to produce accurate inference. On the other hand, MCMC methods (Gibbs Sampling) are simple and can lead to a solution very close to the solution obtained by exact methods given a large number of samples. However, due to randomness in the procedure, it can produce variable results as shown the section 3.4 and they can take a longer time to converge. In our case, the Gibbs Sampling method took almost 15 mins to produce 10^7 samples. For Likelihood Weighted Sampling, its relatively faster, because, we don't need to compute markov blanket. But likelihood sampling produced a different result than the Gibbs Sampling method. From this project, I learned about approximate inference methods for Bayesian Network. I have a clear understanding of how to compute marginal and conditional probabilities for a BN using python.

8 Appendix

There are four .py files in which I have implemented the approximate inference methods. run.py is the driver code file, and gibbs_sampling.py and weighted_sampling.py are the packages for approximate inference. gibbs_analysis.py

References

[1] Conditional probability tables. `https://rpi.box.com/s/rpc97pl4bw36adzambfwr1vp9vo7ethn`.

[2] Q. Ji. *Probabilistic Graphical Models for Computer Vision.* Elsevier Science & Technology, 2019.

[3] David J Spiegelhalter, A Philip Dawid, Steffen L Lauritzen, and Robert G Cowell. Bayesian analysis in expert systems. *Statistical science*, pages 219–247, 1993.