CSE 250A. Assignment 3

Hao-en Sung (A53204772)
wrangle1005@gmail.com

October 17, 2016

3.1 Inference in a chain

(a) Prove that $P(X_{t+1} = j|X_1 = i) = [A^t]_{ij}$, where A^t is the t^{th} power of the matrix A. Hint: use induction.

Sol. When t = 1, according to A_{ij} definition, I have

$$A_{ij} = P(X_2 = j|X_1 = i).$$

Assume that when $t = t'$, $P(X_{t'+1}|X_1 = i) = [A^{t'}]_{i,j}$.
For $t = t' + 1$, I have

$$\begin{aligned} P(X_t' + 2 = j|X_1 = i) &= \frac{\sum_k P(X_{t'+2} = j, X_{t'+1} = k, X_1 = i)}{P(X_1 = i)} \\ &= \sum_k P(X_{t'+2} = j|X_{t'+1} = k) \cdot P(X_{t'+1} = k|X_1 = i) \\ &= \sum_k A_{kj} \cdot [A^{t'}]_{ik} = [A^{t'+1}]_{ij}. \end{aligned}$$

□

(b) Consider the computational complexity of this inference. Devise a simple algorithm, based on matrix-vector multiplication, that scales as $O(n^2 t)$

Sol. It is known that the multiplication between one vector of size n and one matrix of size $n \times n$ cost $O(n^2)$ time complexity. Thus, overall inference complexity for t matrices multiplication is $O(n^2 t)$. □

(c) Show alternatively that the inference can also be done in $O(n^3 \log_2 t)$.

Sol. It is known that the multiplication between two matrices both of size $n \times n$ cost $O(n^3)$ time-complexity. According to *Fast Exponentiation Algorithm*, one can express t in binary format and perform matrix multiplications for $O(\log t)$ times. After that, one can run vector-matrix multiplication to get the inference result. Thus, overall inference complexity is $O(n^3 \log t + n^2) = O(n^3 \log t)$. □

(d) Suppose that the transition matrix A_{ij} is sparse, with at most $m \ll n$ non-zero elements per row. Show that in this case the inference can be done in $O(mnt)$.

Sol. One can first transform the original A matrix into sparse matrix format $A^{(s)}$. Since there are at most m non-zero elements per row in $A^{(s)}$, in vector-matrix multiplication, there are at most m multiplications for each element in vector. Thus, the time complexity can be reduced to $O(mnt)$. □

(e) Show how to compute the posterior probability $P(X_1 = i|X_T = j)$ in terms of the matrix A and the prior probability $P(X_1 = i)$. Hint: use Bayes rule and your answer from part (a).

Sol. Based on the conclusion in (a), I have $P(X_T = j|X_1 = i)$. Thus, I can derive

$$\begin{aligned} P(X_1 = i|X_T = j) &= \frac{P(X_T = j, X_1 = i)}{\sum_k P(X_T = j, X_1 = k)} \\ &= \frac{P(X_T = j|X_1 = i) \cdot P(X_1 = i)}{\sum_k P(X_T = j|X_1 = k) \cdot P(X_1 = k)}. \end{aligned}$$

□

3.2 More inference in a chain

(a) Show how to compute the conditional probability $P(Y_1|X_1)$ that appears in the numerator of Bayes rule from the CPTs of the belief network.

Sol.

$$
\begin{aligned}
P(Y_1|X_1) &= \frac{P(Y_1, X_1)}{P(X_1)} \\
&= \frac{\sum_{X_0} P(X_0) \cdot P(X_1) \cdot P(Y_1|X_0, X_1)}{P(X_1)} \\
&= \sum_{X_0} P(X_0) \cdot P(Y_1|X_0, X_1)
\end{aligned}
$$

□

(b) Show how to compute the marginal probability $P(Y_1)$ that appears in the denominator of Bayes rule from the CPTs of the belief network.

Sol.

$$
P(Y_1) = \sum_{X_0, X_1} P(X_0) \cdot P(X_1) \cdot P(Y_1|X_0, X_1)
$$

□

(c) Simplify the term $P(X_n|Y_1, ..., Y_{n-1})$ that appears in the numerator of Bayes rule.

Sol. Since Y_n is not given, X_n is independent to $Y_1, ..., Y_{n-1}$. Thus, $P(X_n|Y_1, ..., Y_{n-1}) = P(X_n)$. □

(d) Show how to compute the conditional probability $P(Y_n|X_n, Y_1, ..., Y_{n-1})$ that appears in the numerator of Bayes rule. Express your answer in terms of the CPTs of the belief network and the probabilities $P(X_{n-1} = x|Y_1, ..., Y_{n-1})$, which you may assume have already been computed.

Sol. Similar to the procedure in (a), I have

$$
\begin{aligned}
P(Y_n|X_n, Y_1, ..., Y_{n-1}) &= \sum_{X_{n-1}} P(X_{n-1}|Y_1, ..., Y_{n-1}) \cdot P(Y_n|X_{n-1}, X_n, Y_1, ..., Y_{n-1}) \\
&= \sum_{X_{n-1}} P(X_{n-1}|Y_1, ..., Y_{n-1}) \cdot P(Y_n|X_{n-1}, X_n).
\end{aligned}
$$

□

(e) Show how to compute the conditional probability $P(Y_n|Y_1, ..., Y_{n-1})$ that appears in the denominator of Bayes rule. Express your answer in terms of the CPTs of the belief network and the probabilities $P(X_{n-1} = x|Y_1, ..., Y_{n-1})$, which you may assume have already been computed.

Sol. Similar to procedure in (b), I have

$$
\begin{aligned}
P(Y_n|Y_1, ..., Y_{n-1}) &= \sum_{X_{n-1}, X_n} P(X_{n-1}|Y_1, ..., Y_{n-1}) \cdot P(X_n|Y_1, ..., Y_{n-1}) \cdot P(Y_n|X_{n-1}, X_n, Y_n, ..., Y_{n-1}) \\
&= \sum_{X_{n-1}, X_n} P(X_{n-1}|Y_1, ..., Y_{n-1}) \cdot P(X_n) \cdot P(Y_n|X_{n-1}, X_n)
\end{aligned}
$$

□

3.3 Node clustering and polytrees

Sol. Since the definition of polytree is same as normal tree — without loop in undirected graph, only three figures meet that requirement. For the rest two figures, one possible solution is to merge two marked nodes as shown in Fig. 1.

□

3.4 Cutsets and polytrees

Sol. Based on d-separate definition, I can correctly add the bounding boxes as shown in Fig. 2. Only two marked points in the bottom-figure need to be merged for polytree algorithm.

□

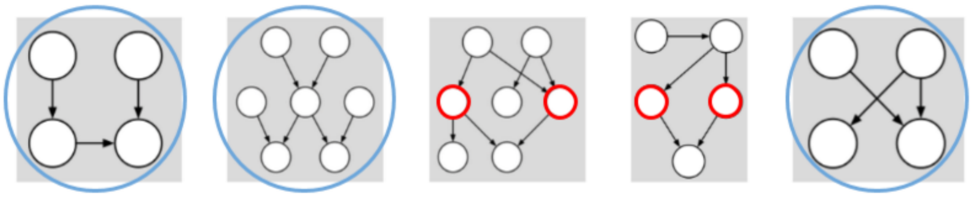


Figure 1: Node Clustering and Polytrees

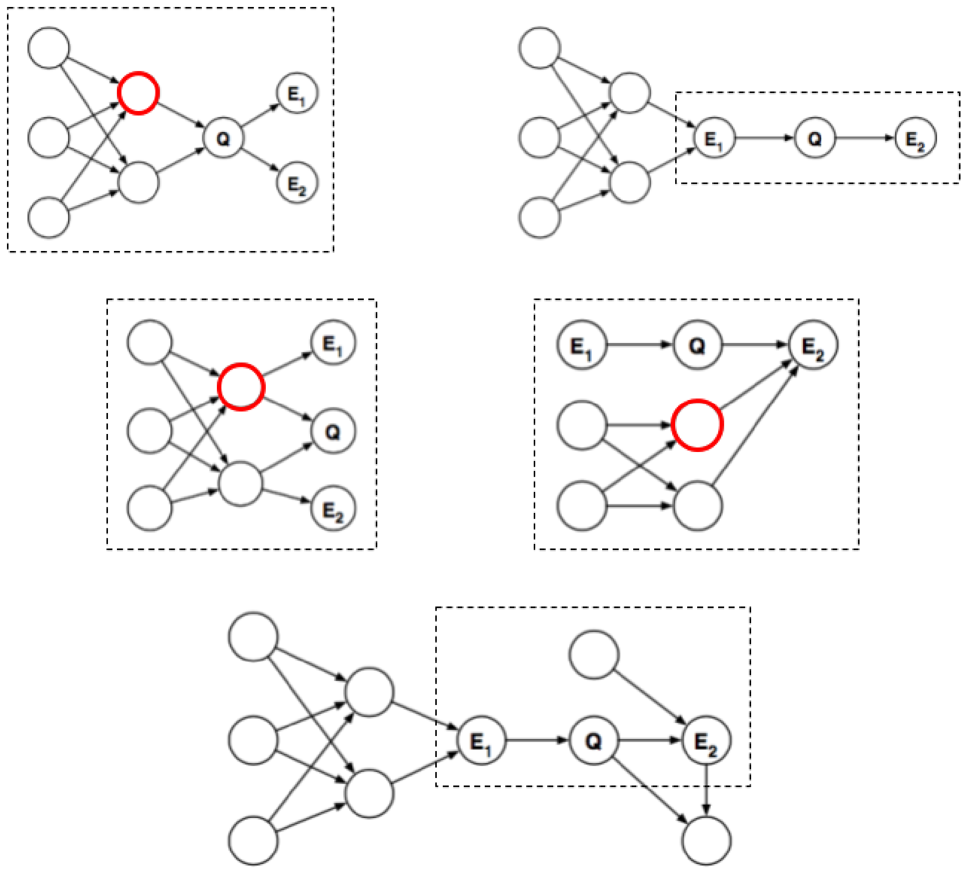


Figure 2: Cutsets and Polytrees

3.5 Node clustering

Sol. For $P(Y|X=0)$ and $P(Y|X=1)$, it is clear that they can fulfill the d-separate property since X is given, as shown in Table 1.

For $P(Y|X=1)$ and $P(Z_1=1|Y)$, they are the same as in CPT.

Y_1	Y_2	Y_3	Y	$P(Y\|X=0)$	$P(Y\|X=1)$	$P(Z_1=1\|Y)$	$P(Z_2=1\|Y)$
0	0	0	1	0.504	0.048	0.2	0.9
1	0	0	2	0.056	0.192	0.3	0.8
0	1	0	3	0.126	0.072	0.4	0.7
0	0	1	4	0.014	0.288	0.5	0.6
1	1	0	5	0.216	0.032	0.6	0.5
1	0	1	6	0.024	0.128	0.7	0.4
0	1	1	7	0.054	0.048	0.8	0.3
1	1	1	8	0.006	0.192	0.9	0.2

Table 1: Node Clustering

□

3.6 Stochastic simulation

(a) Show that the conditional distribution for binary to decimal conversion is normalized; namely, that $\sum_z P(Z=z|B_1, B_2, ..., B_n) = 1$, where the sum is over all integers $z \in [-\infty, +\infty]$.

Sol. For simplicity, I regard $-\infty$ and ∞ as regular numbers in following derivation.

$$
\begin{aligned}
\sum_z P(Z=z|B_1,B_2,...,B_n) &= \frac{1-\alpha}{1+\alpha}\cdot\left(\sum_{z=-\infty}^{f(B)}\alpha^{f(B)-z}+\sum_{z=f(B)+1}^{\infty}\alpha^{z-f(B)}\right)\\
&= \frac{1-\alpha}{1+\alpha}\cdot\left(\alpha^{f(B)}\cdot\sum_{z=-\infty}^{f(B)}\alpha^{-z}+\alpha^{-f(B)}\cdot\sum_{z=f(B)+1}^{\infty}\alpha^{z}\right)\\
&= \frac{1-\alpha}{1+\alpha}\cdot\left(\alpha^{f(B)}\cdot\frac{\alpha^{-f(B)}-\alpha^{\infty+1}}{1-\alpha}+\alpha^{-f(B)}\cdot\frac{\alpha^{f(B)+1}-\alpha^{\infty+1}}{1-\alpha}\right)\\
&= \frac{1-\alpha}{1+\alpha}\cdot\frac{1+\alpha}{1-\alpha}=1
\end{aligned}
$$

□

(b) Consider a network with $n = 10$ bits and noise level $\alpha = 0.2$. Use the method of likelihood weighting to estimate the probability $P(B_i = 1|Z = 128)$ for $i \in \{2, 4, 6, 8, 10\}$.

Sol. I implement *likelihood weighting* within MATLAB as shown in 3. I run $1,000,000$ times random sampling to get averaged results. For easily reference, I also include real probability in brackets.

$$
\begin{aligned}
P(B_2 = 1|Z = 128) &\approx 0.1895\ (0.1923)\\
P(B_4 = 1|Z = 128) &\approx 0.1668\ (0.1667)\\
P(B_6 = 1|Z = 128) &\approx 0.1584\ (0.1667)\\
P(B_8 = 1|Z = 128) &\approx 0.8326\ (0.8333)\\
P(B_{10} = 1|Z = 128) &\approx 3.2410\times 10^{-269}\ (3.2835\times 10^{-269})
\end{aligned}
$$

□

(c) Plot your estimates in part (b) as a function of the number of samples. You should be confident from the plots that your estimates have converged to a good degree of precision (say, at least two significant digits).

Sol. Same as (b), I run $1,000,000$ times random sampling to get averaged results, as shown in 3.

□

(d) Submit a hard-copy printout of your source code. You may program in the language of your choice, and you may use any program at your disposal to plot the results.

Sol. I also attach my simulated code 1 and ground truth code 1 for reference in Appendix. □

3.7 Even more inference

(a) Markov blanket

Sol.

$$
\begin{aligned}
P(B|A,C,D) &= \frac{P(A,B,C,D)}{\sum_B P(A,B,C,D)}\\
&= \frac{P(B|A)\cdot P(D|B,C)}{\sum_B P(B|A)\cdot P(D|B,C)}
\end{aligned}
$$

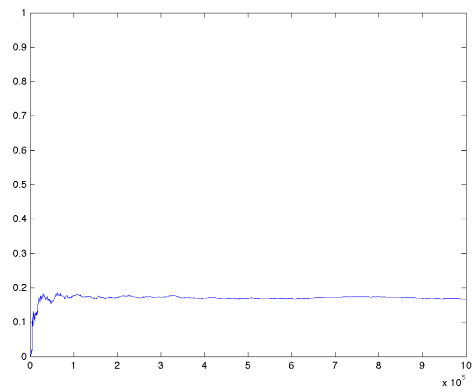
□

(b) Conditional independence

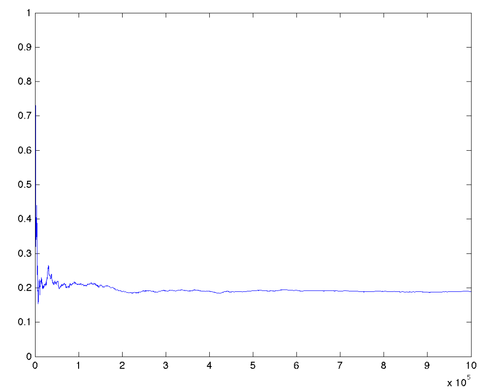
Sol.

$$
\begin{aligned}
P(B|A,C,D,E,F) &= \frac{P(A,B,C,D,E,F)}{\sum_B P(A,B,C,D,E,F)}\\
&= \frac{P(B|A)\cdot P(D|B,C)}{\sum_B P(B|A)\cdot P(D|B,C)}
\end{aligned}
$$

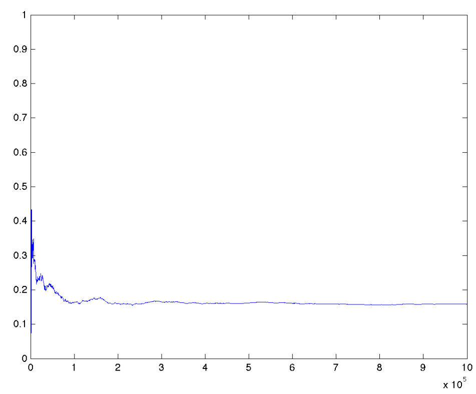
□



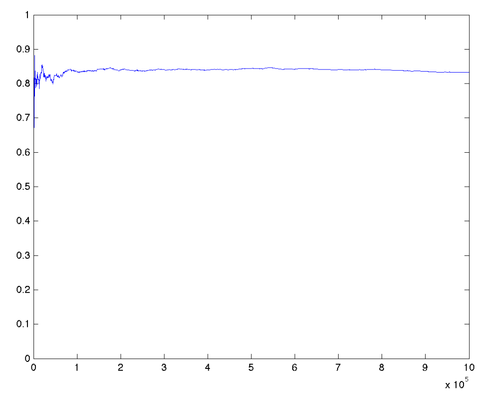
(a) $P(B_2 = 1|Z = 128) = 0.1895$



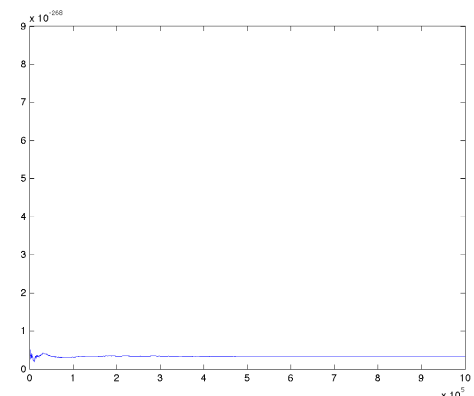
(b) $P(B_4 = 1|Z = 128) = 0.1688$



(c) $P(B_6 = 1|Z = 128) = 0.1584$



(d) $P(B_8 = 1|Z = 128) = 0.8326$



(e) $P(B_{10} = 1|Z = 128) = 3.2410 \times 10^{-269}$

Figure 3: Stochastic Simulation

(c) More conditional independence

Sol.

$$\begin{aligned} P(B,E,F|A,C,D) &= \frac{P(A,B,C,D,E,F)}{\sum_{B,E,F} P(A,B,C,D,E,F)} \\ &= \frac{P(F|A)\cdot P(B|A)\cdot P(D|B,C)\cdot P(E|C)}{\sum_{B,E,F} P(F|A)\cdot P(B|A)\cdot P(D|B,C)\cdot P(E|C)} \end{aligned}$$

□

Appendix

```
% Number of bits
NB = 10;

% Given B_2, B_4, B_6, B_8, B_10
tar = [2, 4, 6, 8, 10];

% Given evidence
z = 128;

% Alpha setting
a = 0.2;

% Number of samples
N = 1000000;

for k = 1:size(tar, 2)
    % Initialize numerator and denominator
    nm = 0;
    dn = 0;

    % Record
    rcd = zeros(1, N);

    for i = 1:N
        % Random joint distribution of B_1, ..., B_10
        t = randi(2^NB)-1;

        % Check I(q, q')
        suc = bitand(t, 2^(tar(k)-1)) ~= 0;

        % Update
        val = (1-a) / (1+a) * a^abs(z-t);
        nm = nm + suc * val;
        dn = dn + val;

        rcd(i) = nm / dn;
    end

    % Save image and ouput result
    res = figure('visible','off');
    plot(rcd);
    rcd(end)
    saveas(res, strcat('B', int2str(2*k), '.png'));
end
```

Listing 1: Simulated Code for Stochastic Simulation

```
for k in range(5):
    nm = 0
    dn = 0

    for i in range(1024):
        suc = ((i & (1 << (2*k+1))) != 0)

        val = (1-0.2) / (1+0.2) * (0.2 ** abs(128-i))
        if suc == True:
            nm = nm + val
        dn = dn + val

    print nm / dn
```

Listing 2: Ground Truth Code for Stochastic Simulation