Probabilistic Modeling and Reasoning

Homework — 4

Nikolaos Liouliakis (AID25001)
Vasileios-Efraim Tsavalia (AID25006)

MSc in Artificial Intelligence and Data Analytics
University of Macedonia

Supervisor: Professor Dimitris Christou-Varsakelis

February 2025

Problem 1

Part 1

When learning the conditional probability tables (CPTs) using maximum likelihood, we will prove in Problem 3 that it yields the empirical distribution. Therefore, after renaming the variables the CPT are:

Old variable name	**New variable name**
fuse assembly malfunction	x_0
drum unit	x_1
toner out	x_2
poor paper quality	x_3
worn roller	x_4
burning smell	x_5
poor print quality	x_6
wrinkled pages	x_7
multiple pages fed	x_8
paper jam	x_9

Table 1: Mapping of original variable names to new variable names.

$P(X_0 = 1)$
$\frac{3}{15}$

Table 2: CPT for X_0

$P(X_1 = 1)$
$\frac{4}{15}$

Table 3: CPT for X_1

$P(X_2 = 1)$
$\frac{5}{15}$

Table 4: CPT for X_2

$P(X_3 = 1)$
$\frac{8}{15}$

Table 5: CPT for X_3

$P(X_4 = 1)$
$\frac{3}{15}$

Table 6: CPT for X_4

$P(X_5 = 1 \vert X_0)$	X_0
$\frac{0}{12}$	0
$\frac{2}{3}$	1

Table 7: CPT for X_5

$P(X_6 = 1 \vert X_1, X_2, X_3)$	X_1	X_2	X_3
$\frac{0}{3}$	0	0	0
$\frac{1}{5}$	0	0	1
$\frac{2}{2}$	0	1	0
$\frac{1}{1}$	0	1	1
$\frac{1}{1}$	1	0	0
$\frac{1}{1}$	1	0	1
$\frac{1}{1}$	1	1	0
$\frac{1}{1}$	1	1	1

Table 8: CPT for X_6

$P(X_7 = 1 \vert X_0, X_3)$	X_0	X_3
$\frac{1}{5}$	0	0
$\frac{2}{7}$	0	1
$\frac{1}{2}$	1	0
$\frac{1}{1}$	1	1

Table 9: CPT for X_7

$P(X_8 = 1 \vert X_3, X_4)$	X_3	X_4
$\frac{0}{5}$	0	0
$\frac{1}{2}$	0	1
$\frac{2}{7}$	1	0
$\frac{1}{1}$	1	1

Table 10: CPT for X_8

$P(X_9 = 1 \mid X_0, X_4)$	X_0	X_4
$\frac{4}{10}$	0	0
$\frac{2}{2}$	0	1
$\frac{1}{2}$	1	0
$\frac{1}{1}$	1	1

Table 11: CPT for X_9

The CPT tables were calculated using the "printer.mat" file and the "CPT_helper.m" MATLAB script.

Part 2

The probability of a fuse assembly malfunction, given that the secretary reports a burning smell, a paper jam, and no other issues, is:

$$P(X_0 = 1 \mid X_5 = 1, X_6 = 0, X_7 = 0, X_8 = 0, X_9 = 1) = 1.0$$

To calculate this probability, the CPTs were stored in the "printer.txt" file and loaded into the python script "BN_prob_calculator.py"

Part 3

To calculate the CPTs using a Bayesian method with a uniform Beta prior $B(1, 1)$, we adjust the formula for parameter estimation by adding $a = 1$ to the numerator and adding $a + b = 2$ to the denominator.

The new CPTs are:

$P(X_0 = 1)$
$\frac{4}{17}$

Table 12: CPT for X_0

$P(X_1 = 1)$
$\frac{5}{17}$

Table 13: CPT for X_1

$P(X_2 = 1)$
$\frac{6}{17}$

Table 14: CPT for X_2

$P(X_3 = 1)$
$\frac{9}{17}$

Table 15: CPT for X_3

$P(X_4 = 1)$
$\frac{4}{17}$

Table 16: CPT for X_4

$P(X_5 = 1 \mid X_0)$	X_0
$\frac{1}{14}$	0
$\frac{3}{5}$	1

Table 17: CPT for X_5

$P(X_6 = 1 \mid X_1, X_2, X_3)$	X_1	X_2	X_3
$\frac{1}{5}$	0	0	0
$\frac{2}{7}$	0	0	1
$\frac{3}{4}$	0	1	0
$\frac{2}{3}$	0	1	1
$\frac{2}{3}$	1	0	0
$\frac{2}{3}$	1	0	1
$\frac{2}{3}$	1	1	0
$\frac{2}{3}$	1	1	1

Table 18: CPT for X_6

$P(X_7 = 1 \mid X_0, X_3)$	X_0	X_3
$\frac{2}{7}$	0	0
$\frac{3}{9}$	0	1
$\frac{2}{4}$	1	0
$\frac{2}{3}$	1	1

Table 19: CPT for X_7

$P(X_8 = 1 \mid X_3, X_4)$	X_3	X_4
$\frac{1}{7}$	0	0
$\frac{2}{4}$	0	1
$\frac{3}{9}$	1	0
$\frac{2}{3}$	1	1

Table 20: CPT for X_8

$P(X_9 = 1 \mid X_0, X_4)$	X_0	X_4
$\frac{5}{12}$	0	0
$\frac{3}{4}$	0	1
$\frac{2}{4}$	1	0
$\frac{2}{3}$	1	1

Table 21: CPT for X_9

The probability of a fuse assembly malfunction, given that the secretary reports a burning smell, a paper jam, and no other issues, is:

$$P(X_0 = 1 \mid X_5 = 1, X_6 = 0, X_7 = 0, X_8 = 0, X_9 = 1) = 0.64123752$$

To calculate this probability, the CPT tables were stored in the "printer_beta_prior.txt" file and loaded into the python script "BN_prob_calculator.py"

Part 4

$P(X_0=0,X_1=0,X_2=0,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.10704581
$P(X_0=0,X_1=0,X_2=0,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.03458403
$P(X_0=0,X_1=0,X_2=0,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.07805424
$P(X_0=0,X_1=0,X_2=0,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.02161502
$P(X_0=0,X_1=0,X_2=1,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.01824644
$P(X_0=0,X_1=0,X_2=1,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00589500
$P(X_0=0,X_1=0,X_2=1,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.01986835
$P(X_0=0,X_1=0,X_2=1,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00550200
$P(X_0=0,X_1=1,X_2=0,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.01858434
$P(X_0=0,X_1=1,X_2=0,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00600417
$P(X_0=0,X_1=1,X_2=0,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.01517721
$P(X_0=0,X_1=1,X_2=0,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00420292
$P(X_0=0,X_1=1,X_2=1,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.01013691
$P(X_0=0,X_1=1,X_2=1,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00327500
$P(X_0=0,X_1=1,X_2=1,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00827847
$P(X_0=0,X_1=1,X_2=1,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00229250
$P(X_0=1,X_1=0,X_2=0,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.23240468
$P(X_0=1,X_1=0,X_2=0,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.05561821
$P(X_0=1,X_1=0,X_2=0,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.12104412
$P(X_0=1,X_1=0,X_2=0,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.02482956
$P(X_0=1,X_1=0,X_2=1,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.03961443
$P(X_0=1,X_1=0,X_2=1,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00948037
$P(X_0=1,X_1=0,X_2=1,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.03081122
$P(X_0=1,X_1=0,X_2=1,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00632025
$P(X_0=1,X_1=1,X_2=0,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.04034803
$P(X_0=1,X_1=1,X_2=0,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00965593
$P(X_0=1,X_1=1,X_2=0,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.02353635
$P(X_0=1,X_1=1,X_2=0,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00482797
$P(X_0=1,X_1=1,X_2=1,X_3=0,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.02200801
$P(X_0=1,X_1=1,X_2=1,X_3=0,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00526687
$P(X_0=1,X_1=1,X_2=1,X_3=1,X_4=0 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.01283801
$P(X_0=1,X_1=1,X_2=1,X_3=1,X_4=1 \mid X_5=1,X_6=0,X_7=0,X_8=0,X_9=1)$	0.00263343

Table 22: All possible diagnosis given the evidence

Therefore, the most likely diagnosis is:

$$P(X_0 = 1, X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0 | X_5 = 1, X_6 = 0, X_7 = 0, X_8 = 0, X_9 = 1) = 0.232404$$

To calculate the above probabilities, the CPTs were stored in the "printer_beta_prior.txt" file and loaded into the python script "BN_prob_calculator.py"

Problem 2

1. **Fitting a Gaussian to each class:**
   Let the training examples from class 1 be:
   $$x_1 = \{0.5, 0.1, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.35, 0.25\}$$
   and from class 2:
   $$x_2 = \{0.9, 0.8, 0.75, 1.0\}$$
   To fit a Gaussian distribution, we need to estimate the mean and variance for each class using Maximum Likelihood Estimation (MLE).
   - For class 1, the mean μ_1 and variance σ_1^2 are given by:
     $$\mu_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} x_{1,i} = 0.26, \quad \sigma_1^2 = \frac{1}{n_1} \sum_{i=1}^{n_1} (x_{1,i} - \mu_1)^2 = 0.0149$$
     where $n_1 = 10$.
   - For class 2, the mean μ_2 and variance σ_2^2 are:
     $$\mu_2 = \frac{1}{n_2} \sum_{i=1}^{n_2} x_{2,i} = 0.8625, \quad \sigma_2^2 = \frac{1}{n_2} \sum_{i=1}^{n_2} (x_{2,i} - \mu_2)^2 = 0.0092$$
     where $n_2 = 4$.
2. **Estimating class probabilities:**
   The class probabilities p_1 and p_2 can be estimated by the relative frequencies of each class in the dataset:
   $$p_1 = \frac{n_1}{n_1 + n_2} = \frac{10}{14}, \quad p_2 = \frac{n_2}{n_1 + n_2} = \frac{4}{14}$$
3. **Calculating the probability that $x = 0.6$ belongs to class 1:**
   Using Bayes' theorem, we have:
   $$P(\text{class } 1 | x = 0.6) = \frac{P(x = 0.6 | \text{class } 1) \cdot p_1}{P(x = 0.6 | \text{class } 1) \cdot p_1 + P(x = 0.6 | \text{class } 2) \cdot p_2}$$
   where
   $$\begin{aligned} P(x = 0.6 | \text{class } 1) &= \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(0.6 - \mu_1)^2}{2\sigma_1^2}\right) \\ &= \frac{1}{\sqrt{2\pi \cdot 0.0149}} \exp\left(-\frac{(0.6 - 0.26)^2}{2 \cdot 0.0149}\right) \\ &= \frac{1}{\sqrt{2\pi}} \frac{1}{0.1221} 0.0206675 \\ &= \frac{1}{\sqrt{2\pi}} 0.1692669942669 \end{aligned}$$

and

$$\begin{aligned}
P(x=0.6|\text{class } 2) &= \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{(0.6-\mu_2)^2}{2\sigma_2^2}\right) \\
&= \frac{1}{\sqrt{2\pi \cdot 0.0092}} \exp\left(-\frac{(0.6-0.8625)^2}{2 \cdot 0.0092}\right) \\
&= \frac{1}{\sqrt{2\pi}} \frac{1}{0.0960} 0.0236379 \\
&= \frac{1}{\sqrt{2\pi}} 0.246228125
\end{aligned}$$

Substituting we get:

$$P(\text{class } 1|x=0.6) = \frac{\frac{1}{\sqrt{2\pi}}0.1692669942669 \cdot \frac{10}{14}}{\frac{1}{\sqrt{2\pi}}0.1692669942669 \cdot \frac{10}{14} + \frac{1}{\sqrt{2\pi}}0.246228125 \cdot \frac{4}{14}} = 0.63216$$

Problem 3

We need to show that the maximum likelihood estimate for the parameters $\theta_s^i(t_i) = p(x_i = s \mid \text{pa}(x_i) = t_i)$ is given by:

$$\theta_s^i(t_i) = \frac{\sum_{n=1}^N \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\text{pa}(x_i^n) = t_i\right]}{\sum_{n=1}^N \sum_s \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\text{pa}(x_i^n) = t_i\right]}$$

Define the Log Likelihood of the Data

The log likelihood for the dataset $\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, assuming the belief network structure and independently gathered observations, is given by:

$$\log p(\mathcal{X}) = \sum_{n=1}^N \sum_{i=1}^K \log p\left(x_i^n \mid \text{pa}(x_i^n)\right)$$

The parameter $\theta_s^i(t_i) = p(x_i = s \mid \text{pa}(x_i) = t_i)$, represents the probability that variable x_i is in state s given that its parent variables are in the state t_i.

Formulate the Lagrangian

To find the maximum likelihood estimate of $\theta_s^i(t_i)$, we impose the constraint that the probabilities for each parent state t_i sum to 1:

$$\sum_s \theta_s^i(t_i) = 1 \quad \text{for each } t_i, i$$

Using a Lagrange multiplier $\lambda_{t_i}^i$ for each constraint, we construct the Lagrangian function:

$$L = \sum_{n=1}^N \sum_{i=1}^K \log p\left(x_i^n \mid \text{pa}(x_i^n)\right) + \sum_{i=1}^K \sum_{t_i} \lambda_{t_i}^i \left(1 - \sum_s \theta_s^i(t_i)\right)$$

where:

$$p\left(x_i^n \mid \text{pa}(x_i^n)\right) = \sum_s \sum_{t_i} \theta_s^i(t_i)^{\mathbb{I}[x_i^n = s]\mathbb{I}[\text{pa}(x_i^n)]}$$

Expanding the log-probability term in terms of $\theta_s^i(t_i)$, we get:

$$L = \sum_{n=1}^{N} \sum_{i=1}^{K} \sum_{s} \sum_{t_i} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right] \log \theta_s^i(t_i) + \sum_{i=1}^{K} \sum_{t_i} \lambda_{t_i}^i \left(1 - \sum_{s} \theta_s^i(t_i)\right)$$

Differentiation with Respect to $\theta_s^i(t_i)$

To maximize L with respect to $\theta_s^i(t_i)$, we take the partial derivative of L with respect to $\theta_s^i(t_i)$ and set it to zero:

$$\frac{\partial L}{\partial \theta_s^i(t_i)} = \sum_{n=1}^{N} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right] \frac{1}{\theta_s^i(t_i)} - \lambda_{t_i}^i = 0$$

Rearranging, we obtain:

$$\theta_s^i(t_i) = \frac{\sum_{n=1}^{N} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right]}{\lambda_{t_i}^i}$$

Solving for $\lambda_{t_i}^i$ Using the Normalization Constraint

We use the constraint $\sum_s \theta_s^i(t_i) = 1$ to determine $\lambda_{t_i}^i$:

$$\sum_{s} \theta_s^i(t_i) = \sum_{s} \frac{\sum_{n=1}^{N} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right]}{\lambda_{t_i}^i} = 1$$

Therefore,

$$\begin{aligned}\lambda_{t_i}^i &= \sum_{s} \sum_{n=1}^{N} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right] \\ &= \sum_{n=1}^{N} \sum_{s} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right]\end{aligned}$$

Substituting $\lambda_{t_i}^i$ back into our expression for $\theta_s^i(t_i)$, we have proven that the maximum likelihood estimate for $\theta_s^i(t_i)$ is:

$$\theta_s^i(t_i) = \frac{\sum_{n=1}^{N} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right]}{\sum_{n=1}^{N} \sum_{s} \mathbb{I}\left[x_i^n = s\right] \mathbb{I}\left[\mathrm{pa}(x_i^n) = t_i\right]}$$

Problem 4

Part 1

To count the number of possible Belief Networks with a given ancestral order, we analyze three cases:

1. Networks with no restriction on the number of parents per node.
2. Networks where each node has at most 2 parents.
3. Networks where each node has at most k parents.

Case 1: No Restriction on the Number of Parents per Node

When there is no restriction on the number of parents, each node in a directed acyclic graph (DAG) can be connected to any subset of the previous nodes in the ancestral order. For n nodes, there are $\binom{n}{2}$ possible directed edges, so the total number of unlabeled DAGs is:

$$2^{\binom{n}{2}}$$

Case 2: Restricting Each Node to At Most 2 Parents

Let a_n denote the total number of unlabeled DAGs with n nodes, where each node has at most 2 parents. In this case, we aim to derive a_n recursively.

From observation, we have:

$$a_1 = 1 \quad \text{and} \quad a_2 = 2$$

since with one node there's only one possible graph, and with two nodes there are two possible configurations (either connected or disconnected).

For $n \geq 3$, we consider three scenarios for the new n^{th} node, corresponding to it having 0, 1, or 2 parents from the previous $n-1$ nodes:

- 0 parents: The new node is isolated, contributing a_{n-1} to the count.
- 1 parent: The new node is connected to one of the $n-1$ previous nodes, giving $(n-1) \cdot a_{n-1}$.
- 2 parents: The new node is connected to two of the previous nodes, yielding $\binom{n-1}{2} \cdot a_{n-1}$.

Thus, the recurrence relation is:

$$a_n = a_{n-1} + (n-1) \cdot a_{n-1} + \binom{n-1}{2} \cdot a_{n-1}, \quad n \geq 3$$

This recurrence has the closed-form solution:

$$a_n = 2^{1-n} \prod_{k=0}^{n-2} \left(\left(\frac{3}{2} + k \right)^2 + \frac{7}{4} \right)$$

Case 3: Restricting Each Node to At Most k Parents

For a general restriction of at most k parents per node, we define $a_{n,k}$ as the total number of unlabeled DAGs with n nodes, where each node has up to k parents. We generalize the recurrence as follows:

For $i \leq k$, the number of possible DAGs with i nodes (where each node can connect freely to all previous nodes) is:

$$a_{1,k} = 1$$

$$a_{i,k} = 2^{\binom{i}{2}} \quad k \geq i \geq 2$$

For $n > k$, we sum over the possible numbers of parents (from 0 up to k) for the new n^{th} node. Each case contributes a term $\binom{n-1}{i} \cdot a_{n-1}$, where i is the number of parents the new node has:

$$a_{n,k} = \sum_{i=0}^{k} \binom{n-1-k}{i} \cdot a_{n-1,k}, \quad n > k$$

The table below compares the total number of possible DAGs with no parent restriction and the number with at most 2 parents per node:

n	**Total DAGs (at most 2 parents)**	**Possible Edges**	**Total DAGs (no restriction)**
1	1	0	1
2	2	1	2
3	8	3	8
4	56	6	64
5	616	10	1024
6	9856	15	32768
7	216832	21	2097152
8	6288128	28	268435456
9	232660736	36	68719476736
10	10702393856	45	35184372088832

Table 23: Comparison of DAG counts with and without restrictions on parent nodes

Therefore, the number of BN in N_a are $a_{8,2} = 6288128$

Part 2

To calculate the computational time to find the optimal member of N_a where $|N_a| = a_{8,2}$ using brute force (computing the BD score for each member of N_a individually), we would need $|N_a| = 6,288,128$ seconds (given that computing the BD score of any member of N_a takes 1 second).

However, by taking advantage of the decomposability of the BD score, we can reduce the computational time significantly. Since the BD score is decomposable, we can maximize the score for each node independently by maximizing the corresponding term for each node, represented by $\prod_n p(v_k^n|\text{pa}(v_k^n))$. The BD score can be expressed as:

$$p(\mathcal{D}) = \prod_k \prod_n p(v_k^n|\text{pa}(v_k^n)) = \prod_k \prod_j \frac{Z(u'(v_k;j))}{Z(u(v_k;j))}$$

Since it takes 1 second to compute the full BD score for any network and the BD score is composed of 8 independent terms (one for each node), we can assume that calculating each term individually takes approximately $\frac{1}{8}$ seconds (the time heavily depends on the number of observations).

Next, we need to determine the number of different configurations for each node:

- For the first node, there is only 1 configuration.
- For the second node, there are 2 configurations (either no connection or connected to the first node).
- For the n-th node, there are $1 + (n-1) + \binom{n-1}{2}$ configurations: (either no connection, connected to one of the previous nodes, or connected to two of the previous nodes).

We can summarize the number of configurations for each node n as follows:

$$b_n = n + \frac{(n-1)(n-2)}{2} \quad n \geq 1$$

Thus, for each node, we need to find the configuration that maximizes its corresponding term out of the b_n configurations. Each configuration requires $\frac{1}{8}$ seconds to evaluate.

The total time to find the optimal member of N_a is therefore:

$$\sum_{n=1}^{8} \frac{1}{8} b_n = \frac{1}{8} \sum_{n=1}^{8} \left(n + \frac{(n-1)(n-2)}{2} \right) = \frac{92}{8} = 11.5 \text{ seconds}$$

This calculation demonstrates that, by leveraging decomposability, we reduce the time from millions of seconds to only 11.5 seconds.

Part 3

First, it is important to note that $|N| < 8!\,|N_a|$. For a general case without the restriction of at most 2 parents per node, the solution is not $n! \cdot 2^{\binom{n-1}{2}}$; instead, it can be found here.

One way to estimate the total time required to find the optimal member of N is to check every possible ancestral order a (of which there are 8!). Using this approach, the total time would be:

$$8! \cdot 11.5 = 463680 \text{ seconds.}$$

It would be interesting to investigate whether a more efficient approach exists to speed up the above calculation. One potential (but untested) idea could involve optimizing both the order and structure simultaneously. By carefully considering the order of operations, this might lead to:

$$\sum_{n=1}^{8} \frac{1}{8} b_n \cdot (8-n) = \frac{1}{8} \sum_{n=1}^{8} \left((8-n) \left(n + \frac{(n-1)(n-2)}{2} \right) \right) = \frac{246}{8} = 30.75 \text{ seconds.}$$

This approach is speculative, and further analysis or testing would be needed to determine whether it is correct and effective.