A Results

A.1 Evaluation setup

For loopy belief propagation (LBP) [Murphy et al., 1999], we use the implementation provided in LibDAI [Mooij, 2010, 2012]. We set the tolerance limit to $10^{-3}$ when time limit is 2 min and $10^{-9}$ for 20 min. For iterative join graph propagation (IJGP) [Mateescu et al., 2010], we used the implementation available on the author’s webpage [Gogate, 2010]. The maximum cluster size in IJGP is set using the parameter $ibound$. This solver starts with the minimal value of $ibound$ and increases it until the runtime and memory constraints are satisfied. A solution is obtained for each $ibound$. The results reported are those obtained for the largest $ibound$ possible for the given time and memory constraints. For WMB, we used the implementation made available by the authors in the Merlin tool [Marinescu, 2016]. Since this implementation uses a fixed $ibound$ value, we wrote a script to run it in anytime fashion similar to IJGP. We report results obtained with the largest value of $ibound$ possible. For sample search with IJGP-based proposal and cutset sampling (ISSwc) [Gogate and Dechter, 2011], we used the implementation provided by the authors on Github [Gogate, 2020]. For ISSwc, appropriate values of $ibound$ and $w$-cutset bound are set by the tool based on the given runtime limit.

A.2 Additional results

For a fair comparison with IBIA using $mcs_p$ of 20 (referred to as ‘IBIA20’), we also obtained the results for ISSwc after fixing both $ibound$ and $w$-cutset bound to 20 (referred to as ‘ISSwc20’). Table 1 compares the results obtained using IBIA20, ISSwc20 and ISSwc (in which the optimal $ibound$ is determined by the solver). The runtime limit was set to 2 min and 20 min, and the memory limit was set to 8 GB. The error obtained using IBIA20 is either smaller than or comparable to ISSwc20 and ISSwc for both time limits in all testcases except DBN. For DBN, in 20 min, both variants reduce to exact inference in many DBN instances and the average error obtained is close to zero.

Table 2 compares the maximum Hellinger distance obtained using IBIA ($mcs_p$=15,20) with published results for adaptive Rao Blackwellisation (ARB) and iterative join graph propagation in [Kelly et al., 2019]. The minimum error obtained is shown in bold. IBIA with $mcs_p = 20$ gives the least error in all cases. The error obtained with $mcs_p = 15$ is smaller than ARB and IJGP in all testcases except Grids_11, Grids_13 and Promedas_12.
Table 1: Comparison of average $HD_{avg}$ and average $HD_{max}$ (shown in gray background) obtained using IBIA with $mcs_p = 20$ (IBIA20). ISSwc with clique size bounds determined by the solver Cogate [2020] (ISSwc) and ISSwc with $ibound$ and $wcutset$ bound fixed to 20 (ISSwc20). Results are shown for two runtime limits, 2 min and 20 min. Entries are marked with '-' if the solution for all testcases could not be obtained within the given time and memory limits. The minimum error obtained for a benchmark is highlighted in bold. The number of instances solved by each solver is shown in the last row. $ev_a$: average number of evidence variables, $va$: average number of variables, $fa$: average number of factors, $wa$: average induced width and $dm_a$: average of the maximum variable domain size.

<table>
<thead>
<tr>
<th></th>
<th>Total #Inst $(e_v, v_a, f_a, w_a, d_m_a)$</th>
<th>2 min</th>
<th>20 min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(ISSwc)</td>
<td>ISSwc20</td>
<td>IBIA20</td>
</tr>
<tr>
<td>BN</td>
<td>97</td>
<td>-</td>
<td>0.037</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0.145</td>
</tr>
<tr>
<td>GridBN</td>
<td>29</td>
<td>0.003</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.051</td>
<td>0.065</td>
</tr>
<tr>
<td>Bnlearn</td>
<td>26</td>
<td>0.012</td>
<td>0.036</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.064</td>
<td>0.094</td>
</tr>
<tr>
<td>Pedigree</td>
<td>24</td>
<td>0.033</td>
<td>0.028</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.292</td>
<td>0.245</td>
</tr>
<tr>
<td>Promedas</td>
<td>64</td>
<td>0.030</td>
<td>0.042</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.139</td>
<td>0.207</td>
</tr>
<tr>
<td>DBN</td>
<td>36</td>
<td>0.016</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.766</td>
<td>0.833</td>
</tr>
<tr>
<td>ObjDetect</td>
<td>79</td>
<td>0.018</td>
<td>0.039</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.189</td>
<td>0.233</td>
</tr>
<tr>
<td>Grids</td>
<td>8</td>
<td>-</td>
<td>0.088</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0.300</td>
</tr>
<tr>
<td>CSP</td>
<td>12</td>
<td>-</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-</td>
<td>0.011</td>
</tr>
<tr>
<td>Segment</td>
<td>50</td>
<td>0</td>
<td>0.002</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>0.036</td>
</tr>
<tr>
<td>Protein</td>
<td>68</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.049</td>
<td>0.030</td>
</tr>
<tr>
<td>#Inst</td>
<td>493</td>
<td>485</td>
<td>488</td>
</tr>
</tbody>
</table>

Table 2: Comparison of maximum Hellinger distance ($HD_{max}$) obtained using IBIA with published results for Gibbs sampling with adaptive Rao Blackwellisation (ARB) and iterative join graph propagation in [Kelly et al., 2019]. Results obtained with $mcs_p = 15$ and $mcs_p = 20$ are shown in columns marked as IBIA15 and IBIA20 respectively. Runtime (in seconds) for IBIA15 and IBIA20 are also shown. Estimates for ARB were obtained within 600 seconds [Kelly et al., 2019] and runtime for IJGP is not reported in [Kelly et al., 2019]. The minimum error obtained for each benchmark is marked in bold. $w$: induced width, $dm$: maximum domain size.

<table>
<thead>
<tr>
<th></th>
<th>$w$</th>
<th>$dm$</th>
<th>$HD_{max}$</th>
<th>Runtime (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>IBIA15</td>
<td>IBIA20</td>
</tr>
<tr>
<td>Alchemy, 11</td>
<td>19</td>
<td>2</td>
<td>0.777</td>
<td>0.062</td>
</tr>
<tr>
<td>CSP 11</td>
<td>16</td>
<td>4</td>
<td>0.513</td>
<td>0.274</td>
</tr>
<tr>
<td>CSP 12</td>
<td>11</td>
<td>4</td>
<td>0.515</td>
<td>0.275</td>
</tr>
<tr>
<td>CSP 13</td>
<td>19</td>
<td>4</td>
<td>0.503</td>
<td>0.290</td>
</tr>
<tr>
<td>Grids 11</td>
<td>21</td>
<td>2</td>
<td>0.543</td>
<td>0.420</td>
</tr>
<tr>
<td>Grids 12</td>
<td>12</td>
<td>2</td>
<td>0.645</td>
<td>0.432</td>
</tr>
<tr>
<td>Grids 13</td>
<td>21</td>
<td>2</td>
<td>0.500</td>
<td>0.544</td>
</tr>
<tr>
<td>Pedigree 11</td>
<td>19</td>
<td>3</td>
<td>0.532</td>
<td>0.576</td>
</tr>
<tr>
<td>Pedigree 12</td>
<td>19</td>
<td>3</td>
<td>0.562</td>
<td>0.506</td>
</tr>
<tr>
<td>Pedigree 13</td>
<td>19</td>
<td>3</td>
<td>0.577</td>
<td>0.611</td>
</tr>
<tr>
<td>Promedus 11</td>
<td>18</td>
<td>2</td>
<td>1.000</td>
<td>0.373</td>
</tr>
<tr>
<td>Promedus 12</td>
<td>20</td>
<td>2</td>
<td>1.000</td>
<td>0.358</td>
</tr>
<tr>
<td>Promedus 13</td>
<td>10</td>
<td>2</td>
<td>1.000</td>
<td>0.432</td>
</tr>
</tbody>
</table>

* The results tabulated in [Kelly et al., 2019] report $-\log_2 HD_{max}$. The table above has the corresponding values of $HD_{max}$.

+ System used: Ubuntu 18.04, with 16GB of RAM, 6 CPUs and 2 hardware threads per CPU [Kelly et al., 2019].
Algorithm 1 shows the steps in the proposed algorithm for the inference of marginals. We first convert the PGM into a sequence of linked CTFs (SLCTF) that contains a sequence of calibrated CTFs \((\text{SCLT } F = \{CTF_k\})\) and a list of links between adjacent CTFs \((\text{SL } = \{L_k\})\). Functions \textit{BuildCTF} and \textit{ApproximateCTF} are used for incremental construction of CTFs and approximation of CTFs respectively. The steps in these functions are explained in detail in Algorithms 1 and 2 in Bathla and Vasudevan [2023]. Links between adjacent CTFs are found using the function \textit{FindLinks} and belief update in the SLCTF is performed using the function \textit{BeliefUpdate}. Following this, the marginal of a variable \(v\) is inferred from clique beliefs in the last CTF that contains \(v\) (line 23).

C Proofs

Notations

- \(\Phi_k\): Set of factors added to construct \(CTF_k\)
- \(X_k\): Set of all non-evidence variables in \(CTF_k\)
- \(X_{k,a}\): Set of all non-evidence variables in \(CTF_{k,a}\)
- \(Y_k\): Set of variables in \(CTF_k\) but not in \(CTF_1, \ldots, CTF_{k-1}\)
- \(Pa_{Y_k}\): Parents of variables in \(Y_k\) in the BN
- \(E_k\): Set of evidence variables in \(Y_k\)
- \(e_k\): Evidence state corresponding to variables in \(E_k\)
- \(C\): A clique in \(CTF_k\)
- \(C'\): A clique in \(CTF_{k,a}\)
- \(SP\): Sepset associated with an edge in \(CTF_k\)
- \(SP'\): Sepset associated with an edge in \(CTF_{k,a}\)
- \(\beta(C)\): Unnormalized clique belief of clique \(C\)
- \(\beta_N(C)\): Normalized clique belief of clique \(C\), \(\beta_N(C) = \frac{\beta(C)}{\sum_{C' \in \Phi_k} \beta(C')}\)
- \(Z_k\): Normalization constant of the distribution encoded by calibrated beliefs in \(CTF_k\)
- \(Q_k(X_k)\): Probability distribution corresponding to \(CTF_k\)
- \(Q_{k,a}(X_{k,a})\): Probability distribution corresponding to \(CTF_{k,a}\)

Propositions related to inference of marginals: Let \(CTF_k\) be a CTF in the SCTF generated by the IBIA framework and \(CTF_{k,a}\) be the corresponding approximate CTF.

**Proposition 1.** The joint belief of variables contained within any clique in the approximate CTF \(CTF_{k,a}\) is the same as that in \(CTF_k\).

**Proof.** The approximation algorithm has two steps, exact marginalization and local marginalization. Exact marginalization involves finding the joint belief by collapsing all cliques containing a variable and then marginalizing the belief by summing over the states of the variable. This does not change the belief of the remaining variables. Local marginalization involves marginalizing a variable from individual cliques and sepsets by summing over its states. Let \(C'\) denote the clique obtained after local marginalization of variable \(v\) from clique \(C\). The updated clique belief \(\beta(C')\) is computed as shown below.

\[
\beta(C') = \sum_v \beta(C)
\]

Once again, summing over the states of a variable does not alter the joint belief of the remaining variables in the clique. 

\[\square\]
Algorithm 1 InferMarginals ($\Phi, mcs_p, mcs_{im}$)

**Input:** $\Phi$: Set of factors in the PGM
$mcs_p$: Maximum clique size bound for each CTF in the sequence
$mcs_{im}$: Maximum clique size bound for the approximate CTF

**Output:** $MAR$: Map containing marginals < variable : margProb >

1: Initialize: $MAR = <>$  \hspace{1cm} ▷ Map < variable : margProb >
   $S_v = \cup_{\phi \in \Phi} Scope(\phi)$  \hspace{1cm} ▷ Set of all variables in the PGM
   $SCTF = []$  \hspace{1cm} ▷ Sequence of calibrated CTFs
   $SL = []$  \hspace{1cm} ▷ List of list of links between all adjacent CTFs
   $k = 1$  \hspace{1cm} ▷ Index of CTF in $SCTF$

2: while $\Phi$.isEmpty() do  \hspace{1cm} ▷ Convert PGM $\Phi$ to $SLCTF = \{SCTF, SL\}$
   3: if $k == 1$ then
      4: $CTF_0 \leftarrow$ Disjoint cliques corresponding to factors in $\Phi$ with disjoint scopes
      5: $\triangleright$ Add factors to $CTF_0$ using BuildCTF (Algorithm 1 in Bathla and Vasudevan [2023])
      6: $CTF_1, \Phi_1 \leftarrow$ BuildCTF ($CTF_0, \Phi, mcs_p$)  \hspace{1cm} ▷ $\Phi_1$: Subset of factors in $\Phi$ added to $CTF_1$
      7: $\Phi \leftarrow \Phi \setminus \Phi_1$  \hspace{1cm} ▷ Remove factors added to $CTF_1$ from $\Phi$
      8: else
         9: $\triangleright$ Add factors to $CTF_{k-1,a}$ using BuildCTF (Algorithm 1 in Bathla and Vasudevan [2023])
      10: $CTF_{k-1}, \Phi_k \leftarrow$ BuildCTF ($CTF_{k-1,a}, \Phi, mcs_p$)  \hspace{1cm} ▷ $\Phi_k$: Subset of factors in $\Phi$ added to $CTF_k$
      11: $\Phi \leftarrow \Phi \setminus \Phi_k$  \hspace{1cm} ▷ Remove factors added to $CTF_k$ from $\Phi$
      12: $L_{k-1} \leftarrow$ FindLinks($CTF_{k-1}, CTF_{k-1,a}, CTF_k$)  \hspace{1cm} ▷ $L_{k-1}$: List of links between $CTF_{k-1}, CTF_k$
      13: $\triangleright$ Add $L_{k-1}$ to the sequence of links $SL$
   end if

15: Calibrate $CTF_k$ using belief propagation
16: $SCTF.append(CTF_k)$  \hspace{1cm} ▷ Add $CTF_k$ to the sequence $SCTF$
17: $\triangleright$ Reduce clique sizes to $mcs_{im}$ using ApproximateCTF (Algorithm 2 in Bathla and Vasudevan [2023])
18: $CTF_{k,a} \leftarrow$ ApproximateCTF ($CTF_k, \Phi, mcs_{im}$)
19: $k \leftarrow k + 1$

21: $SLCTF = \{SCTF, SL\}$  \hspace{1cm} ▷ Sequence of linked CTFs
22: $\triangleright$ Re-calibrate CTFs so that beliefs in all CTFs account for all factors
23: $MAR[v] \leftarrow$ Find marginal of $v$ from $CTF_j$ s.t. $v \in CTF_k, v \notin CTF_{k+1}$ \forall $v \in S_v$  ▷ Infer marginals

25: **procedure** FindLinks($CTF_{k-1}, CTF_{k-1,a}, CTF_k$)
   26: $\triangleright$ Each link is a triplet consisting of $C \in CTF_{k-1}, C' \in CTF_{k-1,a}$ and $\hat{C} \in CTF_k$
   27: for $C', C \in CTF_{k-1,a}$ do
      28: $\triangleright$ Find links corresponding to each clique $C'$ in $CTF_{k-1,a}$
      29: $\triangleright$ Find list of corresponding cliques in $CTF_{k-1}, L_c$
      30: if $C'.isCollapsedClique$ then  \hspace{1cm} ▷ $C'$ is obtained after exact marginalization
         31: $L_c \leftarrow$ List of cliques in $CTF_{k-1}$ that were collapsed to form $C'$
      else  \hspace{1cm} ▷ $C'$ is either obtained after local marginalization or it is present as is in $CTF_k$
         32: $C \leftarrow$ Clique in $CTF_{k-1}$ s.t. $C' \subseteq C$; $L_c = |C|$  \hspace{1cm} ▷ Add all links corresponding to $C'$
      end if
      $\triangleright$ for $C \in L_c$ do $L_{k-1}.append((C, C', \hat{C}))$ end for
   end for
38: return $L_{k-1}$$\triangleright$ Back-propagate beliefs from $CTF_k$ to $CTF_{k-1}$ via all selected links
40: **end procedure**

41: **procedure** BeliefUpdate($SLCTF$)
42: $SCTF, SL = SLCTF$
43: for $k \in len(SCTF)$ down to 2 do  \hspace{1cm} ▷ Update beliefs in $\{CTF_k, k < len(SCTF)\}$
   44: $CTF_{k-1} \leftarrow SCTF[k-1]; CTF_k = SCTF[k]; L_{k-1} = SL[k-1]$  \hspace{1cm} ▷ Back-propagate beliefs from $CTF_k$ to $CTF_{k-1}$ via all selected links
   45: $L_c \leftarrow$ Priority queue with subset of links in $L_{k-1}$ chosen using heuristics described in Section 3.2
   46: for $(C, C', \hat{C}) \in L_c$ do
      47: $\triangleright$ Update $\beta(C)$ in $CTF_{k-1}$ based on $\beta(\hat{C})$ in $CTF_k$
      $\hat{C}$ : Update belief of all other cliques in $CTF_{k-1}$ using single pass message passing with $C$ as root
   end for
49: end for
50: end for
51: **end procedure**
Proposition 2. The clique beliefs in CTF_k account for all factors added to \{CTF_1, \ldots, CTF_k\}.

Proof. CTF_1 is constructed by adding factors to an initial CTF that contains a set of disjoint cliques corresponding to a subset of factors with disjoint scopes. Let \Phi_1 be the set of all factors present in CTF_1 and Z_1 be the corresponding normalization constant. After calibration, the normalized clique belief (\beta_N(C)) of any clique C in CTF_1 can be computed as follows.

\[
\beta_N(C) = \frac{1}{Z_1} \sum_{X_{1 \setminus C}} \prod_{C_i \in CTF_1} \beta(C_i) = \frac{1}{Z_1} \sum_{X_{1 \setminus C}} \prod_{\phi \in \Phi_1} \phi
\]

Therefore, clique beliefs in CTF_1 account for all factors in \Phi_1.

CTF_{1,a} is a calibrated CTF (refer Proposition 6, Bathla and Vasudevan [2023]) that is obtained after approximate marginalization of the variables in X_{1 \setminus X_{1,a}}. Therefore, the joint distribution of variables in CTF_{1,a} also accounts for all factors in \Phi_1. CTF_2 is constructed by adding factors in \Phi_2 to CTF_{1,a}. Therefore, after calibration, the normalized clique belief (\beta_N(C)) of any clique C in CTF_2 can be computed as follows.

\[
\beta_N(C) = \frac{1}{Z_2} \sum_{X_{2 \setminus C}} \frac{1}{Z_1} \sum_{X_{1 \setminus C}} \prod_{C_i \in CTF_{1,a}} \beta(C_i) \prod_{\phi \in \Phi_2} \phi
\]

where, Z_2 is the normalization constant of the distribution in CTF_2. Using equation 1, the clique beliefs in CTF_2 accounts for all factors in \Phi_1 and \Phi_2.

A similar procedure can be repeated for subsequent CTs to show that the proposition holds true for all CTs in the sequence. \qed

Propositions related inference in BNs:

The following propositions hold true for Bayesian networks when each CTF in the SCTF is constructed by adding factors or conditional probability distributions (CPD) of variables in the topological order. \hat{Y}_k denotes the set of variables whose CPDs are added during construction of CTF_k and \varepsilon_k denotes the evidence states of all evidence variables in Y_k.

Proposition 3. The product of factors added in CTFs, \{CTF_1, \ldots, CTF_k\} is a valid joint probability distribution whose normalization constant is the probability of evidence states \varepsilon_1, \ldots, \varepsilon_k.

Proof. Let \mathcal{Y}_k = \{Y_1, \ldots, Y_k\} and \varepsilon_k = \{e_1, \ldots, e_k\}. Since CTs are constructed by adding CPDs of variables in the topological order, the CPDs of parents P_{\mathcal{Y}_k} are present in \{CTF_1, \ldots, CTF_k\}. Therefore, the product of the CPDs is the unnormalized joint probability distribution P(\mathcal{Y}_k, \varepsilon_k). Since the CPDs of all non-evidence variables are normalized to one, the normalization constant is P(\varepsilon_k). \qed

Proposition 4. The normalization constant of the distribution encoded by the calibrated beliefs in CTF_k is the estimate of probability of evidence states \varepsilon_1, \ldots, \varepsilon_k.

Proof. The initial factors assigned to CTF_1 are CPDs of variables in Y_1. Therefore, using Proposition 3 the NC obtained after calibration is Z_1 = P(\varepsilon_1).

CTF_{1,a} is obtained after approximation of CTF_1. All CTs in CTF_{1,a} are calibrated CTs and the normalization constant of the distribution in CTF_{1,a} is same as that of CTF_1 (refer Propositions 6 and 9 in Bathla and Vasudevan [2023]). However, due to local marginalization, the overall distribution represented by CTF_{1,a} is approximate. The probability distribution corresponding to CTF_{1,a} can be written as follows.

\[
Q_{1,a}(X_{1,a}|e_1) = \frac{1}{Z_1} \frac{1}{Z_1} \prod_{C' \in CTF_{1,a}} \beta(C') \prod_{SP' \in CTF_{1,a}} \mu(SP')
\]

\[
\Rightarrow Z_1 Q_{1,a}(X_{1,a}|e_1) = Q_{1,a}(X_{1,a}, e_1)
\]

where X_{1,a} is the set of variables in CTF_{1,a}. \qed
$CTF_2$ is obtained after adding a new set of CPDs of variables in $Y_2$ to $CTF_{1,a}$. Let $X_2 = X_{1,a} \cup \{Y_2 \setminus E_2\}$ denote the set of non-evidence variables in $CTF_2$ and $P_{a_Y}$ denote the parents of variables in $Y_2$. The NC of the distribution encoded by $CTF_2 (Z_2)$ can be computed as follows.

$$Z_2 = \sum_{X_2} \prod_{C' \in CTF_{1,a}} \beta(C') \prod_{SP' \in CTF_{1,a}} \mu(SP') \prod_{y \in Y_2} P(y \mid Pa_y)$$

$$= \sum_{X_2} Q_{1,a}(X_{1,a}, e_1) P(Y_2, e_2 \mid Pa_Y) \quad \text{(using Equation 2)}$$

where $e_2$ are evidence states in $Y_2$. Since $X_2 = X_{1,a} \cup \{Y_2 \setminus E_2\}$ and parent variables in $Pa_Y$ are present either in $X_{1,a}$ or $Y_2$, the above equation can be re-written as follows.

$$Z_2 = \sum_{X_2} Q_2(X_2, e_1, e_2) = Q(e_1, e_2)$$

Therefore, the NC of $CTF_2$ is an estimate of probability of evidence states $e_1$ and $e_2$.

A similar procedure can be repeated for subsequent $CTFs$ to show that the property holds true for all $CTFs$ in the sequence.

**Theorem 1.** Let $I_E$ denote the index of the last $CTF$ in the sequence where the factor corresponding to an evidence variable is added. The posterior marginals of variables present in $CTFs \{CTF_k, k \geq I_E\}$ are preserved and can be computed from any of these $CTFs$.

**Proof.** Let $\varepsilon_{I_E} = \{e_1, \ldots, e_{I_E}\}$ be the set of all evidence states. Let $v$ be a variable present in cliques $C_v \in CTF_{I_E}$, $C'_v \in CTF_{I_E,a}$ and $\tilde{C}_v \in CTF_{I_E+1}$ and let $\beta_N(C_v)$, $\beta_N(C'_v)$ and $\beta_N(\tilde{C}_v)$ be the corresponding normalized clique beliefs. From Proposition 1 the unnormalized belief of variable $v$ in $C_v$ is same as that in $C'_v$. Therefore, the normalized posterior marginal of $v$ obtained from $C_v$ (denoted as $Q_{I_E}(v \mid \varepsilon_{I_E})$) is the same as that obtained from $C'_v$, as given below.

$$Q_{I_E}(v \mid \varepsilon_{I_E}) = \sum_{C_v \setminus v} \beta_N(C_v) = \sum_{C'_v \setminus v} \beta_N(C'_v)$$

Using Equation 2

Since $CTF_{I_E,a}$ is calibrated (Proposition 6 in Bathla and Vasudevan [2023]) and $CTF_{I_E+1}$ is obtained by adding CPDs of variables in $Y_{I_E+1}$ to $CTF_{I_E,a}$, the NC of $CTF_{I_E+1}$ can be computed by summing over all non-evidence variables as follows.

$$Z_{I_E+1} = \sum_{X_{I_E,a}} \prod_{C' \in CTF_{I_E,a}} \beta(C') \prod_{SP' \in CTF_{I_E,a}} \mu(SP') \sum_{Y_{I_E+1} \setminus E_{I_E+1}} \sum_{Y_{I_E+1} \setminus E_{I_E+1}} P(Y_{I_E+1} \mid E_{I_E+1} = \emptyset, \sum_{Y_{I_E+1} \setminus E_{I_E+1}} P(Y_{I_E+1} \mid Pa_{Y_{I_E+1}}) = 1)$$

$$= Z_{I_E} \quad \text{(using Proposition 9 in Bathla and Vasudevan [2023])}$$

Therefore, the posterior marginal of $v$ in $CTF_{I_E+1}$ (denoted as $Q_{I_E+1}(v \mid \varepsilon_{I_E})$) can be computed from the clique belief of $\tilde{C}_v$ as follows.

$$Q_{I_E+1}(v \mid \varepsilon_{I_E}) = \sum_{\tilde{C}_v \setminus v} \beta_N(\tilde{C}_v)$$

$$= \sum_{X_{I_E,a} \setminus \varepsilon_{I_E}} \frac{1}{Z_{I_E}} \prod_{C' \in CTF_{I_E,a}} \beta(C') \prod_{SP' \in CTF_{I_E,a}} \mu(SP') \sum_{Y_{I_E+1} \setminus E_{I_E+1}} \sum_{Y_{I_E+1} \setminus E_{I_E+1}} P(Y_{I_E+1} \mid E_{I_E+1} = \emptyset, \sum_{Y_{I_E+1} \setminus E_{I_E+1}} P(Y_{I_E+1} \mid Pa_{Y_{I_E+1}}) = 1)$$

$$= \sum_{C'_v \setminus v} \beta_N(C'_v) \quad \text{(denoted as } C'_v \text{ in } CTF_{I_E,a} \text{ and } E_{I_E+1} = \emptyset)$$

$$= Q_{I_E}(v \mid \varepsilon_{I_E}) \quad \text{(using Equation 4)}$$

The above procedure can be repeated to show that the posterior marginal of $v$ is also consistent in all subsequent $CTFs$ that contain $v$. 

\[\square\]
References


Vibhav Gogate. Iterative join graph propagation. https://personal.utdallas.edu/~vibhav

Vibhav Gogate. IJGP-sampling and samplesearch (PR and MAR tasks). https://github.com/


