Solving Marginal MAP Exactly by Probabilistic Circuit Transformations

YooJung Choi

Joint work with Tal Friedman and Guy Van den Broeck

Probabilistic Circuits and Logic Workshop @ Simons Institute – *Oct 18, 2023*

Marginal MAP (MMAP)

Given a set of query variables $\boldsymbol{Q} \subset \boldsymbol{X}$ and evidence $\boldsymbol{e}$,

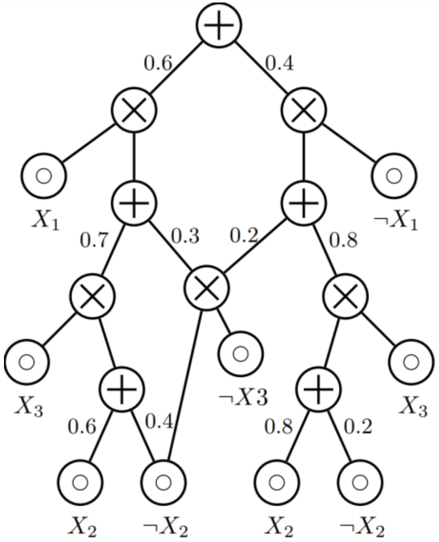
$$\arg\max_{\boldsymbol{q} \in \mathsf{val}(\mathbf{Q})} p(\boldsymbol{q}, \boldsymbol{e}) = \arg\max_{\boldsymbol{q} \in \mathsf{val}(\mathbf{Q})} \sum_{\boldsymbol{h} \in \mathsf{val}(\mathbf{H})} p(\boldsymbol{q}, \boldsymbol{h}, \boldsymbol{e})$$

⇒ *i.e. MAP of a marginal distribution on* $\boldsymbol{Q}$

⇒ *in general, NP^{PP}-hard*

MMAP on PCs

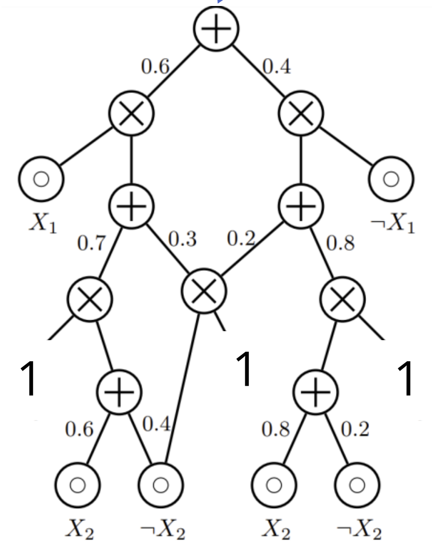
$$\arg\max_{\boldsymbol{q} \in \mathsf{val}(\mathbf{Q})} p(\boldsymbol{q}, \boldsymbol{e}) = \arg\max_{\boldsymbol{q} \in \mathsf{val}(\mathbf{Q})} \sum_{\boldsymbol{h} \in \mathsf{val}(\mathbf{H})} p(\boldsymbol{q}, \boldsymbol{h}, \boldsymbol{e})$$



- Smooth + decomposable ⇒ tractable marginal
  - e.g. $p(X_1 = 1, X_2 = 0)$
- \+ deterministic ⇒ tractable MAP
  - e.g. $\max_{X_1 X_2 X_3} p(X_1, X_2, X_3)$
- MMAP: ***NP-hard*** *even for PCs that are tractable for marginals & MAP*
  - Intuition: need the PC $p(\boldsymbol{Q})$ to be deterministic

MMAP on PCs

$$\arg\max_{\boldsymbol{q}\in\mathsf{val}(\mathbf{Q})} p(\boldsymbol{q},\boldsymbol{e}) = \arg\max_{\boldsymbol{q}\in\mathsf{val}(\mathbf{Q})} \sum_{\boldsymbol{h}\in\mathsf{val}(\mathbf{H})} p(\boldsymbol{q},\boldsymbol{h},\boldsymbol{e})$$



$\mathbf{Q} = \{X_1, X_2\}$

- Smooth + decomposable ⇒ tractable marginal
  - e.g. $p(X_1 = 1, X_2 = 0)$
- \+ deterministic ⇒ tractable MAP
  - e.g. $\max_{X_1 X_2 X_3} p(X_1, X_2, X_3)$
- MMAP: ***NP-hard** even for PCs that are tractable for marginals & MAP*
  - Intuition: need the PC $p(\boldsymbol{Q})$ to be deterministic

MMAP on PCs

- Enforce circuit constraints to get linear-time MMAP
  - E.g. constrained pseudo-tree (AND/OR search) [Marinescu, Dechter & Ihler ’14], constrained vtree ((P)SDD) [Oztok, Choi & Darwiche ‘16]
  - Marginal determinism (aka Q-determinism)
  - *Circuit size may blow up*
  - *Need to change the circuit for a different query variable set* ***Q***
- Branch-and-bound search [Huang, Chavira & Darwiche ‘06; Mei, Jiang & Tu ‘18]

Our approach: iterative circuit transformations

Bounds on MMAP

Upper bound through a single feedforward pass [Huang et al. '06]

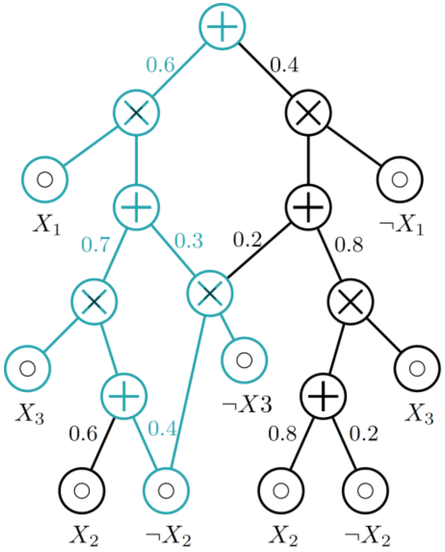
Lower bound: $p(\boldsymbol{q})$ for any $\boldsymbol{q} \in val(\boldsymbol{Q})$ works

Q: can we tighten these bounds further?

$\Rightarrow$ *transform the PC to get better bounds*

Circuit pruning for MMAP

Some parts of the circuit may be irrelevant for the MMAP solution



- Example: computing $p(X_1 = 1, X_2 = 0)$
  - Only the highlighted edges are used
  - Remaining edges propagate zero
- $X_1 = 1, X_2 = 0$ is the MMAP solution for $\mathbf{Q} = \{X_1, X_2\}$
  - Pruning any black edge does not affect the MMAP solution

Q: can we efficiently identify which edges can be safely pruned?

Edge bounds for MMAP

For every edge, what is the maximum marginal probability $p(\boldsymbol{q})$ that uses/activates that edge?

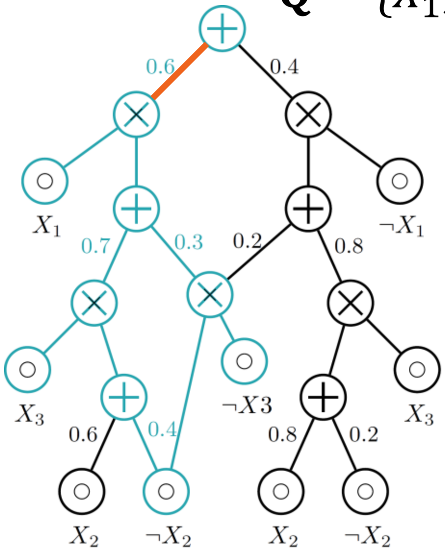
$$\forall(n,c)\text{: define } \mathrm{EB(n,c)} \geq \max_{\boldsymbol{q}\in\mathcal{C}_{n,c}} p(\boldsymbol{q})$$

$$\mathcal{C}_{n,c} = \{\boldsymbol{q} \in \mathrm{val}(\boldsymbol{Q}) \colon p(\boldsymbol{q}) \text{ "activates" edge } (n,c)\}$$

$$(X_1 = 1, X_2 = 0) \in \mathcal{C}_{n,c}, \quad (X_1 = 1, X_2 = 0) \notin \mathcal{C}_{n,c}$$

$$\mathcal{C}_{r,1} = \{(X_1 = 1, X_2 = 0), (X_1 = 1, X_2 = 1)\}$$

$\mathbf{Q} = \{X_1, X_2\}$



Edge bounds for MMAP

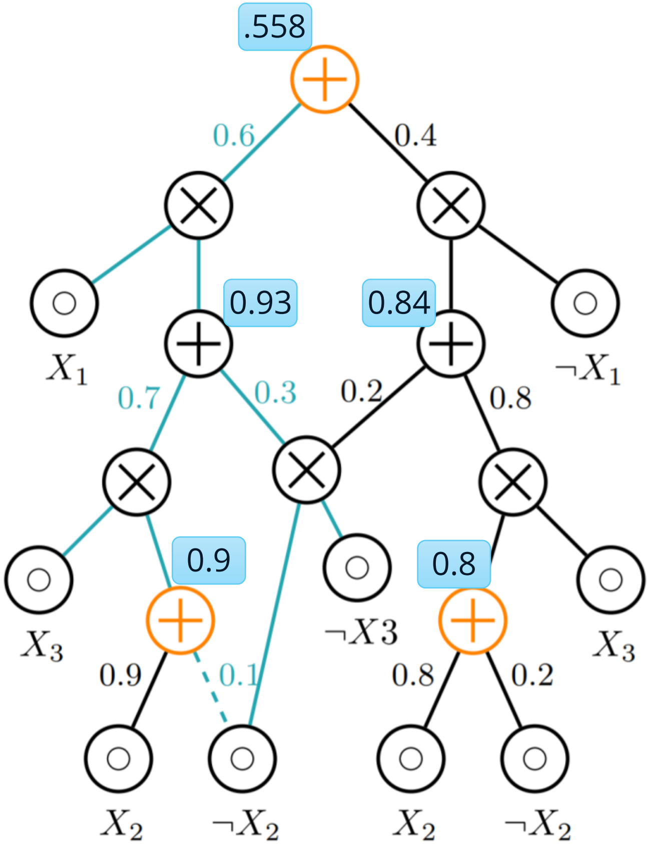
For every edge, what is the maximum marginal probability $p(\boldsymbol{q})$ that uses/activates that edge?

$$\forall(n, c)\text{: define } \mathrm{EB(n, c)} \geq \max_{\boldsymbol{q} \in \mathcal{C}_{n,c}} p(\boldsymbol{q})$$

$$\mathcal{C}_{n,c} = \{\boldsymbol{q} \in \mathrm{val}(\boldsymbol{Q}) \colon p(\boldsymbol{q}) \text{ "activates" edge } (n, c)\}$$

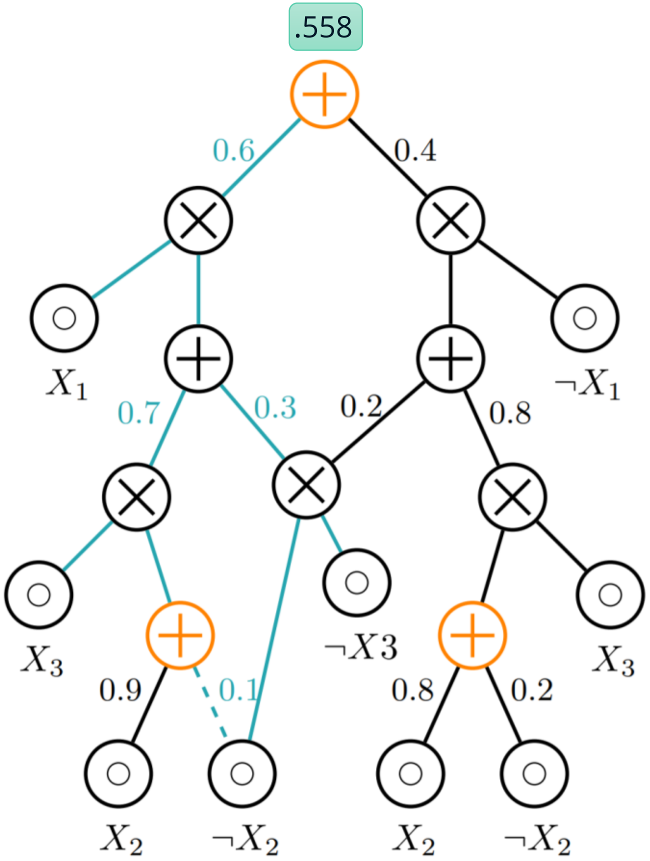
Given a lower bound l on MMAP, we can ***safely prune any edge*** (n, c) ***if*** $\mathrm{EB}(n, c) < l$.

For a smooth and decomposable PC, ***all edge bounds*** *can be computed with* ***a single feedforward & backward pass*** through the circuit.



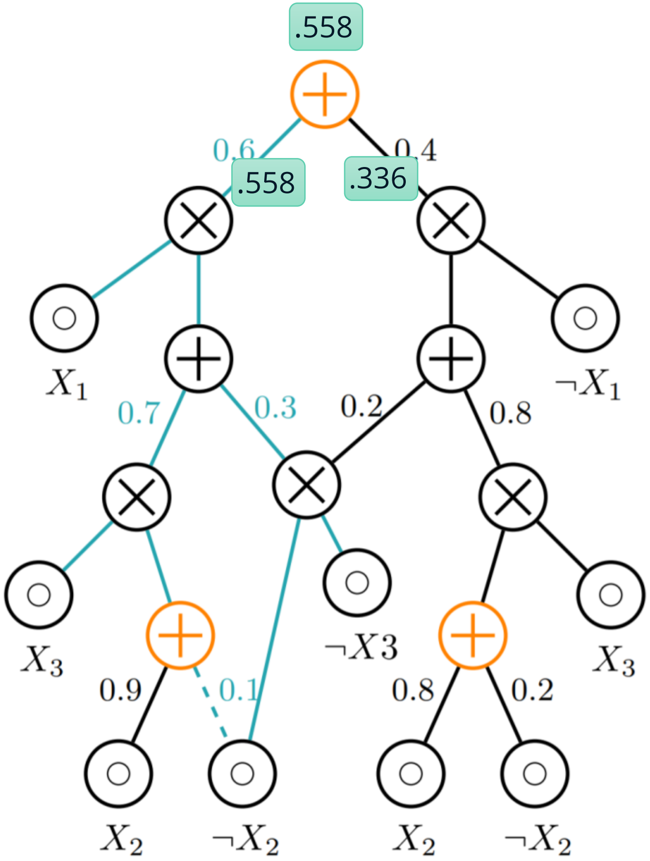
Feedforward pass: upper-bound on MMAP [Huang et al. '06]

Q-deterministic sum => max



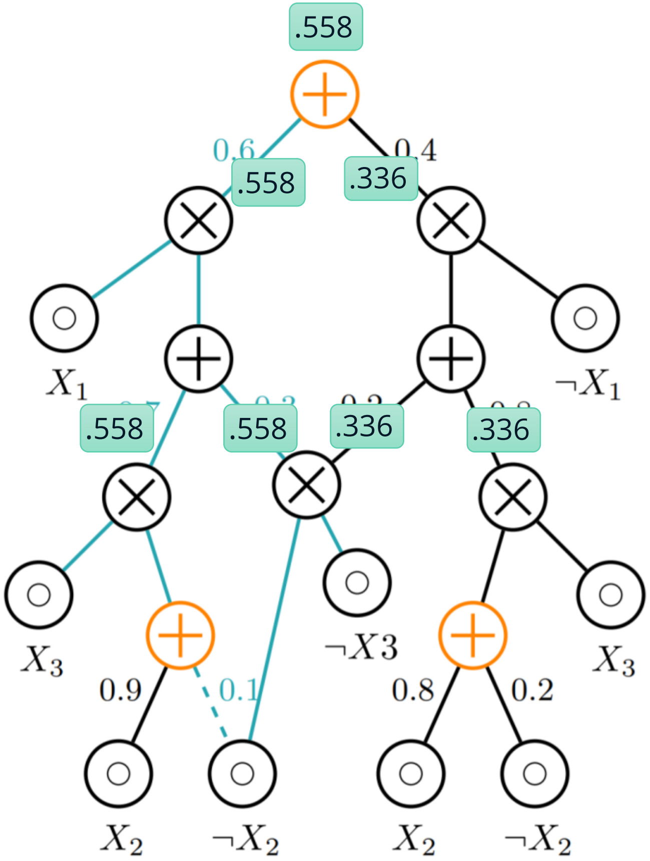
Feedforward pass: upper-bound on MMAP [Huang et al. '06]

Backward pass: tighten $\boxed{\mathrm{EB}(n, c)}$ at every **Q**-deterministic sum n



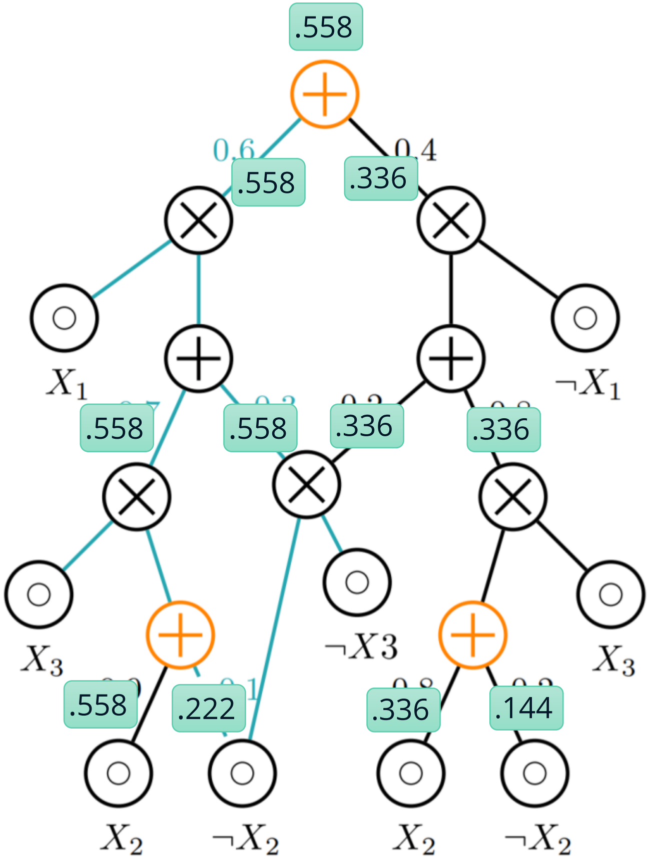
Feedforward pass: upper-bound on MMAP [Huang et al. '06]

Backward pass: tighten $\boxed{\mathrm{EB}(n, c)}$ at every **Q**-deterministic sum n



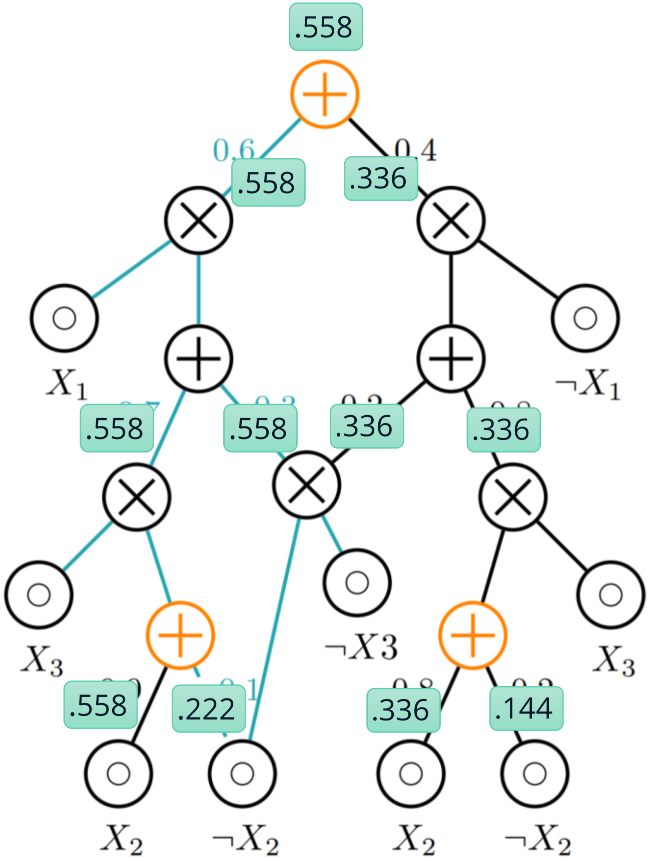
Feedforward pass: upper-bound on MMAP [Huang et al. '06]

Backward pass: tighten $\boxed{\mathrm{EB}(n, c)}$ at every **Q**-deterministic sum n



Feedforward pass: upper-bound on MMAP [Huang et al. '06]

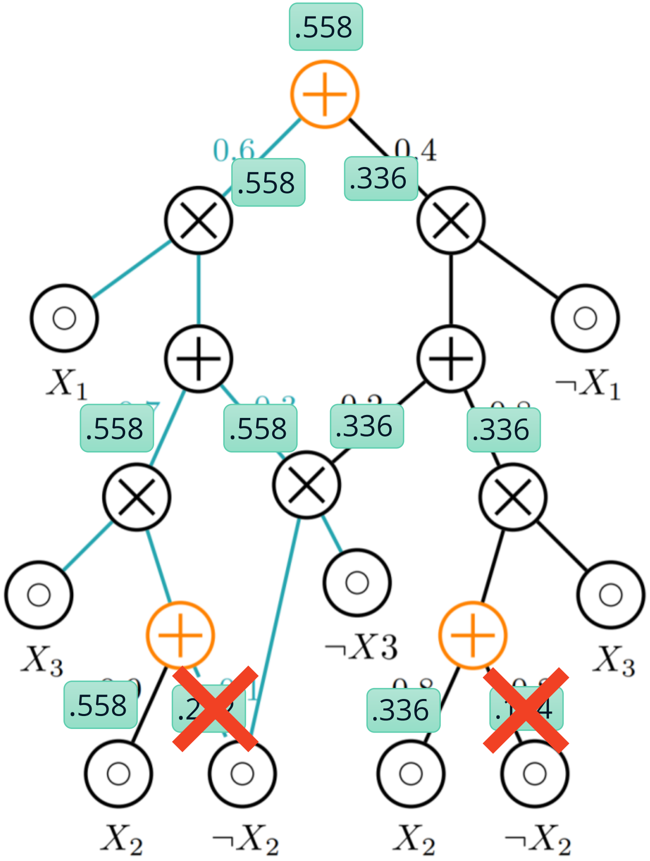
Backward pass: tighten $\mathrm{EB}(n, c)$ at every **Q**-deterministic sum n



Feedforward pass: upper-bound on MMAP [Huang et al. ’06]

Backward pass: tighten $\boxed{\mathrm{EB}(n, c)}$ at every **Q**-deterministic sum n

e.g. using $p(X_1 = 0, X_2 = 1) = 0.256$ as lower bound

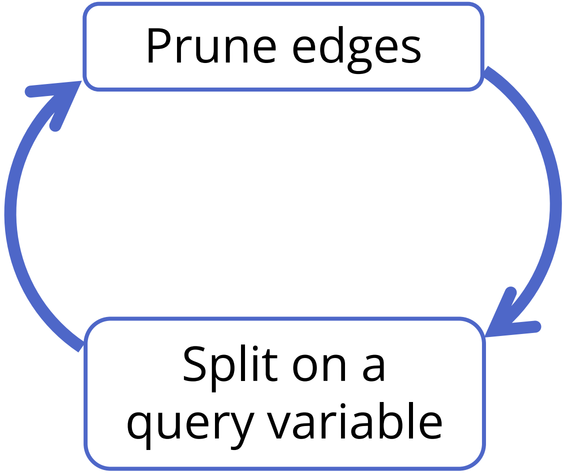


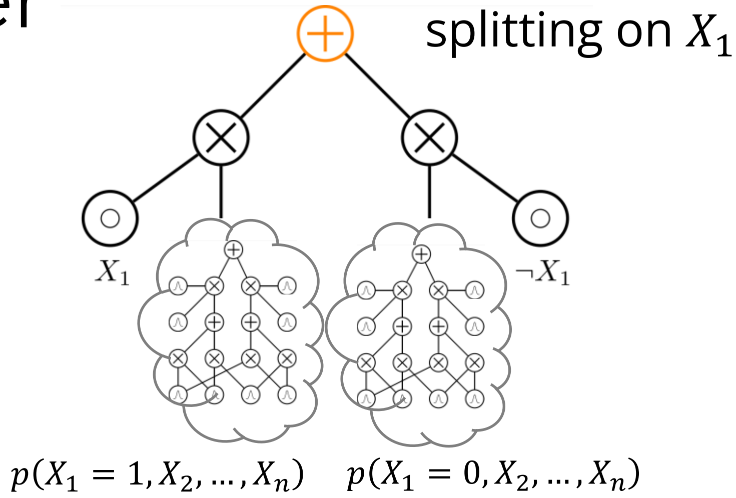
Feedforward pass: upper-bound on MMAP [Huang et al. '06]

Backward pass: tighten $\boxed{\mathrm{EB}(n,c)}$ at every **Q**-deterministic sum n

e.g. using $p(X_1 = 0, X_2 = 1) = 0.256$ as lower bound

Iterative MMAP solver

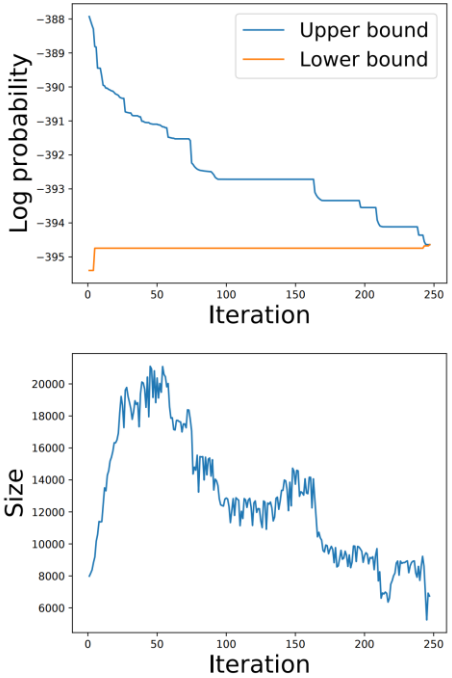




- Each split tightens the bound
- After splitting on $Q_1, \dots, Q_n$
  $\Rightarrow$ linear-time MMAP for $\boldsymbol{Q} = \{Q_1, \dots, Q_n\}$

Empirical evaluation

Example run



Average run time in seconds (# instances solved)

Dataset	(30%, 30%, 40%) MaxSPN	(30%, 30%, 40%) (ours)	(50%, 20%, 30%) MaxSPN	(50%, 20%, 30%) (ours)
NLTCS	**0.004** (10)	0.54 (10)	**0.01** (10)	0.63 (10)
MSNBC	**0.01** (10)	0.50 (10)	**0.03** (10)	0.73 (10)
KDD	**0.02** (10)	0.64 (10)	**0.04** (10)	0.68 (10)
Plants	**0.27** (10)	1.36 (10)	2.95 (10)	**2.72** (10)
Audio	188.59 (10)	**2.87** (10)	2041.33 (6)	**13.70 (10)**
Jester	265.50 (10)	**6.17** (10)	2913.04 (2)	**14.74 (10)**
Netflix	344.71 (10)	**5.61** (10)	– (0)	**47.18 (10)**
Accidents	**0.54** (10)	2.00 (10)	109.56 (10)	**15.86** (10)
Retail	**0.03** (10)	0.61 (10)	**0.06** (10)	0.81 (10)
Pumsb-star	273.70 (10)	**6.04** (10)	2208.27 (7)	**20.88 (10)**
DNA	2809.44 (4)	**9.16 (10)**	– (0)	**505.75 (9)**
Kosarek	1.60 (10)	**0.98** (10)	48.74 (10)	**3.41** (10)
MSWeb	25.70 (10)	**0.96** (10)	1543.49 (10)	**1.28** (10)
Book	– (0)	**7.25 (10)**	– (0)	**46.50 (10)**
EachMovie	– (0)	**93.66 (10)**	– (0)	**1216.89 (8)**
WebKB	– (0)	**102.37 (10)**	– (0)	**575.68 (10)**
Reuters-52	– (0)	**22.91 (10)**	– (0)	**120.58 (10)**
20 NewsGrp.	– (0)	**88.13 (10)**	– (0)	**504.52 (9)**
BBC	– (0)	**766.93 (9)**	– (0)	**2757.18 (3)**
Ad	– (0)	**344.81 (10)**	– (0)	**1254.37 (8)**
Total Solved	124	**199**	105	**187**

Jun Mei, Yong Jiang, and Kewei Tu. “Maximum A Posteriori Inference in Sum-Product Networks.“ In: AAAI 2018.

Conclusion

- Iterative pruning and splitting to tighten MMAP bounds
    - Each split may (worst-case) double the circuit size, but pruning can be effective *in practice*
- Also an iterative MPE solver for non-deterministic PCs
- Can we generalize the bounds to other queries that require determinism for tractability?