Probabilistic Circuits:
Marginal Maximum a Posteriori Queries

Markus Pawellek

January 21, 2022

Outline

Introduction

Introduction

- Marginal maximum a posteriori (MMAP) combine marginal (MAR) and maximum a posteriori (MAP) inference

Introduction

- Marginal maximum a posteriori (MMAP) combine marginal (MAR) and maximum a posteriori (MAP) inference
- We need sufficient conditions for tractability

Introduction

- Marginal maximum a posteriori (MMAP) combine marginal (MAR) and maximum a posteriori (MAP) inference
- We need sufficient conditions for tractability
- We use MAR and MAP algorithm

Introduction

- Marginal maximum a posteriori (MMAP) combine marginal (MAR) and maximum a posteriori (MAP) inference
- We need sufficient conditions for tractability
- We use MAR and MAP algorithm
- Notations varies a little bit: no boldface letters

Background

Background

X . . . finite set of random variables

p . . . joint probability distribution over X

$\mathcal{C}$ . . . PC over X with $\mathcal{C} = (\mathcal{G}, \vartheta)$

Q . . . set of query variables with $Q \subset X$

Background: MMAP Queries

- query variables Q, evidence variables E, marginal variables Z form partition of X
- $e \in E$, intevals $\mathcal{I} \subset \mathrm{val}(Z)$

$$\underset{q \in \mathrm{val}(Q)}{\arg\max}\ p\left(Q = q \mid E = e, Z \in \mathcal{I}\right)$$

$$\underset{q \in \mathrm{val}(Q)}{\arg\max} \int_{\mathcal{I}} p(q, e, z)\, \mathrm{d}Z$$

Background: MMAP Connection

$$\arg\max_{q \in \mathrm{val}(Q)} \int_{\mathcal{J}} p(q, e, z)\, \mathrm{d}Z$$

- **case** $Q = \emptyset$: MAR Query

$$\int_{\mathcal{J}} p(e, z)\, \mathrm{d}Z$$

- **case** $Z = \emptyset$: MAP Query

$$\arg\max_{q \in \mathrm{val}(Q)} p(q, e)$$

Background: Review

$\mathcal{C}$ is tractable for MAR queries

$\iff$ $\mathcal{G}$ is decomposable and smooth

$\mathcal{C}$ is tractable for MAP queries

$\iff$ $\mathcal{G}$ is consistent and deterministic

$\mathcal{G}$ is decomposable $\implies$ $\mathcal{G}$ is consistent

$\mathcal{C}$ is tractable for both MAR and MAP queries

$\iff$ $\mathcal{G}$ is decomposable, smooth, and deterministic

Marginal Determinism

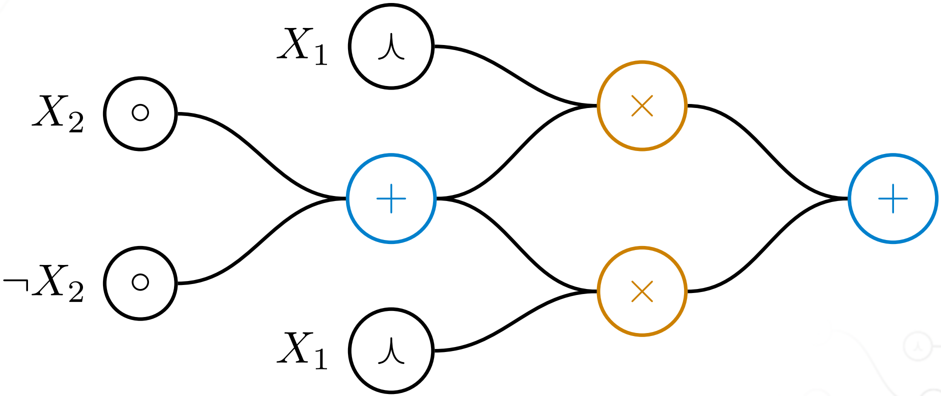
Marginal Determinism

A sum node is marginal deterministic with respect to Q if for all partial states $q \in Q$ at most one of its inputs is non-zero.

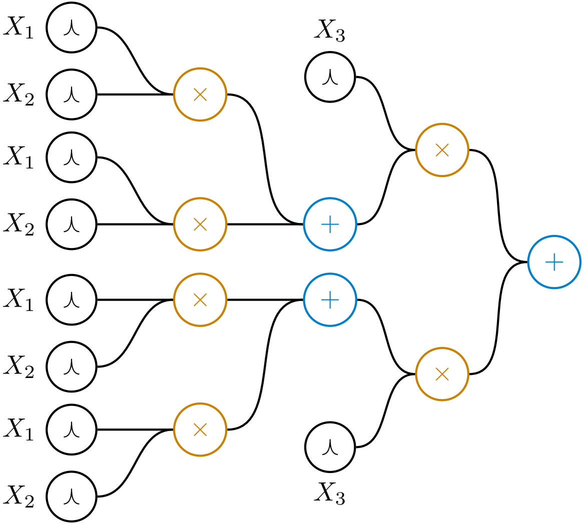
A circuit structure $\mathcal{G}$ is marginal deterministic with respect to Q if for all sum units $n \in \mathcal{G}$ with $\varphi(n) \cap Q \neq \emptyset$, n is marginal deterministic with respect to Q.

- Simple generalization of determinism
- Structural property about the support of input units
- Defined with respect to Q

Marginal Determinism: Example



Marginal Determinism: Example



Tractable Computation

Tractable Computation: General

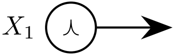
- Adjust feed-forward algorithms for MAR and MAP queries

Tractable Computation: General

- Adjust feed-forward algorithms for MAR and MAP queries
- At the end, backward pass is needed for modes

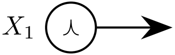
Tractable Computation: General

- Adjust feed-forward algorithms for MAR and MAP queries
- At the end, backward pass is needed for modes
- Assume, input units provide correct output



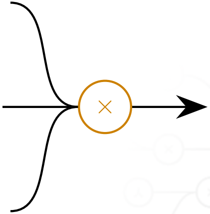
Tractable Computation: General

- Adjust feed-forward algorithms for MAR and MAP queries
- At the end, backward pass is needed for modes
- Assume, input units provide correct output



- Product units are handled identically in MAR and MAP

$$r_n \longleftarrow \prod_{c \in \mathrm{in}(n)} r_c$$



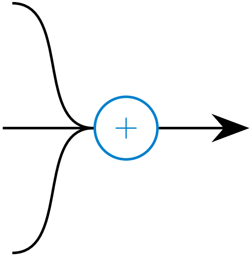
Tractable Computation: Sum Units

if $\varphi(n) \cap Q \neq \emptyset$ **then**

$$r_n \longleftarrow \max_{c \in \mathrm{in}(n)} \vartheta_{nc} r_c$$

else

$$r_n \longleftarrow \sum_{c \in \mathrm{in}(n)} \vartheta_{nc} r_c$$



Sufficient Conditions

Sufficient Conditions

Theorem: Let $\mathcal{G}$ be smooth, decomposable, and Q-marginal deterministic. Then for any parameterization ϑ the above algorithm tractably computes MMAP queries of $\mathcal{C}$ over Q.

- Proof by induction
- Input units are correct by assumption

$$\mathcal{Q}(e, \mathcal{I}) = \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \mathcal{C}(Z, q, e) \, \mathrm{d}Z$$

Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

$$\mathcal{Q}(e, \mathcal{I}) = \max_{q_1, \ldots, q_k \in \mathrm{val}(Q)} \int_{\mathcal{I}} \prod_{i=1}^{k} \mathcal{C}_i(Z_i, q_i, e_i) \, \mathrm{d}Z$$

Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

$$\mathcal{Q}(e, \mathcal{I}) = \max_{q_1, \ldots, q_k \in \mathrm{val}(Q)} \int_{\mathcal{I}} \prod_{i=1}^{k} \mathcal{C}_i(Z_i, q_i, e_i) \,\mathrm{d}Z$$

$$= \max_{q_1, \ldots, q_k \in \mathrm{val}(Q)} \prod_{i=1}^{k} \int_{\mathcal{I}} \mathcal{C}_i(Z_i, q_i, e_i) \,\mathrm{d}Z_i$$

Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

$$\begin{aligned}
\mathcal{Q}(e, \mathcal{I}) &= \max_{q_1, \ldots, q_k \in \mathrm{val}(Q)} \int_{\mathcal{I}} \prod_{i=1}^{k} \mathcal{C}_i(Z_i, q_i, e_i) \, \mathrm{d}Z \\
&= \max_{q_1, \ldots, q_k \in \mathrm{val}(Q)} \prod_{i=1}^{k} \int_{\mathcal{I}} \mathcal{C}_i(Z_i, q_i, e_i) \, \mathrm{d}Z_i \\
&= \prod_{i=1}^{k} \max_{q_i \in \mathrm{val}(Q)} \int_{\mathcal{I}_i} \mathcal{C}_i(Z_i, q_i, e_i) \, \mathrm{d}Z_i
\end{aligned}$$

Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

$$
\begin{aligned}
\mathcal{Q}(e, \mathcal{I}) &= \max_{q_1,\ldots,q_k \in \mathrm{val}(Q)} \int_{\mathcal{I}} \prod_{i=1}^{k} \mathcal{C}_i(Z_i, q_i, e_i)\, \mathrm{d}Z \\
&= \max_{q_1,\ldots,q_k \in \mathrm{val}(Q)} \prod_{i=1}^{k} \int_{\mathcal{I}} \mathcal{C}_i(Z_i, q_i, e_i)\, \mathrm{d}Z_i \\
&= \prod_{i=1}^{k} \max_{q_i \in \mathrm{val}(Q)} \int_{\mathcal{I}_i} \mathcal{C}_i(Z_i, q_i, e_i)\, \mathrm{d}Z_i \\
&= \prod_{i=1}^{k} \mathcal{C}_i(e_i, \mathcal{I}_i)
\end{aligned}
$$

Sufficient Conditions: Sum Units

- Sum units with no query variable in their scope reduce to MAR queries which has alread been proven

Sufficient Conditions: Sum Units

- Sum units with no query variable in their scope reduce to MAR queries which has alread been proven
- So $\varphi(n) \cap Q \neq \emptyset$

Sufficient Conditions: Sum Units

- Sum units with no query variable in their scope reduce to MAR queries which has alread been proven
- So $\varphi(n) \cap Q \neq \emptyset$

$$\mathcal{Q}(e, \mathcal{I}) = \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z$$

Sufficient Conditions: Sum Units

- Sum units with no query variable in their scope reduce to MAR queries which has alread been proven
- So $\varphi(n) \cap Q \neq \emptyset$

$$
\begin{aligned}
\mathcal{Q}(e, \mathcal{I}) &= \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e) \,\mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \sum_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e) \,\mathrm{d}Z
\end{aligned}
$$

Sufficient Conditions: Sum Units

- Sum units with no query variable in their scope reduce to MAR queries which has alread been proven
- So $\varphi(n) \cap Q \neq \emptyset$

$$\begin{aligned}
\mathcal{Q}(e, \mathcal{I}) &= \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \sum_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \max_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z
\end{aligned}$$

Sufficient Conditions: Sum Units

- Sum units with no query variable in their scope reduce to MAR queries which has alread been proven
- So $\varphi(n) \cap Q \neq \emptyset$

$$
\begin{aligned}
\mathcal{Q}(e, \mathcal{I}) &= \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \sum_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \max_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z \\
&= \max_{i \in \mathrm{in}(n)} \vartheta_i \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z
\end{aligned}
$$

Sufficient Conditions: Sum Units

- Sum units with no query variable in their scope reduce to MAR queries which has alread been proven
- So $\varphi(n) \cap Q \neq \emptyset$

$$
\begin{aligned}
\mathcal{Q}(e, \mathcal{I}) &= \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \sum_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \max_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z \\
&= \max_{i \in \mathrm{in}(n)} \vartheta_i \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z \\
&= \max_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{Q}_i(e, \mathcal{I})
\end{aligned}
$$

Conclusions

Conclusions

- MMAP queries are typically be NP-hard

Conclusions

- MMAP queries are typically be NP-hard
- Complexity depends on set of query variables

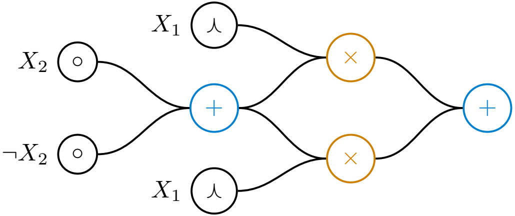
Conclusions

- MMAP queries are typically be NP-hard
- Complexity depends on set of query variables
- marginal determinism together with smoothness and decomposability seems to be sufficient for tractable computations

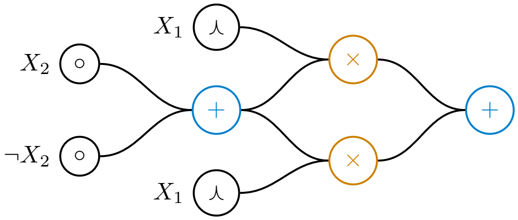
Conclusions

- MMAP queries are typically be NP-hard
- Complexity depends on set of query variables
- marginal determinism together with smoothness and decomposability seems to be sufficient for tractable computations
- sum units have to compute maxima when support contains query variables

Conclusions: Outlook

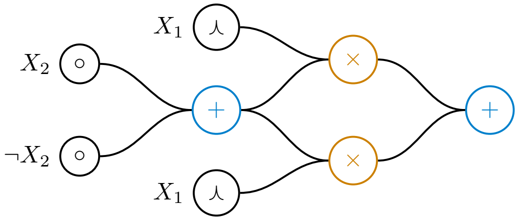


Conclusions: Outlook



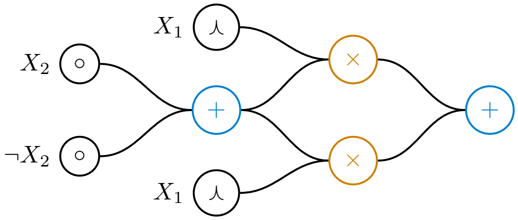
- Marginal determinism for all sets of query variables

Conclusions: Outlook



- Marginal determinism for all sets of query variables
- Sum-maximizer circuits (no general algorithm)

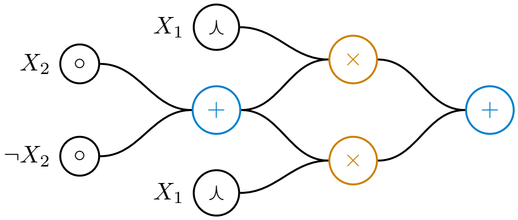
Conclusions: Outlook



- Marginal determinism for all sets of query variables
- Sum-maximizer circuits (no general algorithm)
- Tractable computation of information-theoretic measures

Conclusions: Outlook

Thank you for Your Attention!



- Marginal determinism for all sets of query variables
- Sum-maximizer circuits (no general algorithm)
- Tractable computation of information-theoretic measures

References

(1) YooJung Choi, Antonio Vergari, and Guy Van den Broeck. "Probabilistic Circuits: A Unifying Framework for Tractable Probabilistic Models". In: (October 2020). URL: http://starai.cs.ucla.edu/papers/ProbCirc20.pdf (visited on 01/19/2022).