Probabilistic Circuits: Marginal Maximum a Posteriori Queries

Markus Pawellek

January 21, 2022

Outline

Introduction

Background

Marginal Determinism

Tractable Computation

Sufficient Conditions

Conclusions

Introduction

Introduction

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

“ *Which combination of roads is most likely to be jammed on Monday?*”

Introduction

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

  “*Which combination of roads is most likely to be jammed on Monday?*”

- Sufficient conditions for tractable computation needed

Introduction

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

“*Which combination of roads is most likely to be jammed on Monday?*”

- Sufficient conditions for tractable computation needed
- Usage of MAR and MAP algorithms

Introduction

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

  "*Which combination of roads is most likely to be jammed on Monday?*"

- Sufficient conditions for tractable computation needed
- Usage of MAR and MAP algorithms
- Here, notations varies a little bit: no boldface letters

Background

Background: MMAP Queries

- Finite set X of random variables
- Joint distribution p over X

Background: MMAP Queries

- Finite set X of random variables
- Joint distribution p over X

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

for

Background: MMAP Queries

- Finite set X of random variables
- Joint distribution p over X

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

for

- Partition $\{Q,E,Z\}$ of X
- Partial state $e\in E$
- Generalized interval $\mathcal{I}\subset\mathrm{val}(Z)$

Background: MMAP Queries

- Finite set X of random variables
- Joint distribution p over X

$$\operatorname*{arg\,max}_{q \in \mathrm{val}(Q)} \; p\,(Q = q \mid E = e, Z \in \mathfrak{I}) \;=\; \operatorname*{arg\,max}_{q \in \mathrm{val}(Q)} \int_{\mathfrak{I}} p(q, e, z)\, \mathrm{d}Z$$

for

- Partition $\{Q, E, Z\}$ of X
- Partial state $e \in E$
- Generalized interval $\mathfrak{I} \subset \mathrm{val}(Z)$

Background: MAR and MAP

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

Background: MAR and MAP

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

- $\textbf{case } Q = \emptyset :$ MAR Query

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

Background: MAR and MAP

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

- **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

Probabilistic Circuit $\mathcal{C}$ with circuit structure $\mathcal{G}$

Background: Review

Probabilistic Circuit $\mathcal{C}$ with circuit structure $\mathcal{G}$

$\mathcal{C}$ is tractable for MAR queries $\iff$ $\mathcal{G}$ is smooth and decomposable

$\mathcal{C}$ is tractable for MAP queries $\iff$ $\mathcal{G}$ is deterministic and consistent

Background: Review

Probabilistic Circuit $\mathcal{C}$ with circuit structure $\mathcal{G}$

$\mathcal{C}$ is tractable for MAR queries $\iff$ $\mathcal{G}$ is smooth and decomposable

$\mathcal{C}$ is tractable for MAP queries $\iff$ $\mathcal{G}$ is deterministic and consistent

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

Background: Review

Probabilistic Circuit $\mathcal{C}$ with circuit structure $\mathcal{G}$

$\mathcal{C}$ is tractable for MAR queries $\iff$ $\mathcal{G}$ is smooth and decomposable

$\mathcal{C}$ is tractable for MAP queries $\iff$ $\mathcal{G}$ is deterministic and consistent

$\mathcal{G}$ 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

Simple generalization of determinism

Marginal Determinism

Simple generalization of determinism

sum unit is Q-marginal deterministic

$:\Longleftrightarrow$ for all partial states $q \in Q$,

at most one of its inputs is non-zero

Marginal Determinism

Simple generalization of determinism

sum unit is Q-marginal deterministic

$:\iff$ for all partial states $q \in Q$,

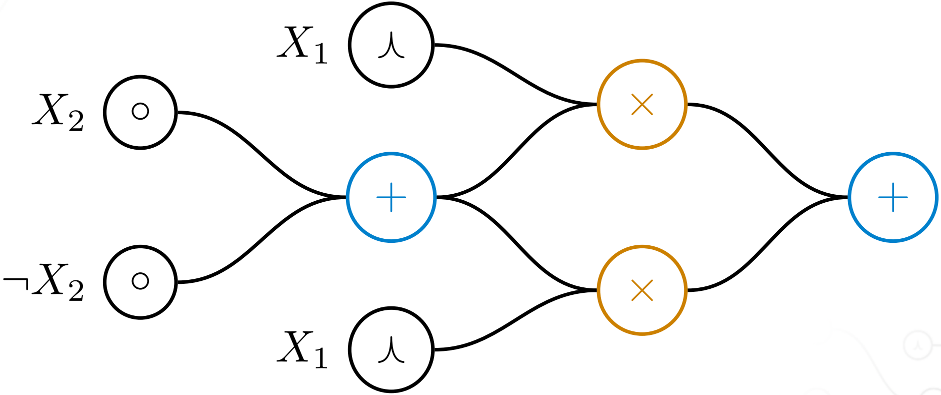
at most one of its inputs is non-zero

$\mathcal{G}$ is Q-marginal deterministic

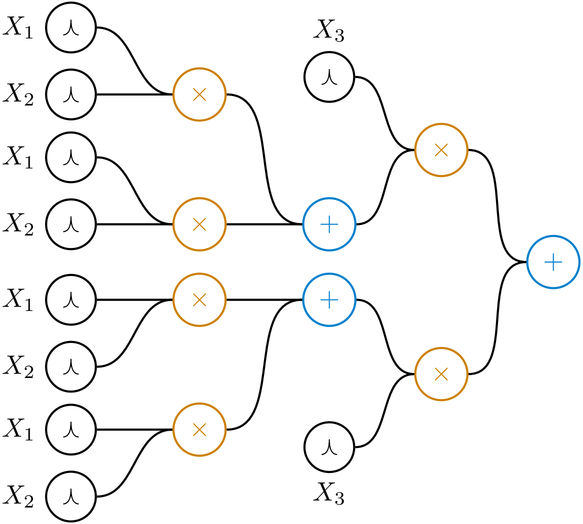
$:\iff$ for all sum units $n \in \mathcal{G}$ with $\varphi(n) \cap Q \neq \emptyset$,

n is marginal deterministic with respect to Q

Marginal Determinism: Example



Marginal Determinism: Example



Tractable Computation

Tractable Computation: General

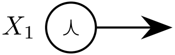
- Adjust feed-forward algorithms for MAR and MAP queries

Tractable Computation: General

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

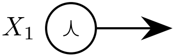
Tractable Computation: General

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



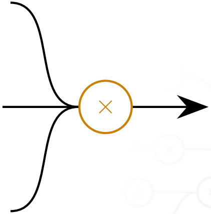
Tractable Computation: General

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

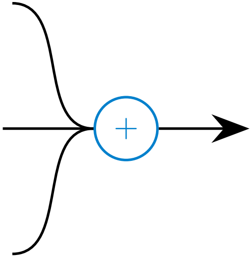


- Product units are handled identically in MAR and MAP

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

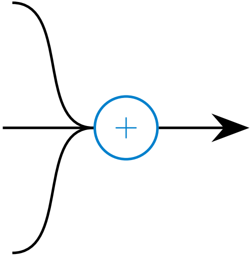


Tractable Computation: Sum Units



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:

$\mathcal{G}$ be smooth, decomposable, and Q-marginal deterministic

⟹ for any parameterization ϑ

the algorithm tractably computes MMAP queries of $\mathcal{C}$ over Q

Sufficient Conditions

Theorem:

$\mathcal{G}$ be smooth, decomposable, and Q-marginal deterministic

$\Longrightarrow$ for any parameterization ϑ

the algorithm tractably computes MMAP queries of $\mathcal{C}$ over Q

- Proof by induction with root node $n \in \mathcal{G}$

Sufficient Conditions

Theorem:

$\mathcal{G}$ be smooth, decomposable, and Q-marginal deterministic

$\implies$ for any parameterization ϑ

the algorithm tractably computes MMAP queries of $\mathcal{C}$ over Q

- Proof by induction with root node $n \in \mathcal{G}$
- Output of input units is correct by assumption

Sufficient Conditions

Theorem:

$\mathcal{G}$ be smooth, decomposable, and Q-marginal deterministic

$\implies$ for any parameterization ϑ

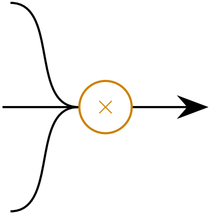
the algorithm tractably computes MMAP queries of $\mathcal{C}$ over Q

- Proof by induction with root node $n \in \mathcal{G}$
- Output of input units is correct by assumption

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

Sufficient Conditions: Product Units

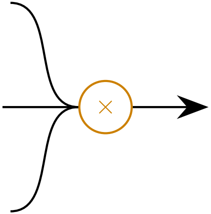
- Apply decomposability and partition Z, Q, and E



Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

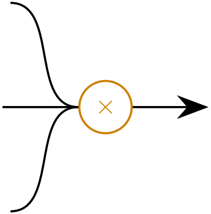
$$\mathcal{Q}(e, \mathcal{J}) = \max_{q_1,\ldots,q_k \in \mathrm{val}(Q)} \int_{\mathcal{J}} \prod_{i=1}^{k} \mathcal{C}_i(Z_i, q_i, e_i)\,\mathrm{d}Z$$



Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

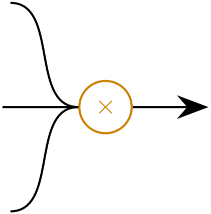
$$\begin{aligned}\mathcal{Q}(e,\mathcal{J}) &= \max_{q_1,\dots,q_k \in \mathrm{val}(Q)} \int_{\mathcal{J}} \prod_{i=1}^{k} \mathcal{C}_i(Z_i, q_i, e_i)\,\mathrm{d}Z \\ &= \max_{q_1,\dots,q_k \in \mathrm{val}(Q)} \prod_{i=1}^{k} \int_{\mathcal{J}} \mathcal{C}_i(Z_i, q_i, e_i)\,\mathrm{d}Z_i\end{aligned}$$



Sufficient Conditions: Product Units

- Apply decomposability and partition Z, Q, and E

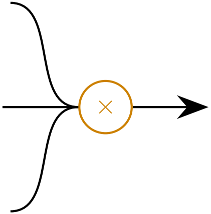
$$
\begin{aligned}
\mathcal{Q}(e, \mathcal{I}) &= \max_{q_1, \ldots, q_k \in \mathrm{val}(Q)} \int_{\mathcal{I}} \prod_{i=1}^{k} \mathcal{C}_i(Z_i, q_i, e_i)\, \mathrm{d}Z \\
&= \max_{q_1, \ldots, q_k \in \mathrm{val}(Q)} \prod_{i=1}^{k} \int_{\mathcal{I}} \mathcal{C}_i(Z_i, q_i, e_i)\, \mathrm{d}Z_i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



Sufficient Conditions: Product Units

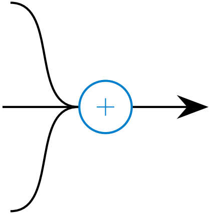
- Apply decomposability and partition Z, Q, and E

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



Sufficient Conditions: Sum Units

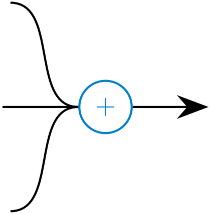
- Sum units with $\varphi(n) \cap Q = \emptyset$ reduce to MAR queries



Sufficient Conditions: Sum Units

- Sum units with $\varphi(n) \cap Q = \emptyset$ reduce to MAR queries

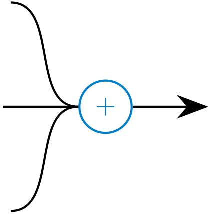
$$\mathcal{Q}(e, \mathcal{I}) = \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z$$



Sufficient Conditions: Sum Units

- Sum units with $\varphi(n) \cap Q = \emptyset$ reduce to MAR queries

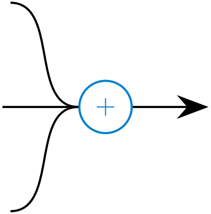
$$\begin{aligned}\mathcal{Q}(e, \mathcal{I}) &= \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{I}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z \\ &= \max_{q \in \mathrm{val}(Q)} \sum_{i \in \mathrm{in}(n)} \int_{\mathcal{I}} \vartheta_i \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z\end{aligned}$$



Sufficient Conditions: Sum Units

- Sum units with $\varphi(n) \cap Q = \emptyset$ reduce to MAR queries

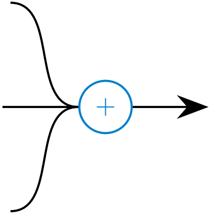
$$\begin{aligned}
\mathcal{Q}(e, \mathcal{J}) &= \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{J}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \sum_{i \in \mathrm{in}(n)} \int_{\mathcal{J}} \vartheta_i \mathcal{C}_i(Z, q, e)\, \mathrm{d}Z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



Sufficient Conditions: Sum Units

- Sum units with $\varphi(n) \cap Q = \emptyset$ reduce to MAR queries

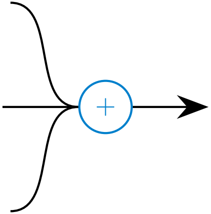
$$\begin{aligned}
\mathcal{Q}(e, \mathcal{J}) &= \max_{q \in \mathrm{val}(Q)} \int_{\mathcal{J}} \sum_{i \in \mathrm{in}(n)} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z \\
&= \max_{q \in \mathrm{val}(Q)} \sum_{i \in \mathrm{in}(n)} \int_{\mathcal{J}} \vartheta_i \mathcal{C}_i(Z, q, e) \, \mathrm{d}Z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



Sufficient Conditions: Sum Units

- Sum units with $\varphi(n) \cap Q = \emptyset$ reduce to MAR queries

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



Conclusions

Conclusions

- MMAP queries are typically be NP-hard

Conclusions

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

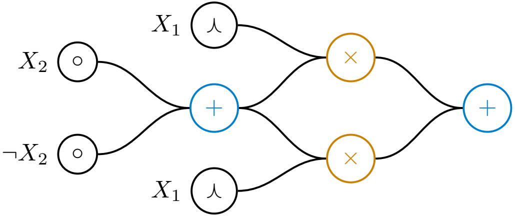
Conclusions

- MMAP queries are typically be NP-hard
- Complexity depends on set of query variables
- Marginal determinism, smoothness, and decomposability are sufficient but strong conditions for tractable computation

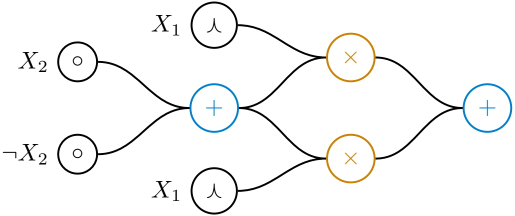
Conclusions

- MMAP queries are typically be NP-hard
- Complexity depends on set of query variables
- Marginal determinism, smoothness, and decomposability are sufficient but strong conditions for tractable computation
- Sum units compute maxima if support contains query variables

Conclusions: Outlook

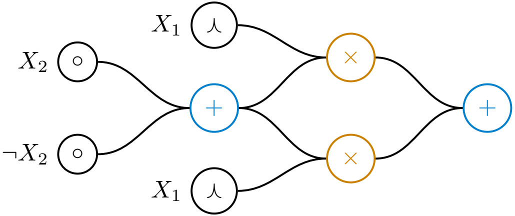


Conclusions: Outlook



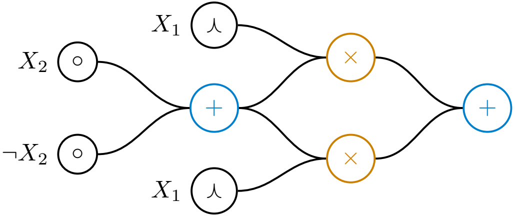
- Marginal determinism for all sets of query variables

Conclusions: Outlook



- Marginal determinism for all sets of query variables
- Sum-maximizer circuits

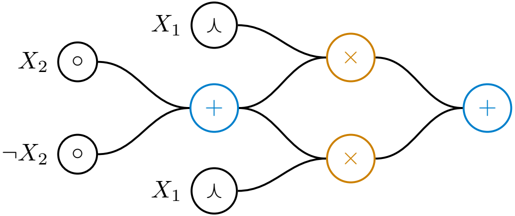
Conclusions: Outlook



- Marginal determinism for all sets of query variables
- Sum-maximizer circuits
- Tractable computation of information-theoretic measures

Conclusions: Outlook

Thank you for Your Attention!



- Marginal determinism for all sets of query variables
- Sum-maximizer circuits
- 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).