Probabilistic Circuits: Marginal Maximum a Posteriori Queries

Markus Pawellek
markus.pawellek@mailbox.org

Abstract

Marginal maximum a posteriori (MMAP) queries combine the aspects of both marginal (MAR) and maximum a posteriori (MAP) inference. Introducing marginal determinism, we are able to provide sufficient conditions for a probabilistic circuit (PC) to be tractable for these queries. To handle MMAP queries in this case, the computation of sum units based on the algorithms for MAR and MAP queries is adjusted.

1 Introduction

This report is a summary of the tractable computation of marginal maximum a posteriori (MMAP) queries for probabilistic circuits (PCs) based on the sections 8.1 and 8.2 of the article "Probabilistic Circuits: A Unifying Framework for Tractable Probabilistic Models" by Choi, Vergari, and Broeck (2020). To facilitate notation, no boldface variables are used.

2 Background

For the scope of this report, we define the following.

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 $Q \subset X$

DEFINITION: (MMAP Query Class)

Define $\mathcal{Q}_{\mathrm{MMAP}}(Q)$ as the set of MMAP queries over Q such that the following holds.

$$\begin{aligned} &\forall \text{ partitions } \{E, Z\} \text{ of } X \setminus Q : \\ &\forall\ e \in E \text{ and intervals } \mathcal{I} \subset \mathrm{val}(Z) : \\ &\exists\ \mathcal{A} \in \mathcal{Q}_{\mathrm{MMAP}}(Q) : \quad \mathcal{A} \text{ computes} \\ &\qquad \operatorname*{arg\,max}_{q \in \mathrm{val}(Q)}\ p\,(Q = q \mid E = e, Z \in \mathcal{I}) \end{aligned}$$

Because maximization is not affected by normalization constants, we can also use the following equivalent expression for the above definition.

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

As a result, the connection to marginal (MAR) and maximum a posteriori (MAP) queries can be seen.

$$\begin{aligned} &\textbf{case } Q = \emptyset : \quad \mathcal{Q}_{\mathrm{MMAP}}(Q) = \mathcal{Q}_{\mathrm{MAR}} \\ &\textbf{case } Q = X : \quad \mathcal{Q}_{\mathrm{MMAP}}(Q) \subset \mathcal{Q}_{\mathrm{MAP}} \end{aligned}$$

Now, recapitulate the characterizing properties for a PC to be tractable for those queries.

$$\begin{aligned} &\mathcal{C} \text{ is tractable for MAR queries} \\ &\quad \iff \mathcal{G} \text{ is decomposable and smooth} \end{aligned}$$

$$\begin{aligned} &\mathcal{C} \text{ is tractable for MAP queries} \\ &\quad \iff \mathcal{G} \text{ is consistent and deterministic} \end{aligned}$$

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

3 Tractable Computations of MMAP Queries

DEFINITION: (Marginal Determinism)

A sum node $n \in \mathcal{G}$ with inputs r_c for $c \in \mathrm{in}(n)$ is Q-marginal deterministic if

$$\begin{aligned} \forall q \in \mathrm{val}(Q) : &\ \forall c \in \mathrm{in}(n) : r_c = 0 \\ &\quad \vee\ \exists! c \in \mathrm{in}(n) : r_c \neq 0 \end{aligned}$$

If this holds for all sum nodes $n \in \mathcal{G}$ with $\varphi(n) \cap Q \neq \emptyset$, also $\mathcal{G}$ is called Q-marginal deterministic.

Marginal determinism is a simple generalization of determinism. Based on this definition, it now becomes clear, how to use the MAP and MAR algorithms to tractably compute MMAP queries given $\mathcal{G}$ being smooth, decomposable, and Q-marginal deterministic.

 © Markus Pawellek, January 21, 2022

As always, we assume that an input unit already computes the given query correctly. Concerning the output of product units, the computations for MAR and MAP queries are identical and only involve the multiplication of their inputs. As a result, product units shall be treated the same way for this algorithm. For sum units on the other hand, a decision has to be made based on whether their scope intersects with Q.

Let $n \in \mathcal{G}$ be a sum unit with output r_n, inputs r_c, and weights ϑ_{nc} for $c \in \mathrm{in}(n)$ in a feed-forward traversal of $\mathcal{C}$. Then its result r_n can be computed by the following snippet.

Sum Unit Computation for MMAP Algorithm

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

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

else

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

Remember from the MAP algorithm, that at the end of the whole algorithm a backward pass is needed to retrieve the modes of the input distributions that contributed to the output.

THEOREM: (Sufficient Conditions)

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:
Let E and Z be a partition of $X \setminus Q$. Furthermore, let $e \in E$ be a partial state, $\mathcal{I} \in \mathrm{val}(Z)$ an interval, and $\mathcal{Q}(e, \mathcal{I})$ be the MMAP query. Then the statement is proved by induction.

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

The root node $n \in \mathcal{G}$ can be one of three types: input unit, sum unit, or product unit. By definition, input units already output the correct values for $\mathcal{Q}(e, \mathcal{I})$.
Due to decomposability, for product units, we are allowed to assume that Z, Q, and E are partitioned into Z_i, Q_i, and E_i for $i, k \in \mathbb{N}$ and $i \leq k$, respectively.

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

Sum units, whose scope contains no query variable, only compute MAR queries. Their correct behavior was already proven. So, assume n to be a sum unit with $\varphi(n) \cap Q \neq \emptyset$.

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

For this, smoothness and Q-marginal determinism has been used. Applying these equations recursively down to input units, this concludes the proof. □

4 Conclusions

The computation complexity of the set of marginal MMAP queries may typically be NP-hard and is highly dependent on the set of query variables. For examining the efficient evaluation of MMAP queries for PCs, the structural property of marginal determinism was defined. Together with smoothness and decomposability, these are sufficient conditions to make PCs tractable for MMAP queries. Nevertheless, using the more general structure of sum-maximizer circuits, it can be shown that these properties are indeed not necessary.

As an outlook: In contrast to other structural properties, marginal determinism was defined with respect to its query set. Assuming marginal determinism for all possible subsets of X, $\mathcal{C}$ cannot contain a full-support distribution without it to be fully factorized. The family of such PCs thus is not expressive and would therefore impose a too strong restriction.

References

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

© Markus Pawellek, January 21, 2022