Introduction

We prove that every Sum-Product Network (SPN) can be converted into a Bayesian Network (BN) in linear time and space.
The generated BN has a simple directed bipartite graphical structure with Algebraic Decision Diagrams (ADDs) as representations of the local probability distributions.
Applying the Variable Elimination algorithm (VE) to the generated BN will recover the original SPN.
We introduce normal SPN and present a theoretical analysis of the consistency and decomposability properties.

Our Results

Normal SPN:
• Complete and decomposable.
• Locally normalized weights.

Theorem 1:
For any complete and consistent SPN \( \mathcal{S} \), there exists a normal SPN \( \mathcal{S}' \) such that \( \text{Pr}_{\mathcal{S}}(\cdot) = \text{Pr}_{\mathcal{S}'}(\cdot) \) and \( |\mathcal{S}'| = O(|\mathcal{S}|^2) \).

Background

Definition (Poon and Domingos):
• A rooted DAG with indicator variables as leaves and sum nodes, product nodes as internal nodes.
• Edges from sum nodes are associated with nonnegative weights.
• The value of a product node is the product of the values of its children. The value of a sum node is the weighted sum of the values of its children. The value of an SPN is the value of its root.

Scope: The set of variables that have indicators among the node’s descendents.
Complete: An SPN is complete iff each sum node has children with the same scope.
Consistent: An SPN is consistent iff no variable appears negated in one child of a product node and non-negated in another.

Decomposable: An SPN is decomposable iff for every product node \( v \), \( \text{scope}(v) \cap \text{scope}(v_j) = \emptyset \) where \( v, v_j \in \text{CN}(v), i \neq j \).

Algebraic Decision Diagram: A graphical representation a real function with boolean input variables, where the graph is a rooted DAG.

Remark 5:
Assuming sum nodes alternate with product nodes in SPN \( \mathcal{S} \), the depth of \( \mathcal{S} \) is proportional to the maximum in-degree of the nodes in \( \mathcal{B} \), which, as a result, is proportional to a lower bound of the tree-width of \( \mathcal{B} \).

Theorem 6 (BN to SPN):
Given the BN \( \mathcal{B} \) with ADD representation of CPDs generated from a complete and decomposable SPN \( \mathcal{S} \) over Boolean variables \( X_{1,N} \), the original SPN \( \mathcal{S} \) can be recovered by applying the Variable Elimination algorithm to \( \mathcal{B} \) in \( O(|\mathcal{N}|^2) \).

Corollary 3:
There exists an algorithm that converts any complete and consistent SPN \( \mathcal{S} \) over Boolean variables \( X_{1,N} \) into a BN \( \mathcal{B} \) with CPDs represented by ADDs in time \( O(|\mathcal{N}|) \). Furthermore, \( \mathcal{S} \) and \( \mathcal{B} \) represent the same distribution and \( |\mathcal{B}| = O(|\mathcal{N}|) \).

Remark 4:
The BN \( \mathcal{B} \) generated from \( \mathcal{S} \) has a simple bipartite DAG structure, where all the source nodes are hidden variables and the terminal nodes are the Boolean variables \( X_{1,N} \).

Conclusion

• The CSI among variables helps to reduce the inference complexity to enable efficient exact inference even for graphical models with large tree-width.
• There may exist other techniques to convert an SPN into a BN with a more compact representation and also a smaller tree-width.
• Structure and parameter learning for SPNs can benefit from the Bayesian network perspective.
• The analysis showed in this paper can be straightforwardly applied to analyze the relationship between SPNs and Markov networks.