On the Relationship between Sum-Product Networks and Bayesian Networks

Han Zhao, Mazen Melibari and Pascal Poupart
Presented by: Han Zhao

UNIVERSITY OF WATERLOO

July 8th, 2015
Outline

Motivation

Background
  Sum-Product Network
  Algebraic Decision Diagram

Main Result
  Sum-Product Network to Bayesian Network
  Main Theorems
  Bayesian Network to Sum-Product Network

Summary
Motivation

- How to convert a Sum-Product network into a Bayesian network?
- Will the conversion lead to an exponential blow-up?
Sum-Product Network

Definition

A Sum-Product Network is a
- Rooted directed acyclic graph of indicator variables, sum nodes and product nodes.
- Value of a product node is the product of its children.
- Value of a sum node is the weighted sum of its children, where the weights are nonnegative.

\[
\begin{align*}
\mathbb{I}_{x_1} & \times 6 \quad \mathbb{I}_{\bar{x}_1} \\
\mathbb{I}_{x_2} & \times 6 \quad \mathbb{I}_{\bar{x}_2}
\end{align*}
\]
Sum-Product Network

Network Polynomial

\[
f(X_1, X_2) = 10(6I_{x_1} + 4I_{\overline{x}_1})(6I_{x_2} + 14I_{\overline{x}_2}) + 6(6I_{x_1} + 4I_{\overline{x}_1})(2I_{x_2} + 8I_{\overline{x}_2}) + 9(9I_{x_1} + I_{\overline{x}_1})(2I_{x_2} + 8I_{\overline{x}_2})
\]

\[
= 594I_{x_1}I_{x_2} + 1776I_{x_1}I_{\overline{x}_2} + 306I_{\overline{x}_1}I_{x_2} + 824I_{\overline{x}_1}I_{\overline{x}_2}
\]
Joint/Marginal/Conditional queries can be answered in linear time in Sum-Product Network.

Marginal Inference

\[ \Pr(X_1 = 1) \]

\[
f(X_1, X_2) = 594\mathbb{I}_{x_1}\mathbb{I}_{\overline{x_2}} + 1776\mathbb{I}_{x_1}\mathbb{I}_{\overline{x_2}} + 306\mathbb{I}_{\overline{x_1}}\mathbb{I}_{x_2} + 824\mathbb{I}_{\overline{x_1}}\mathbb{I}_{\overline{x_2}}
\]
Sum-Product Network

Inference

Joint/Marginal/Conditional queries can be answered in linear time in Sum-Product Network.

Marginal Inference

\[ \Pr(X_1 = 1) \] Setting \( \mathbb{I}_{x_1} = 1, \mathbb{I}_{\bar{x}_1} = 0, \mathbb{I}_{x_2} = 1, \mathbb{I}_{\bar{x}_2} = 1. \)

\[
f(X_1, X_2) = 594\mathbb{I}_{x_1}\mathbb{I}_{x_2} + 1776\mathbb{I}_{x_1}\mathbb{I}_{\bar{x}_2} + 306\mathbb{I}_{\bar{x}_1}\mathbb{I}_{x_2} + 824\mathbb{I}_{\bar{x}_1}\mathbb{I}_{\bar{x}_2}
\]
Sum-Product Network

Inference

Joint/Marginal/Conditional queries can be answered in linear time in Sum-Product Network.

Marginal Inference

Pr($X_1 = 1$) ? Setting $I_{x_1} = 1$, $I_{\bar{x}_1} = 0$, $I_{x_2} = 1$, $I_{\bar{x}_2} = 1$.

\[
f(X_1, X_2) = 594I_{x_1}I_{x_2} + 1776I_{x_1}I_{\bar{x}_2} + 306I_{\bar{x}_1}I_{x_2} + 824I_{\bar{x}_1}I_{\bar{x}_2}
\]
Sum-Product Network

Inference

Joint/Marginal/Conditional queries can be answered in linear time in Sum-Product Network.

Marginal Inference

\[ \Pr(X_1 = 1) \]

Setting \( \mathbb{I}_{x_1} = 1, \mathbb{I}_{\bar{x}_1} = 0, \mathbb{I}_{x_2} = 1, \mathbb{I}_{\bar{x}_2} = 1 \).

\[
f(X_1, X_2) = 594 \mathbb{I}_{x_1} \mathbb{I}_{x_2} + 1776 \mathbb{I}_{x_1} \mathbb{I}_{\bar{x}_2} + 306 \mathbb{I}_{\bar{x}_1} \mathbb{I}_{x_2} + 824 \mathbb{I}_{\bar{x}_1} \mathbb{I}_{\bar{x}_2}
\]
Sum-Product Network

Inference

Joint/Marginal/Conditional queries can be answered in linear time in Sum-Product Network.

Marginal Inference

\[ \Pr(X_1 = 1) \] Setting \( \mathbb{I}_{x_1} = 1, \mathbb{I}_{\bar{x}_1} = 0, \mathbb{I}_{x_2} = 1, \mathbb{I}_{\bar{x}_2} = 1. \]

\[
f(X_1, X_2) = 594\mathbb{I}_{x_1}\mathbb{I}_{x_2} + 1776\mathbb{I}_{x_1}\mathbb{I}_{\bar{x}_2} + 306\mathbb{I}_{\bar{x}_1}\mathbb{I}_{x_2} + 824\mathbb{I}_{\bar{x}_1}\mathbb{I}_{\bar{x}_2}
\]
Sum-Product Network

Inference

Joint/Marginal/Conditional queries can be answered in linear time in Sum-Product Network.

Marginal Inference

\[ \Pr(X_1 = 1) \]

Setting \( \mathbb{I}_{x_1} = 1, \mathbb{I}_{\bar{x}_1} = 0, \mathbb{I}_{x_2} = 1, \mathbb{I}_{\bar{x}_2} = 1. \)

\[
\Pr(X_1 = 1) = \frac{2370}{594+1776+306+824} = \frac{2370}{3500}
\]
Definition (scope)

The scope of a node in an SPN is defined as the set of variables that have indicators among the node’s descendants: For any node $v$ in an SPN, if $v$ is a terminal node, say, an indicator variable over $X$, then $\text{scope}(v) = \{X\}$, else $\text{scope}(v) = \bigcup \tilde{v} \in \text{Ch}(v) \text{scope}(\tilde{v})$.

\[
\text{scope}(I_{x_1}) = \text{scope}(I_{\overline{x}_1}) = \{X_1\}
\]

\[
\text{scope}(I_{x_2}) = \text{scope}(I_{\overline{x}_2}) = \{X_2\}
\]

\[
\text{scope}(\otimes) = \{X_1, X_2\}
\]

\[
\text{scope}(\text{Root}) = \{X_1, X_2\}
\]
Complete and Consistent

Definition (Complete)
An SPN is complete iff each sum node has children with the same scope.

Definition (Consistent)
An SPN is consistent iff no variable appears negated in one child of a product node and non-negated in another.

Definition (Valid)
An SPN is said to be valid iff it defines a (unnormalized) probability distribution.
Sum-Product Network
Probability semantics

Theorem (Poon and Domingos)

*If an SPN* \( S \) *is complete and consistent, then it is valid.*

\[
f(X_1, X_2) = 594 \mathbb{I}_{x_1} \mathbb{I}_{x_2} + 1776 \mathbb{I}_{x_1} \mathbb{I}_{\bar{x}_2} + 306 \mathbb{I}_{\bar{x}_1} \mathbb{I}_{x_2} + 824 \mathbb{I}_{\bar{x}_1} \mathbb{I}_{\bar{x}_2}
\]

\[
\Pr_S(X_1, X_2) = \frac{1}{3500} f(X_1, X_2)
\]

Definition (Decomposable)

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

Decomposability induces consistency.
Algebraic Decision Diagram Representation

Algebraic Decision Diagram

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$f(\cdot)$</th>
<th>$X_1$</th>
<th>$X_2$</th>
<th>$X_3$</th>
<th>$X_4$</th>
<th>$f(\cdot)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.4</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

- When $X_1 = 0$, $f(\cdot)$ is independent of the value taken by $X_2$.
- When $X_1 = X_2 = 1$, $f(\cdot)$ is independent of $X_3$, $X_4$. 
Road Map

1. Define *Normal SPN*, a sub-class of SPN.
Road Map

1. Define Normal SPN, a sub-class of SPN.
2. Show any SPN can be converted into its normal form in quadratic time and space.
1. Define *Normal SPN*, a sub-class of SPN.
2. Show any SPN can be converted into its normal form in quadratic time and space.
3. Use normal SPNs as a bridge to show the relationship between SPNs and BNs.
Definition (Normal Sum-Product Network)

An SPN is said to be normal if

1. It is complete and decomposable.
2. For each sum node in the SPN, the weights of the edges emanating from the sum node are nonnegative and sum to 1.
Definition (Normal Sum-Product Network)

An SPN is said to be normal if

1. It is complete and decomposable.
2. For each sum node in the SPN, the weights of the edges emanating from the sum node are nonnegative and sum to 1.

Theorem (Normal Transformation)

For any complete and consistent SPN $S$, there exists a normal SPN $S'$ such that $\text{Pr}_S(\cdot) = \text{Pr}_{S'}(\cdot)$ and $|S'| = O(|S|^2)$. 
Given a normal SPN $S$ over $X_{1:N}$, construct:

- A hidden node $H_v$ for each sum node $v$ in $S$. 
Given a normal SPN $S$ over $X_{1:N}$, construct:

- A hidden node $H_v$ for each sum node $v$ in $S$.
- An observable node $X_n$ for each variable $X_n$ in $S$. 

\begin{center}
\begin{tikzpicture}[node distance = 1.5cm, thick, main node/.style = {circle, draw, font = \normalsize},
  sum node/.style = {main node, fill = white},
  hidden node/.style = {main node, fill = white},
  observable node/.style = {main node, fill = white}]

  \node[sum node] (sum1) {+};
  \node[hidden node] (hidden1) [below left of = sum1] {H};
  \node[observable node] (obs1) [below right of = sum1] {$X_1$};
  \node[observable node] (obs2) [below right of = hidden1] {$X_1$};
  \node[observable node] (obs3) [below right of = obs1] {$X_2$};
  \node[observable node] (obs4) [below right of = obs2] {$X_2$};

  \path[->] (hidden1) edge node {} (sum1);
  \path[->] (sum1) edge node {} (obs1);
  \path[->] (sum1) edge node {} (obs2);
  \path[->] (sum1) edge node {} (obs3);
  \path[->] (sum1) edge node {} (obs4);

\end{tikzpicture}
\end{center}
Given a normal SPN $S$ over $\mathbf{X}_{1:N}$, construct:

- A hidden node $H_v$ for each sum node $v$ in $S$.
- An observable node $X_n$ for each variable $X_n$ in $S$.
- A directed from $H_v$ to $X_n$ iff $X_n$ appears in the sub-SPN rooted at $v$ in $S$. 

![Diagram](image-url)
Given a normal SPN $S$ over $X_{1:N}$, construct:

- A decision stump from the sum node $v$ for each hidden node $H_v$. 

```
(0.6, 0.4)  (0.9, 0.1)  (0.3, 0.7)  (0.2, 0.8)
×     ×     ×     ×
X_1  X_1  X_2  X_2
```

```
+  +  +  +
4/7  6/35  9/35
```

```
A_H =
```

```
H
```

```
H
```

```
4/7
h_1  h_2  h_3
```

```
H
```

```
H
```

```
X_1  X_2
```

```
X_1
```

```
X_2
```

```
X_1
```

```
X_2
```

```
X_1
```

```
X_2
```

Given a normal SPN $S$ over $X_{1:N}$, construct:

- A decision stump from the sum node $v$ for each hidden node $H_v$.
- An induced sub-SPN $S_{X_n}$ by node set $\{X_n\}$ from $S$ and then contract all the product nodes in $S_{X_n}$. 
Given a normal SPN \( S \) over \( X_{1:N} \), construct:

- A decision stump from the sum node \( v \) for each hidden node \( H_v \).
- An induced sub-SPN \( S_{X_n} \) by node set \( \{X_n\} \) from \( S \) and then contract all the product nodes in \( S_{X_n} \).
Let $|S|$ be the size of the SPN, i.e., the number of nodes plus the number of edges in the graph. For a BN $B$, the size of $B$, $|B|$, is defined by the size of the graph plus the size of all the CPDs in $B$. 

**Theorem (SPN-BN)**

There exists an algorithm that converts any complete and decomposable SPN $S$ over Boolean variables $X_1:N$ into a BN $B$ with CPDs represented by ADDs in time $O(N|S|)$. Furthermore, $S$ and $B$ represent the same distribution and $|B| = O(N|S|)$.

**Corollary (SPN-BN)**

There exists an algorithm that converts any complete and consistent SPN $S$ over Boolean variables $X_1:N$ into a BN $B$ with CPDs represented by ADDs in time $O(N^2|S|)$. Furthermore, $S$ and $B$ represent the same distribution and $|B| = O(N|S|^2)$. 


Main Theorems

SPN-BN

Let $|S|$ be the size of the SPN, i.e., the number of nodes plus the number of edges in the graph. For a BN $B$, the size of $B$, $|B|$, is defined by the size of the graph plus the size of all the CPDs in $B$.

**Theorem (SPN-BN)**

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

SPN-BN

Let \(|S|\) be the size of the SPN, i.e., the number of nodes plus the number of edges in the graph. For a BN \(B\), the size of \(B\), \(|B|\), is defined by the size of the graph plus the size of all the CPDs in \(B\).

**Theorem (SPN-BN)**

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

**Corollary (SPN-BN)**

There exists an algorithm that converts any complete and consistent SPN \(S\) over Boolean variables \(X_{1:N}\) into a BN \(B\) with CPDs represented by ADDs in time \(O(N|S|^2)\). Furthermore, \(S\) and \(B\) represent the same distribution and \(|B| = O(N|S|^2)\).
Remark
The BN $B$ generated from $S$ has a simple bipartite DAG structure, where all the source nodes are hidden variables and the terminal nodes are the Boolean variables $X_{1:N}$.
BN-SPN
Algorithm

Extend Algebraic Decision Diagram to Symbolic Algebraic Decision Diagram where $+,-,\times,/$ are allowed to be internal nodes.
BN-SPN

Algorithm

Extend Algebraic Decision Diagram to Symbolic Algebraic Decision Diagram where $\, +, -, \times, / \,$ are allowed to be internal nodes.

Define two operations in symbolic ADD:

- *Multiplication* between pairs of symbolic ADDs
- *Summing Out* one internal variable in symbolic ADD

Theorem (BN-SPN)

There exists a variable ordering such that applying Variable Elimination with the ordering to BN with ADDs builds the original SPN $S$ in $O(|S|)$. 
BN-SPN

Algorithm

Extend Algebraic Decision Diagram to *Symbolic* Algebraic Decision Diagram where $+, -, \times, /$ are allowed to be internal nodes.

Define two operations in symbolic ADD:

- *Multiplication* between pairs of symbolic ADDs
- *Summing Out* one internal variable in symbolic ADD

**Theorem (BN-SPN)**

There exists a variable ordering such that applying Variable Elimination with the ordering to BN with ADDs builds the original SPN $S$ in $O(N |S|)$. 

![Diagram](image.png)
Remark
The combination of the above two theorems shows that distributions for which SPNs allow a compact representation and efficient inference, BNs with ADDs also allow a compact representation and efficient inference (i.e., no exponential blow up).
Summary

- How to convert a Sum-Product network into a Bayesian network?

- Will the conversion lead to a exponential blow-up?
Summary

- How to convert a Sum-Product network into a Bayesian network?
  We provide a constructive algorithm to convert any complete and consistent SPNs into bipartite BNs with ADD representations.

- Will the conversion lead to an exponential blow-up?
Summary

- How to convert a Sum-Product network into a Bayesian network?
  We provide a constructive algorithm to convert any complete and consistent SPNs into bipartite BNs with ADD representations.

- Will the conversion lead to an exponential blow-up?
  Both the time complexity for the construction and the space complexity for the constructed BN can be bounded by a linear function of $|S|$.
Summary

- How to convert a Sum-Product network into a Bayesian network?
  We provide a constructive algorithm to convert any complete and consistent SPNs into bipartite BNs with ADD representations.

- Will the conversion lead to an exponential blow-up?
  Both the time complexity for the construction and the space complexity for the constructed BN can be bounded by a linear function of $|S|$.

- SPNs and BNs with ADDs share the same representational power.
- SPNs with any depth $\equiv$ directed bipartite BN.
- SPNs are *history recording* or *caching* of the inference process on BN.
Thanks

full version: arXiv:1501.01239