Supplementary Material for Learning Sum-Product Networks with Direct and Indirect Variable Interactions

Amirmohammad Rooshenas PEDRAM@CS.UOREGON.EDU
Daniel Lowd LOWD@CS.UOREGON.EDU
Department of Computer and Information Science, University of Oregon

In this document, we provide the proofs for the propositions stated in the paper.

Proposition 1. *For discrete domains, every decomposable and smooth AC can be represented as an equivalent SPN with fewer or equal nodes and edges.*

Proof. We show this constructively. Given an AC, we can convert it to a valid SPN representing the same function in four steps:

1. If the root is not a sum node, set the root to a new sum node whose single child is the former root.
2. For each sum node, set the initial weights of all outgoing edges to 1.
3. For each parameter node, find the first sum node on each path to the root and multiply its outgoing edge weight along that path by the parameter value. (Do not multiply the same edge weight by any given parameter more than once, even if that edge occurs in multiple paths to the root. We assume that parameter nodes only occur as children of product nodes.)
4. Remove all parameter nodes from the network.
5. Replace each indicator node $\lambda_{X_i=v}$ with a deterministic univariate distribution, $P(X_i = v) = 1$.

Because the AC was decomposable, each product in the resulting SPN must be over disjoint scopes. Because the AC was smooth, each sum in the resulting SPN must be over identical scopes. Therefore, the SPN is valid. Since all indicator nodes are removed and at most one new node is added, the new SPN must have fewer or equal nodes and edges than the original AC.

To prove that the SPN evaluates to the same function as the original AC, we use induction to show that each sum node evaluates to the same value as before. Since the root node is a sum, this suffices to show that the new SPN is equivalent.

Consider each outgoing edge from the sum node to one of its children. There are three cases to consider:

- If the child is a leaf, then the child's value is a deterministic distribution which is clearly identical to the parameter node in AC. The edge weight will be 1, since the leaf could not have had any parameter node descendants that were removed.
- If the child is another sum node, then by the inductive hypothesis, its value must be the same as in the AC. The weight of this edge must also be 1, since any parameter node descendant must have at least one sum node that is "closer," namely the child sum node.
- If the child is a product node, then its value might be different from the AC. Without loss of generality, consider a product node and all of its product node children, and all of their product node children, etc. together as a single product. (This is valid because multiplication is commutative and associative.) The elements in this product are only sum nodes and leaf nodes, both of which have the same values as in the AC. One or more parameter nodes could have been removed from this product when constructing the SPN. These parameters have been incorporated into the edge weight, since the parent sum node is the first sum node on any path from the parameters to the root that passes through this edge. Therefore, the value of the product node times its edge weight is equal to the value of the product node in the AC.

Thus, each child of the sum node has the same value as in the original AC once it has been multiplied by the edge weight, so the sum node computes the same value as in the AC. For the base case of a sum node with no sum node descendants, the above arguments still suffice and no longer depend on the inductive hypothesis. Therefore, by structural induction, every sum node computes the same value as in the AC.

□

Proposition 2. *For discrete domains, every SPN can be represented as an AC with at most a linear increase in the number of edges.*

Proof. We again show this constructively. Given an SPN, we can convert it to a decomposable and smooth AC representing the same distribution in three steps:

1. Create indicator nodes for each variable/value combination.
2. Replace each univariate distribution with a sum of products of parameter and indicator nodes. Specifically, create one parameter node for each state of the target variable, representing the respective probability of that state. Create one product node for each state with the corresponding indicator node and new parameter node as children.
3. Replace each outgoing edge of each sum node with a product of the original child and a new parameter node representing the original weight from that edge.

Assuming the domain of each variable is bounded by a constant, each edge is replaced by at most a constant number of edges. Therefore, the number of edges in the resulting AC is linear in the number of edges in the original SPN.

We now use induction to show that, for each node in the SPN, there is a node in the AC that computes the same value. We have three types of nodes to consider:

- As the base case, each leaf in the SPN is represented by a new sum node created in the second step. By construction, this sum node clearly represents the same value as the leaf distribution in the SPN.
- Each product node in the SPN is unchanged in the AC conversion. Since, by the inductive hypothesis, its children compute the same values as in the SPN, so does the product.
- Each sum node in the SPN is represented by a similar node in the AC. The AC node is an unweighted sum over products of the SPN edge weights and nodes representing the corresponding SPN children. By the inductive hypothesis, these children compute the same values as their SPN counterparts, so the weighted sum performed by the AC node (with the help of new product and parameter nodes) is identical to the SPN sum node.

The root of the AC represents the root of the SPN, so by induction, they compute the same value and the two models represent the same distribution. □