Learning and Inference in Tractable Probabilistic Knowledge Bases

SUPPLEMENTARY MATERIAL

Proof of Theorem 1

Proof. We provide a constructive proof, describing the architecture of an SPN constructed from any given TPKB. We prove that (a) the SPN's distribution is that of the TPKB and (b) the SPN is valid, that is, it computes all marginals and the partition function correctly. Finally, we show that the size of the SPN is linear in the size of the TPKB. By Theorem 1 of Gens and Domingos [9], which applies to all valid SPNs, we can conclude that the partition function of a TPKB can be computed in time linear in its size.

The proof consists of three parts.

(i) We first construct a subSPN $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ for every pair $(\mathtt{O}, \mathtt{C})$ where $\mathtt{O}$ is a leaf in the part decomposition, $\mathtt{C}$ a leaf in the class hierarchy, and $\mathtt{O}$ a possible instance of $\mathtt{C}$. We show that each of these subSPNs is valid and computes the weight of subworlds and the corresponding partition function correctly.

(ii) We construct a subSPN $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ for every pair $(\mathtt{O}, \mathtt{C})$ where $\mathtt{O}$ is a leaf in the part decomposition and $\mathtt{C}$ is the class that object $\mathtt{O}$ was declared to be an instance of. We show that each of these subSPNs is valid and computes the weight of subworlds and the corresponding partition function correctly. This is accomplished via a proof by induction on the height of a tree (a subtree of the class hierarchy) of which $\mathtt{C}$ is the root node.

(iii) Finally, we construct a subSPN for the pair $(\mathtt{O}_0, \mathtt{C}_0)$ and show that it is valid and computes the weight of subworlds correctly. This is accomplished by a proof of induction on the height of the part decomposition.

We begin by proving part (i). We denote by $\mathbf{L}_i$ the i-th layer of the part decomposition where i is the height of the layer. Hence, the leaf nodes of the part decomposition are in layer $\mathbf{L}_0$. Moreover, let $\mathbf{C}_0$ be the set of leaf classes of the class hierarchy the object $\mathtt{O}$ can possibly be an instance of. Let $\mathtt{O} \in \mathbf{L}_0$ and $\mathtt{C} \in \mathbf{C}_0$. For each such pair, we construct a subSPN $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ as follows. We create a product node with two product nodes as children. One child represents the product of the attribute distributions. For each attribute declared for class $\mathtt{C}$ there is one sum node representing the efficiently summable/integrable attribute distribution with the (implicitly given) indicator variables $\mathtt{A}(\mathtt{O}, \mathtt{D})$ and $\neg\mathtt{A}(\mathtt{O}, \mathtt{D})$. For each relation $\mathtt{R}$ declared for class $\mathtt{C}$, there is one sum node representing the relation distribution with indicator variables $\mathtt{R}(\mathtt{O})$ and $\neg\mathtt{R}(\mathtt{O})$. Since $\mathtt{O}$ does not have parts, these indicator variables represent atoms of unary predicates. Finally, we add the indicator variable $\mathtt{Is}(\mathtt{O}, \mathtt{C})$ as a child node to the top product node. Figure 3 depicts the subSPN for a pair $(\mathtt{O}, \mathtt{C})$. Here, we do not need the depicted product nodes representing the subclass and part distributions because leaf objects do not have parts and leaf classes do not have subclasses. It is now straighforward to verify that the subSPN $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ computes $\phi(\mathtt{O}, \mathtt{C}, \mathbf{W})$ for every possible subworld $\mathbf{W}$ with top object $\mathtt{O}$ and top class $\mathtt{C}$. Since each possible world determines the truth value of all indicator variables, the SPN also computes the partition function $Z_{(\mathtt{O},\mathtt{C})}$ correctly. Moreover, it is straightforward to verify that the subSPN is valid by showing that it is complete and decomposable [27]. This concludes the first part of the proof.

We now proceed to prove part (ii). We construct subSPNs $\mathsf{SPN}(\mathtt{O}, \mathtt{C}_0)$ for every pair $(\mathtt{O}, \mathtt{C}_0)$ where $\mathtt{O}$ is a leaf in the part decomposition and $\mathtt{C}_0$ is a class that $\mathtt{O}$ is declared to be an instance of. We accomplish this with an inductive proof on the height of the subtree of the class hierarchy with $\mathtt{C}_0$ as root node. We denote this subtree as the *class tree* of the pair $(\mathtt{O}, \mathtt{C}_0)$ and write $\mathtt{Tree}_{(\mathtt{O},\mathtt{C}_0)}$. In the preceding step of the main proof, we showed the bases case, that is, for all $\mathtt{C} \in \mathtt{Tree}_{(\mathtt{O},\mathtt{C}_0)}$ with height 0 we constructed a valid SPN that computes the weight of subworlds and the corresponding partition function correctly. Let class $\mathtt{C}$ be a node in $\mathtt{Tree}_{(\mathtt{O},\mathtt{C}_0)}$ and let the height of $\mathtt{C}$ be $h > 0$. The inductive hypothesis is that, for all classes $\mathtt{C}'$ in $\mathtt{Tree}_{(\mathtt{O},\mathtt{C}_0)}$ with height smaller than h, the subSPNs $\mathsf{SPN}(\mathtt{O}, \mathtt{C}')$ are valid and compute the weight of subworlds and the partition function correctly.

Let $S_1, \ldots, S_n$ be the direct subclasses declared for class $\mathtt{C}$ with height 0. We now introduce a new product node with **four child nodes**.

1. A **sum node** evaluating the weighted sum over the SPNs $\mathsf{SPN}(\mathtt{O}, \mathtt{S}_i), 1 \leq i \leq n$. Whenever there are attributes [relations] declared for class $\mathtt{S}_i$ that are not

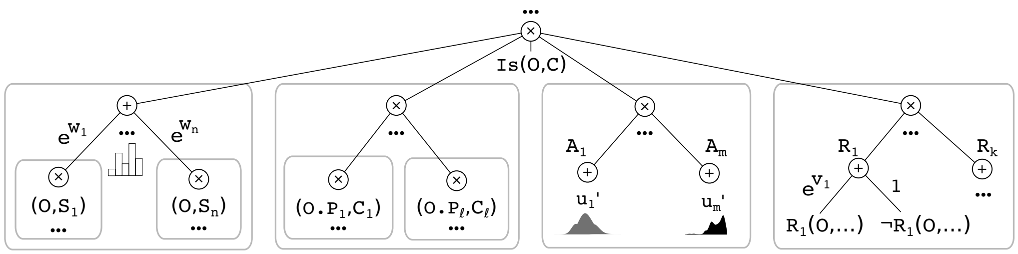

Figure 3: The partial SPN corresponding to an object O being an instance of a generic class C.

declared for $\mathtt{S}_j$, $1 \leq i \neq j \leq n$, we make the indicator variables $\neg\mathtt{A}(\mathtt{O}, ...)[\neg\mathtt{R}(\mathtt{O}, \ldots)]$ children of the top node of $\mathsf{SPN}(\mathtt{O}, \mathtt{S}_j)$. Moreover, for every class $\mathtt{S}_i$, we make the indicator variable $\neg\mathtt{Is}(\mathtt{O}, \mathtt{S}_j)$ a child of the top node of $\mathsf{SPN}(\mathtt{O}, \mathtt{S}_i)$, for every $i \neq j$. These steps do not change the validity of $\mathsf{SPN}(\mathtt{O}, \mathtt{S}_i)$ but make the sum node that evaluates the weighted sum over the SPNs $\mathsf{SPN}(\mathtt{O}, \mathtt{S}_i), 1 \leq i \leq n$, complete.

2. A **product node** that evaluates the product of the attributes' weight functions. For each attribute declared for class C with weight function $\mathbf{u}$ and domain $\mathbf{D}$, we add one sum node representing the efficiently summable/integrable weight function having the (implicitly given) indicator variables $\mathtt{A}(\mathtt{O}, \mathtt{D})$ and $\neg\mathtt{A}(\mathtt{O}, \mathtt{D})$ for every $\mathtt{D} \in \mathbf{D}$ as children. If the attribute A is also declared for a class $\mathtt{C}'$ that is a subclass of C in $\mathtt{Tree}_{(\mathtt{O},\mathtt{C}_0)}$, we do not add the corresponding sum node. Instead, we alter the weight function of the corresponding sum node in $\mathsf{SPN}(\mathtt{O}, \mathtt{C}')$ for every $\mathtt{C}' \in \mathtt{Tree}_{(\mathtt{O},\mathtt{C}_0)}$ with height 0. Intuitively, we merge the sum nodes representing attribute weight functions at different levels of the class tree and bring them to the lowest level of the class tree to ensure decomposability. Let the weight function for attribute A declared for class C have domain $\mathbf{D}$ and weight function $\mathbf{u}$. If there exists a sum node representing the weight function $\mathbf{u}'$ for attribute A in $\mathsf{SPN}(\mathtt{O}, \mathtt{C}')$, then we replace this weight function with the element-wise multiplication of the two weight function $\mathbf{u}' \odot \mathbf{u}$. If there does *not* exist a sum node in $\mathsf{SPN}(\mathtt{O}, \mathtt{C}')$ representing the weight function for attribute A we add the sum node with weight function $\mathbf{u}$ to $\mathsf{SPN}(\mathtt{O}, \mathtt{C}')$'s product node representing the product of attributes. These steps enforce the decomposability of the top product node of $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ and leave all other altered product nodes of the subSPNs decomposable.

3. A **product node** that evaluates the product of the relations' weights. For each relation declared for class C, we add one sum node representing the relation distribution with indicator variables $\mathtt{R}(\mathtt{O})$ and $\neg\mathtt{R}(\mathtt{O})$. Analogous to attributes that are declared for subclasses, we handle relations that are declared in subclasses of C. This renders the product node decomposable.

4. An **indicator variable** $\mathtt{Is}(\mathtt{O}, \mathtt{C})$.

Figure 3 illustrates the subSPN for a generic class-instance pair $(\mathtt{O}, \mathtt{C})$. Since objects that are leaves in the part decomposition do not have parts, the product node evaluating products of parts is absent in the above construction. It is straightforward to verify that the resulting SPNs are valid, since all children are valid SPNs based on the induction hypothesis. Moreover, the SPNs compute $\phi(\mathtt{O}, \mathtt{C}, \mathbf{W})$ for every possible subworld $\mathbf{W}$ with top object O and top class C. Since every possible world determines the truth value of all indicator variables, the SPN also computes the partition function $Z_{(\mathtt{O},\mathtt{C})}$ correctly.

We are now in the position to inductively construct a valid and correct SPN for the top object and the top class of the TPKB, proving part (iii). We accomplish this by induction on the height of the part decomposition. The previous step of the proof showed the base case: we can construct, for every object $\mathtt{O} \in \mathbf{L}_0$ and for every class C the object O is declared to be an instance of, a valid SPN $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ that computes the weight of subworlds and the partition function correctly.

Let class $\mathtt{O} \in \mathbf{L}_h$ and let $h > 0$. The inductive hypothesis is that, for all objects $\mathtt{O} \in \mathbf{L}_i$, $i < h$, and all classes it is declared to be an instance of, we can construct a SPN $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ that is valid and computes the weight of subworlds and the partition function correctly. We need to show that we can, for every $\mathtt{O} \in \mathbf{L}_0$ and for every C it is declared to be an instance of, construct an SPN $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ that is valid and computes the weight of subworlds and the partition function correctly. We now construct the SPN exactly as before, except that there is an additional product node evaluating the product of the parts of class C (see Figure 3). By the induction hypothesis, each of the part's subSPNs are valid and compute the weight of possible worlds and the partition function correctly. Since the indicator variables of each of the subSPNs are disjoint the product node evaluating the product of parts is also valid. With the same arguments made before, we can show that $\mathsf{SPN}(\mathtt{O}, \mathtt{C})$ is valid and computes the weight of subworlds and the partition function correctly. Moreover, the size of the SPN is linear in the size of the TPKB. This concludes the proof. □