Inside-Outside Probability Computation for Belief Propagation

Taisuke Sato
Tokyo Institute of Technology
2-12-1 Ôokayama Meguro-ku Tokyo Japan 152-8552

Abstract

In this paper we prove that the well-known correspondence between the forward-backward algorithm for hidden Markov models (HMMs) and belief propagation (BP) applied to HMMs can be generalized to one between BP for junction trees and the generalized inside-outside probability computation for probabilistic logic programs applied to junction trees.

1 Introduction

Bayesian networks (BNs) and probabilistic context free grammars (PCFGs) are two basic probabilistic frameworks in uncertainty modeling (BNs) and in statistical natural language processing respectively. Although they are independently developed, there is a strong indication of their close relationship. For example both include hidden Markov models (HMMs) as a common subclass. Furthermore belief propagation (BP) applied to HMMs coincides with the forward-backward algorithm for HMMs [Smyth *et al.*, 1997] which is a specialization of probability computation used in the Inside-Outside (IO) algorithm for PCFGs [Baker, 1979]. Nonetheless, however, no exact correspondence beyond this one is known to our knowledge.

In this paper[1] we upgrade this correspondence. We prove that the inside-outside probability computation in the IO algorithm, when generalized for probabilistic logic programs and applied to junction trees, yields BP. In particular we prove that collecting evidence (resp. distributing evidence) in BP coincides with the computation of inside probabilities (resp. outside probabilities) in this generalized IO computation.

To prove the computational correspondence between BNs and PCFGs in a unified manner, we need a general language that can describe BNs and PCFGs[2]. We choose PRISM [Sato and Kameya, 2001] as a first-order probabilistic language for this purpose. We also need "*propositionalization*" of BNs [Sato and Kameya, 2001; McAllester *et al.*, 2004; Chavira and Darwiche, 2005]. By propositionalization we mean to represent a discrete random variable X having n values $\{v_1, \ldots, v_n\}$ by a set $\{X_{v_1}, \ldots, X_{v_n}\}$ of mutually exclusive binary random variables such that $X_{v_i} = 1$ (true) iff $X = v_i$ $(1 \leq i \leq n)$. This propositionalization explodes the number of states in a BN. However the benefit often outweighs the explosion because it makes possible to share computation with finer grain size value-dependently at runtime by dynamic programming and rule out 0 probability computation at compile time. It explains, though we omit details, why probability computation in polynomial time cannot be expected of the direct application of BNs to PCFGs [Pynadath and Wellman, 1996] while it is carried out in $O(n^3)$ time where n is the sentence length by PRISM and by case-factor diagrams (CFDs) [McAllester *et al.*, 2004].

In what follows, we first review basic notions in Section 2. We then prove main theorems after a series of lemmas in Section 3. Due to space limitations, the description is sketchy and the reader is assumed to be familiar with logic programming [Doets, 1994] and BP in junction trees [Jensen, 1996; Lauritzen and Spiegelhalter, 1988; Shafer and Shenoy, 1990]. PRISM, a probabilistic logic programming language used in this paper, is detailed in [Sato and Kameya, 2001].

2 Background

2.1 Bayesian networks and junction trees

A Bayesian network BN is a directed acyclic graph defining a joint distribution $P(X_1 = x_1, \ldots, X_N = x_N) = \Pi_{i=1}^{N} P(X_i = x_i \mid \Pi_i = \pi_i)$ such that nodes are random variables $X_1, \ldots, X_N$ and if a node X_i has parent nodes $\Pi_i = X_{s_1}, \ldots, X_{s_k}$ $(k \geq 0)$, a conditional probability table representing a conditional distribution $P(X_i = x_i \mid \Pi_i = \pi_i)$ is associated with X_i $(1 \leq i \leq N)$. Here a lower case letter x_i denotes a value of X_i and similarly for π_i. We denote the range of X_i by $R(X_i)$ and the direct product $R(X_{s_1}) \times \cdots \times R(X_{s_k})$ by $R(\Pi_i)$ and write $x_i \in R(X_i)$ and $\pi_i \in R(\Pi_i)$. Hereafter we use $P(x_1, \ldots, x_N)$ for $P(X_1 = x_1, \ldots, X_N = x_N)$ etc. When we consider $P(x_i \mid \pi_i)$ as a function of x_i, π_i, the set $\{x_i\} \cup \pi_i$ is called the *arguments* of $P(x_i \mid \pi_i)$. Let $\alpha = \{s_1, \ldots, s_k\}$ be a set of variable indices $(\subseteq \{1, \ldots, N\})$. X_α stands for the set of variables $\{X_{s_1}, \ldots, X_{s_k}\}$. For example if $\alpha = \{1, 2, 3\}$,

[1] We assume in this paper that BNs are discrete and BP is without normalization.

[2] Note that BNs are a propositional framework that deal with finitely many random variables while PCFGs allow recursion and have to deal with infinitely many random variables describing probabilistic choices in a sentence derivation.

$X_\alpha = \{X_1, X_2, X_3\}$. This notation extends to vectors.

A junction (join) tree $T = (V, E)$ for BN is a tree satisfying the following conditions. First a node is a set X_α of variables in BN. In what follows we use X_α and its index set α interchangeably. An edge connecting α and β is labeled by $\alpha \cap \beta$. Second $P(x_1, \ldots, x_N) = \Pi_{\alpha \in V} \phi_\alpha(x_\alpha)$ must hold where $\phi_\alpha(x_\alpha)$, a *potential (function)*, is a product of some (or no) conditional distributions such that their arguments are included in x_α. The third condition is the *running intersection property* (RIP) which dictates that if nodes α and β have a common variable, it must be contained in every node on the path between α and β. RIP ensures the conditional independence of the subtrees given variables in the node and is the key property for efficient probability computation by BP. Given a BN, a junction tree is constructed by way of triangulation or variable elimination [Jensen, 1996; Kask *et al.*, 2001; Lauritzen and Spiegelhalter, 1988].

2.2 PCFGs and inside-outside probabilities

A PCFG is a CFG with probabilities assigned to production rules in such way that if a nonterminal A has N rules, $A \to \alpha_1, \ldots, A \to \alpha_N$, a probability p_i is assigned to $A \to \alpha_i$ for i ($1 \leq i \leq N$) and $\sum_{i=1}^{N} p_i = 1$ holds. p_i is the probability of choosing $A \to \alpha_i$ to expand A in a probabilistic derivation. PCFGs are a basic tool for statistical natural language processing and include HMMs as a subclass.

Let $A(i, j)$ denote an event that a nonterminal A probabilistically derives the substring $w_i, \ldots, w_j$ of a sentence $L = w_1, \ldots, w_n$ ($1 \leq i \leq j \leq n$). The probability of $A(i, j)$ is called the *inside probability* of $A(i, j)$ and defined as the sum of the products of probabilities associated with rules in a derivation belonging to $A(i, j)$. Similarly the *outside probability* of $A(i, j)$ w.r.t. L is the sum of products of the probabilities associated rules used in a derivation that starts from the start symbol and derives $w_1, \ldots, w_{i-1} A w_{j+1}, \ldots, w_n$. The product of inside-outside probabilities of $A(i, j)$ gives the probability of deriving L via $A(i, j)$. Inside-outside probabilities are computed by dynamic programming in time $O(|L|^3)$. We next generalize inside-outside probabilities in the context of probabilistic logic programming.

2.3 Probabilistic logic programming language PRISM

We briefly explain a probabilistic logic programming language PRISM. In a nutshell, PRISM is Prolog extended with *tabling*[3], a probabilistic built-in predicate called `msw` (multi-ary random switch) and a generic parameter learning routine that learn parameters embedded in a program by computing generalized inside-outside probabilities [Sato and Kameya, 2001].

One of the basic ideas of PRISM is propositionalization of random variables using a special built-in predicate `msw`/3. Let V be a discrete random variable with a set $R(V)$ of ground terms as its range. To represent a proposition $V = v$ ($v \in R(V)$), we introduce a ground term i as a name (identifier) for V and a ground msw atom $\texttt{msw}(i, n, v)$ which is true iff the outcome of an n-th trial of V named i is v. Here n is a natural number. V as iids are represented by the set $\{\texttt{msw}(i, n, v) \mid v \in R(V), n = 0, 1, \ldots\}$ of `msw` atoms. These `msw` atoms must satisfy certain conditions[4]. The probability of $\texttt{msw}(i, n, v)$ being true is called a *parameter*.

In PRISM a program $DB = R \cup F$ consists of a set R of definite clauses whose head is not an `msw` atom and a set F of `msw` atoms together with a *base distribution* P_F defining probabilities (parameters) of `msw` atoms in F. Simple distributions are definable soley in terms of `msw` atoms but complex distributions are constructed by using definite clauses. In our semantics, DB defines a probability measure $P_{DB}(\cdot)$ over the set of Herbrand interpretations (*distribution semantics* [Sato and Kameya, 2001]). Hereafter we consider $P_{DB}(\cdot)$ as a distribution on ground atoms as well.

2.4 Propositionalization through tabled search

In our approach, $P_{DB}(G)$, the probability of an atom G defined by a program DB, is computed in two steps. The first step is propositionalization. We apply the SLD refutation procedure [Doets, 1994] to DB and $\Leftarrow G$ as a top-goal, we search for all SLD proofs of G and reduce G to a logically equivalent but propositional DNF formula $E_1 \vee \cdots \vee E_k$ where E_i ($1 \leq i \leq k$), an *explanation* of G, is a conjunction of ground `msw` atoms. They record probabilistic choices made in the process of constructing an SLD proof of G and each `msw` atom represents a probabilistic event $V = v$ for some random variable V. Then in the second step, we compute the probability of G as $P_{DB}(G) = P_{DB}(E_1 \vee \cdots \vee E_k)$.

In general since there are exponentially many proofs and so are explanations, the naive proof search would produce an exponential size DNF. Fortunately however by introducing *tabling* in the first step, we can often produce an equivalent but polynomial size boolean formula such that common subexpressions in the E_i's are factored out as *tabled atom*s. The factored formula, $\mathrm{Expl}(G)$, becomes a set (conjunction) of definitions of the form $H \Leftrightarrow W$ where a tabled atom H is defined by W which is a conjunction of tabled atoms and `msw` atoms. We assume the defining relation of these tabled atoms is acyclic. For convenience we sometimes think of each definition as an AND-OR graph and conventionally call $\mathrm{Expl}(G)$ an *explanation graph* as the collection of such AND-OR graphs. Hereafter $\mathrm{Expl}(G)$ stands for the factored formulas and their graphical representation as well.

2.5 Generalized inside-outside probabilities

To compute $P_{DB}(G)$, we convert each definition $H \Leftrightarrow A_1 \vee \ldots \vee A_L$ in $\mathrm{Expl}(G)$ where $A_i = B_1 \wedge \cdots \wedge B_{M_i} \wedge \texttt{msw}_1 \wedge \cdots \wedge$

[3] Tabling here means to memoize goals whose predicate symbol is declared as *table predicate* and to cache successful goals in a table for later reuse [Zhou and Sato, 2003]. An atom containing a table predicate is called a *tabled atom*.

[4] We require that $P_F(\texttt{msw}(i, n, v) \wedge \texttt{msw}(i, n, v')) = 0$ for $v \neq v' \in R(V)$ and $P_F(\bigvee_{v \in R(V)} \texttt{msw}(i, n, v)) = \sum_{v \in R(V)} P_F(\texttt{msw}(i, n, v)) = 1$ holds for any n. Also when $i \neq i'$ or $n \neq n'$, $\texttt{msw}(i, n, v)$ and $\texttt{msw}(i', n', v)$ must be independent and $\texttt{msw}(i, n, v)$ and $\texttt{msw}(i, n', v)$ must be identically distributed.

$\mathtt{msw}_{N_i}$ $(1 \leq i \leq L)$ to a numerical sum-product equation.

$$\begin{array}{rcl} P_{DB}(H) & = & P_{DB}(A_1) + \cdots + P_{DB}(A_L) \\ P_{DB}(A_i) & = & P_{DB}(B_1) \cdots P_{DB}(B_{M_i}) \cdot \\ & & \quad P_{DB}(\mathtt{msw}_1) \cdots P_{DB}(\mathtt{msw}_{N_i}) \end{array} \quad (1)$$

Note that this conversion assumes the *mutual exclusiveness* of disjuncts $\{A_1, \ldots, A_L\}$ and the *independence* of conjuncts $\{B_1, \ldots, B_{M_i}, \mathtt{msw}_1, \ldots, \mathtt{msw}_{N_i}\}$. Although guaranteeing these two conditions is basically the user's responsibility, they are automatically satisfied as far as the PRISM program describing junction trees is concerned (see Section 3 and Lemma 3.3). We denote by $\mathrm{Eq}(G)$ the set of converted equations.

For a ground atom A, we call $P_{DB}(A)$ a *P-variable*. P-variables are just numerical variables named by ground atoms. As we assume the defining relation of tabled atoms is acyclic, P-variables in $\mathrm{Eq}(G)$ can be linearly ordered so that $\mathrm{Eq}(G)$ is efficiently solved in a bottom-up manner by dynamic programming in time proportional to the size of $\mathrm{Eq}(G)$. Also the acyclicity implies that a higher P-variable is a multivariate polynomial in the lower P-variables, and hence we can take the derivative of a higher P-variable as a function of the lower P-variables.

Suppose we are given a program DB. In an analogy to inside-outside probabilities in PCFGs, we define a *generalized inside probability* $\mathrm{inside}(A)$ of a ground atom A by $\mathrm{inside}(A) \stackrel{\mathrm{def}}{=} P_{DB}(A)$ and extend the definition to a conjunction W of ground atoms by $\mathrm{inside}(W) \stackrel{\mathrm{def}}{=} P_{DB}(W)$.

We also define a *generalized outside probability* $\mathrm{outside}(G\,;A)$ of A w.r.t. a top-goal G as follows. First enumerate A's occurrences in $\mathrm{Expl}(G)$ as

$$\begin{cases} H_1 \Leftrightarrow (A \wedge W_{1,1}) \vee \cdots \vee (A \wedge W_{1,i_1}) \\ \quad \cdots \\ H_J \Leftrightarrow (A \wedge W_{J,1}) \vee \cdots \vee (A \wedge W_{J,i_J}). \end{cases}$$

Then $\mathrm{outside}(G\,;A)$ is recursively computed by Eq. 2[5].

$$\begin{array}{l} \mathrm{outside}(G\,;G) \;=\; 1 \\ \mathrm{outside}(G\,;A) \;= \mathrm{outside}(G\,;H_1) \prod_{j=1}^{i_1} \mathrm{inside}(W_{1,j}) \\ \qquad + \cdots + \mathrm{outside}(G\,;H_J) \prod_{j=1}^{i_J} \mathrm{inside}(W_{J,j}). \end{array} \quad (2)$$

Using Eq. 2, all outside probabilities are computed in time in the size of $\mathrm{Eq}(G)$ [Sato and Kameya, 2001]. We can prove that for a ground atom A, the product $\mathrm{inside}(A) \cdot \mathrm{outside}(A)$ is the average number of occurrences of A in a proof of G and that our definition is a generalization of the usual definition of inside-outside probabilities in PCFGs [Lafferty, 1993].

[5] As mentioned above, $P_{DB}(G)$, a P-variable, is a function of other P-variables $P_{DB}(A)$ and the mathematical definition of $\mathrm{outside}(G\,;A)$ is

$$\mathrm{outside}(G\,;A) \stackrel{\mathrm{def}}{=} \frac{\partial\, P_{DB}(G)}{\partial\, P_{DB}(A)}$$

which derives Eq. 2.

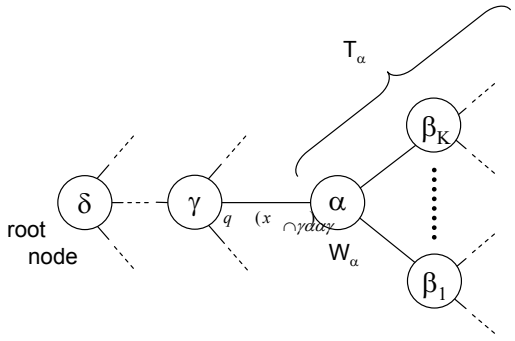

Figure 1: Junction tree $T = \langle V, E \rangle$

3 Belief propagation as the generalized IO computation

In this section, we prove that the generalized IO computation, i.e. the computation of generalized inside-outside probabilities, subsumes BP in junction trees.

3.1 Program for BP messages

Suppose we have a BN defining a joint distribution $P(X_1 = x_1, \ldots, X_N = x_N) = \Pi_{i=1}^{N} P(x_i = X_i \mid \Pi_i = \pi_i)$ and a junction tree $T = (V, E)$ such that $P(x_1, \ldots, x_N) = \Pi_{\alpha \in V} \phi_\alpha(x_\alpha)$. Let δ be the root node of T[6].

We construct a PRISM program $DB_T = F_T \cup R_T$ that describes BP in T as follows[7]. Introduce for each conditional probability $P(x_i \mid \pi_i)$ a ground msw atom $\mathtt{msw}(\mathtt{bn}(i, \pi_i), \mathtt{once}, x_i)$. If X_i has no parent put $\pi_i = []$ (null list). Define a finite set F_T of msw atoms by

$$F_T \stackrel{\mathrm{def}}{=} \{\, \mathtt{msw}(\mathtt{bn}(i, \pi_i), \mathtt{once}, x_i) \mid 1 \leq i \leq N, x_i \in R(X_i), \pi_i \in R(\Pi_i) \,\}$$

and set parameters $\theta_{\mathtt{bn}(i,\pi_i),x_i}$ by

$$\theta_{\mathtt{bn}(i,\pi_i),x_i} = P_F(\mathtt{msw}(\mathtt{bn}(i, \pi_i), \mathtt{once}, x_i)) = P(x_i \mid \pi_i).$$

Then it is easy to see that every joint probability is represented by a conjunction of these ground msw atoms, i.e. we have

$$P(x_1, \ldots, x_N) = P_F\left(\textstyle\bigwedge_{i=1}^{N} \mathtt{msw}(\mathtt{bn}(i, \pi_i), \mathtt{once}, x_i)\right).$$

Next introduce an atom $\mathtt{msw}(\mathtt{bn}(i, \Pi_i), \mathtt{once}, X_i)$ containing variables X_i and Π_i for each i $(1 \leq i \leq N)$. $\mathtt{msw}(\mathtt{bn}(i, \Pi_i), \mathtt{once}, X_i)$ represents the conditional distribution $P(X_i = x_i \mid \Pi_i = \pi_i)$[8]. For every node α in the junction tree T, define a conjunction $W_\alpha(X_\alpha)$ representing the potential $\phi_\alpha(x_\alpha)$ of α and introduce a clause C_α defining a *message atom* $q_{\alpha\gamma}(X_{\alpha \cap \gamma})$ that describes a message in BP sent

[6] In what follows, for simplicity we assume no evidence is given. When some variables are observed however, all conclusions remain valid, except that they are fixed to the observed values.

[7] Programming convention follows Prolog.

[8] Here we use intentionally X_i both as a logical variable and as a random variable to make explicit the correspondence between general msw atoms and conditional distributions in BN.

from α to its parent node γ. They are respectively defined as

$$W_\alpha(X_\alpha) \stackrel{\text{def}}{=} \bigwedge_{P(x_i|\pi_i)\in\phi_\alpha} \mathtt{msw}(\mathtt{bn}(i,\Pi_i),\mathtt{once},X_i),$$
$$C_\alpha \stackrel{\text{def}}{=} q_{\alpha\gamma}(X_{\alpha\cap\gamma}) \Leftarrow W_\alpha(X_\alpha)\wedge q_{\beta_1\alpha}(X_{\beta_1\cap\alpha})\wedge\cdots\wedge q_{\beta_K\alpha}(X_{\beta_K\cap\alpha}).$$

Here $\beta_1,\ldots,\beta_K$ $(K \geq 0)$ are the child nodes of α in T. The next lemma states that $W_\alpha(X_\alpha)$ correctly describes the potential of node α. The proof is straightforward and omitted.

Lemma 3.1

$$P_F(W_\alpha(x_\alpha)) \;=\; \prod_{i:P(x_i|\pi_i)\in\phi_\alpha} P(x_i \mid \pi_i) \;=\; \phi_\alpha(x_\alpha).$$

□

For the root node δ in T, it has no parent but we add a special parent node 0 to V and define C_δ as

$$C_\delta \stackrel{\text{def}}{=} q_{\delta 0} \Leftarrow W_\delta(X_\delta)\wedge q_{\beta'_1\delta}(X_{\beta'_1\cap\delta})\wedge\cdots\wedge q_{\beta'_{K'}\delta}(X_{\beta'_{K'}\cap\delta})$$

where $\beta'_1,\ldots,\beta'_{K'}$ are the child nodes of δ. $q_{\delta 0}$ has no arguments but calls every message atom directly or indirectly. Finally put

$$R_T \stackrel{\text{def}}{=} \{C_\alpha \mid \alpha \in V, T=(V,E)\}.$$

We illustrate a small example. Take a discrete Bayesian network BN_1 on $\{X_1,\ldots,X_5\}$ on the left-hand side of Figure 2 and its junction tree T_1 on the right-hand side with the root node γ_1. Dotted lines in BN_1 indicate edges added by triangulation. Figure 3 shows the definitions of message atoms for T_1.

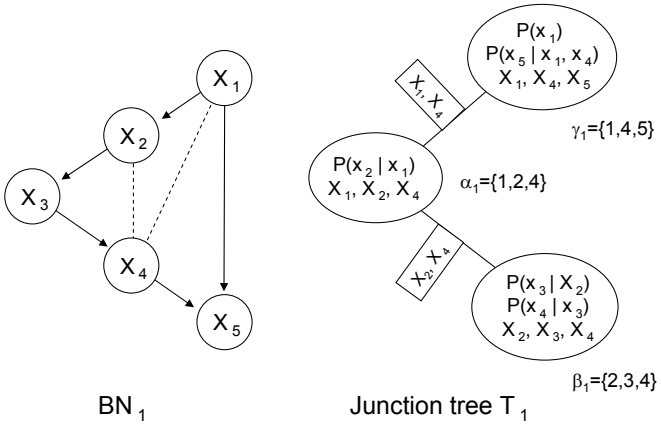

Figure 2: BN_1 and a junction tree T_1

3.2 Explanation graphs for BP messages

Let $DB_T = F_T \cup R_T$ be the program constructed in Subsection 3.1. After declaring every $q_{\alpha\gamma}$ predicate as a table predicate, we apply tabled search for all proofs to a top-goal $\Leftarrow q_{\delta 0}$ where δ is the root node of T. The search always terminates and yields an explanation graph $\mathrm{Expl}(q_{\delta 0})$ which contains a

$$R_{T_1}\left\{\begin{array}{l} q_{\gamma_1 0} \Leftarrow \mathtt{msw}(\mathtt{bn}(1,\mathtt{once},[]),X_1)\wedge \\ \qquad \mathtt{msw}(\mathtt{bn}(5,\mathtt{once},[X_4,X_1]),X_5)\wedge \\ \qquad q_{\alpha_1\gamma_1}(X_1,X_4) \\ q_{\alpha_1\gamma_1}(X_1,X_4) \Leftarrow \\ \quad \mathtt{msw}(\mathtt{bn}(2,\mathtt{once},[X_1]),X_2)\ \wedge q_{\beta_1\alpha_1}(X_2,X_4) \\ q_{\beta_1\alpha_1}(X_2,X_4) \Leftarrow \\ \quad \mathtt{msw}(\mathtt{bn}(3,\mathtt{once},[X_2]),X_3)\wedge \\ \quad \mathtt{msw}(\mathtt{bn}(4,\mathtt{once},[X_3]),X_4) \end{array}\right.$$

Figure 3: The definitions of message atoms for T_1.

definition of tabled atom $q_{\alpha\gamma}(x_{\alpha\cap\gamma})$ for every node α in T as shown below.

$$q_{\alpha\gamma}(x_{\alpha\cap\gamma}) \;\Leftrightarrow\; \bigvee_{x_{\alpha\setminus\gamma}} \left(W_\alpha(x_\alpha)\wedge\textstyle\bigwedge_{i=1}^K q_{\beta_i\alpha}(x_{\beta_i\cap\alpha})\right). \tag{3}$$

Here $x_{\alpha\cap\gamma}$ denotes an arbitrary ground instantiation of $X_{\alpha\cap\gamma}$ and $q_{\alpha\gamma}(x_{\alpha\cap\gamma})$ represents the message sent from α to γ, α's parent node, after receiving messages from α's child nodes $\beta_1,\ldots,\beta_K$ (see Figure 1).

We next prove that the recursive equations Eq. 3 are "solved" uniquely. Let T_α be the subtree of T rooted at X_α and X_{ξ_α} be the set of variables appearing in T_α. We first introduce a formula $\tau_\alpha(X_{\xi_\alpha})$ for α by Eq. 4 and rewrite it to Eq. 5. It represents the potential of the subtree T_α.

$$\tau_\alpha(X_{\xi_\alpha}) \stackrel{\text{def}}{=} \textstyle\bigwedge_{\rho\in T_\alpha} W_\rho(X_\rho) \tag{4}$$
$$= W_\alpha(X_\alpha)\wedge\textstyle\bigwedge_{i=1}^K \tau_{\beta_i}(X_{\xi_{\beta_i}}) \tag{5}$$

Lemma 3.2

$$\begin{aligned} \xi_\alpha &= \alpha\cup\textstyle\bigcup_{i=1}^K \xi_{\beta_i} \\ \xi_{\beta_i}\setminus\gamma &= (\xi_{\beta_i}\setminus\alpha)\cup(\beta_i\cap(\alpha\setminus\gamma)) \ \text{(from RIP of } T) \\ \xi_\alpha\setminus\gamma &= (\alpha\setminus\gamma)\cup\textstyle\bigcup_{i=1}^K(\xi_{\beta_i}\setminus\alpha) \end{aligned} \tag{6}$$

□

Proposition 3.1 $q_{\alpha\gamma}(x_{\alpha\cap\gamma}) = \bigvee_{x_{\xi_\alpha\setminus\gamma}} \tau_\alpha(x_{\xi_\alpha})$ □

(Proof) By well-founded induction on T. When α is a leaf in T, the proposition is obviously true. Assume it is true w.r.t. the child nodes of α.

$$\begin{aligned} &\bigvee_{x_{\xi_\alpha\setminus\gamma}} \tau_\alpha(x_{\xi_\alpha}) \\ &= \bigvee_{x_{\alpha\setminus\gamma}}\bigvee_{x_{\beta_1\setminus\alpha}}\cdots\bigvee_{x_{\beta_K\setminus\alpha}} \tau_\alpha(x_{\xi_\alpha}) \ \text{by Eq. 6} \\ &\quad ((\alpha\setminus\gamma) \text{ and } (\beta_i\setminus\alpha)\text{'s are mutually disjoint}) \\ &= \bigvee_{x_{\alpha\setminus\gamma}}\left(W_\alpha(x_\alpha)\wedge\bigwedge_{i=1}^K\left(\bigvee_{x_{\xi_{\beta_i}\setminus\alpha}}\tau_{\beta_i}(x_{\xi_{\beta_i}})\right)\right) \ \text{by Eq. 5} \\ &= \bigvee_{x_{\alpha\setminus\gamma}}\left(W_\alpha(x_\alpha)\wedge\bigwedge_{i=1}^K q_{\beta_i\alpha}(x_{\beta_i\cap\alpha})\right) \ \text{by induction} \\ &= q_{\alpha\gamma}(x_{\alpha\cap\gamma}) \ \text{by Eq. 3} \qquad \text{Q.E.D.} \end{aligned}$$

3.3 BP and the generalized IO computation

Let P_{DB_T} be the distribution defined by DB_T. The generalized inside probability of the tabled atom $\mathrm{inside}(q_{\alpha\gamma}(x_{\alpha\cap\gamma}))$ and the generalized outside probability $\mathrm{outside}(q_{\delta 0}\,; q_{\alpha\gamma}(x_{\alpha\cap\gamma}))$ w.r.t. $q_{\delta 0}$ can be computed using Eq. 3 by sum-product computation specified in Eq. 1 *if* the independence of conjuncts and the mutual exclusiveness of disjuncts on the right-hand side of Eq. 3 are guaranteed.

Since each `msw` atom $\mathtt{msw}(\mathtt{bn}(i,\Pi_i),\mathtt{once},X_i)$ occurs only once in some W_α reflecting the fact that a conditional distribution function $P(x_i \mid \pi_i)$ in the BN belongs exclusively to one potential ϕ_α in the junction tree, $W_\alpha(x_\alpha)$ and the $q_{\beta_i\alpha}(x_{\beta_i\cap\alpha})$'s do not share any `msw` atoms. Hence the first condition, the independence condition, is satisfied automatically. On the other hand, proving the mutual exclusiveness condition is not straightforward. Lemma 3.3 below assures the exclusiveness condition when combined with Proposition 3.1.

Note that $\xi_\alpha \cap \gamma = \alpha \cap \gamma$ holds thanks to RIP of T. So we rewrite $\tau_\alpha(X_{\xi_\alpha})$ as

$$\begin{aligned}\tau_\alpha(X_{\xi_\alpha}) &= \tau_\alpha(X_{\xi_\alpha\setminus\gamma}, X_{\xi_\alpha\cap\gamma})\\ &= \tau_\alpha(X_{\xi_\alpha\setminus\alpha}, X_{\alpha\cap\gamma}).\end{aligned}$$

Lemma 3.3 *Let $x_{\xi_\alpha\setminus\gamma}$ and $x'_{\xi_\alpha\setminus\gamma}$ be two different ground instantiations of $X_{\xi_\alpha\setminus\gamma}$. Then for an arbitrary ground instantiation $x_{\alpha\cap\gamma}$ of $X_{\alpha\cap\gamma}$, we have $\neg\left(\tau_\alpha(x_{\xi_\alpha\setminus\gamma}, x_{\alpha\cap\gamma}) \wedge \tau_\alpha(x'_{\xi_\alpha\setminus\gamma}, x_{\alpha\cap\gamma})\right)$.* □

(Sketch of proof) Without loss of generality, we can write $X_{\xi_\alpha\setminus\gamma} = X_{i_1},\ldots,X_{i_M}$ in such a way that if X_{i_j} is a parent node of X_{i_k} in the original BN, X_{i_j} precedes X_{i_k} in this list. As $x_{\xi_\alpha\setminus\gamma} = (x_{i_1},\ldots,x_{i_M}) \neq x'_{\xi_\alpha\setminus\gamma} = (x'_{i_1},\ldots,x'_{i_M})$, there is a variable X_{i_s} such that $x_{i_1} = x'_{i_1},\ldots,x_{i_{s-1}} = x'_{i_{s-1}}, x_{i_s} \neq x'_{i_s}$ $(1 \leq s \leq M)$. Then first we note $\Pi_{i_s} \subseteq X_{\xi_\alpha}$ holds since X_{i_s} appears only in T_α and the conditional distribution $P(X_{i_s} \mid \Pi_{i_s})$ must be contained in some potential in T_α by RIP. Second we have $\Pi_{i_s} \cap X_{\xi_\alpha\setminus\gamma} \subseteq \{X_{i_1},\ldots,X_{i_{s-1}}\}$. We also have $\Pi_{i_s} \cap X_{\xi_\alpha\cap\gamma} = \Pi_{i_s} \cap X_{\alpha\cap\gamma}$. We can conclude from these facts that $(x_{\xi_\alpha\setminus\gamma}, x_{\alpha\cap\gamma})$ instantiates $P(X_{i_s} \mid \Pi_{i_s})$ to $P(x_{i_s} \mid \pi_{i_s})$ while $(x'_{\xi_\alpha\setminus\gamma}, x_{\alpha\cap\gamma})$ instantiates $P(X_{i_s} \mid \Pi_{i_s})$ to $P(x'_{i_s} \mid \pi_{i_s})$ where $x_{i_s} \neq x'_{i_s}$. The rest is immediate and omitted. Q.E.D.

We now prove main theorems by applying computation in Eq. 1 to the tabled atoms defined by Eq. 3. Recall that $\mathrm{inside}(A) = P_{DB_T}(A)$ for a ground atom A where P_{DB_T} is the distribution defined by DB_T. We derive an equation satisfied by inside probabilities of tabled atoms.

Theorem 3.1

$$\mathrm{inside}(q_{\alpha\gamma}(x_{\alpha\cap\gamma})) = \sum_{x_{\alpha\setminus\gamma}} \phi_\alpha(x_\alpha) \prod_{i=1}^{K} \mathrm{inside}(q_{\beta_i\alpha}(x_{\beta_i\cap\alpha})).$$

□

(Proof)

$$\begin{aligned}&\mathrm{inside}(q_{\alpha\gamma}(x_{\alpha\cap\gamma}))\\ &= P_{DB_T}(q_{\alpha\gamma}(x_{\alpha\cap\gamma}))\\ &= P_{DB_T}\left(\bigvee\nolimits_{x_{\xi_\alpha\setminus\gamma}} \tau_\alpha(x_{\xi_\alpha})\right) \text{ by Proposition 3.1}\\ &= \textstyle\sum_{x_{\xi_\alpha\setminus\gamma}} P_{DB_T}(\tau_\alpha(x_{\xi_\alpha})) \text{ by Lemma 3.3}\\ &= \textstyle\sum_{x_{\alpha\setminus\gamma}} \sum_{x_{\beta_1\setminus\alpha}} \cdots \sum_{x_{\beta_K\setminus\alpha}} P_{DB_T}(W_\alpha(x_\alpha)) \cdot \\ &\qquad \textstyle\prod_{i=1}^{K} P_{DB_T}(\tau_{\beta_i}(x_{\xi_{\beta_i}})) \quad \text{by Eq. 6}\\ &= \sum_{x_{\alpha\setminus\gamma}} P_{DB_T}(W_\alpha(x_\alpha)) \prod_{i=1}^{K} P_{DB_T}\left(\bigvee_{x_{\beta_i\setminus\alpha}} \tau_{\beta_i}(x_{\xi_{\beta_i}})\right)\\ &= \sum_{x_{\alpha\setminus\gamma}} \phi(x_\alpha) \prod_{i=1}^{K} P_{DB_T}(q_{\beta_i\alpha}(x_{\xi_{\beta_i}\cap\alpha})) \text{ by Lemma 3.1}\\ &= \sum_{x_{\alpha\setminus\gamma}} \phi(x_\alpha) \prod_{i=1}^{K} \mathrm{inside}(q_{\beta_i\alpha}(x_{\xi_{\beta_i}\cap\alpha})) \qquad \text{Q.E.D.}\end{aligned}$$

Theorem 3.1 tells us that the generalized inside probabilities of tabled atoms satisfy exactly the same equations as messages in the collecting evidence phase of BP in T with the root node δ [Jensen, 1996; Lauritzen and Spiegelhalter, 1988; Shafer and Shenoy, 1990]. Hence, the bottom-up computation of generalized inside probabilities is identical to BP in the collecting evidence phase.

Let P_1 be the distribution defined by BN_1 in Figure 2. The equations for generalized inside probabilities of tabled atoms for the junction tree T_1 are:

$$\begin{aligned}&\mathrm{inside}(q_{\alpha_1\gamma_1}(x_1,x_4))\\ &\quad= \textstyle\sum_{x_2} P_{DB_{T_1}}(\mathtt{msw}(\mathtt{bn}(2,[x_1]),x_2)) \cdot\\ &\qquad\qquad \mathrm{inside}(q_{\beta_1\alpha_1}(x_2,x_4))\\ &\quad= \textstyle\sum_{x_2} P_1(x_2 \mid x_1)\mathrm{inside}(q_{\beta_1\alpha_1}(x_2,x_4))\\ &\mathrm{inside}(q_{\beta_1\alpha_1}(x_2,x_4))\\ &\quad= \textstyle\sum_{x_3} P_{DB_{T_1}}(\mathtt{msw}(\mathtt{bn}(3,[x_2]),x_3)) \cdot\\ &\qquad\qquad P_{DB_{T_1}}(\mathtt{msw}(\mathtt{bn}(4,[x_3]),x_4))\\ &\quad= \textstyle\sum_{x_3} P_1(x_3 \mid x_2) \cdot P_1(x_4 \mid x_3).\end{aligned}$$

We next compute generalized outside probabilities of tabled atoms. Without loss of generality, we compute the outside probability of a tabled atom for β_1. We apply the definition of generalized outside probability in Eq. 2 to $\mathrm{Expl}(q_{\delta 0})$ while noting that a tabled atom $q_{\beta_1\alpha}(x_{\beta_1\cap\alpha})$ occurs in $\mathrm{Expl}(q_{\delta 0})$ as in Eq. 3. We obtain recursive equations about generalized outside probabilities as follows.

Theorem 3.2

$$\begin{aligned}\mathrm{outside}(q_{\beta_1\alpha}(x_{\beta_1\cap\alpha})) &= \textstyle\sum_{x_{\alpha\setminus\beta_1}} \phi_\alpha(x_\alpha)\\ &\quad \mathrm{outside}(q_{\alpha\gamma}(x_{\alpha\cap\gamma})) \textstyle\prod_{i=2}^{K} \mathrm{inside}(q_{\beta_i\alpha}(x_{\beta_i\cap\alpha})).\end{aligned}$$

□

(Proof)

$$\begin{aligned}
&\mathrm{outside}(q_{\beta_1\alpha}(x_{\beta_1\cap\alpha}))\\
&= \sum_{x_{(\alpha\cap\gamma)\setminus(\beta_1\cap\alpha)}} \mathrm{outside}(q_{\alpha\gamma}(x_{\alpha\cap\gamma})) \sum_{x_{(\alpha\setminus\gamma)\setminus(\beta_1\cap\alpha)}} \phi_\alpha(x_\alpha)\\
&\quad \textstyle\prod_{i=2}^{K}\mathrm{inside}(q_{\beta_i\alpha}(x_{\beta_i\cap\alpha})) \quad \text{by Eq. 2}\\
&= \sum_{x_\Delta}\mathrm{outside}(q_{\alpha\gamma}(x_{\alpha\cap\gamma}))\phi_\alpha(x_\alpha)\prod_{i=2}^{K}\mathrm{inside}(q_{\beta_i\alpha}(x_{\beta_i\cap\alpha}))\\
&\quad \text{where } \Delta = ((\alpha\cap\gamma)\setminus(\alpha\cap\beta_1))\cup((\alpha\setminus\gamma)\setminus(\beta_1\cap\alpha))\\
&\qquad\qquad = \alpha\setminus\beta_1\\
&= \sum_{x_{\alpha\setminus\beta_1}} \phi_\alpha(x_\alpha)\mathrm{outside}(q_{\alpha\gamma}(x_{\alpha\cap\gamma}))\prod_{i=2}^{K}\mathrm{inside}(q_{\beta_i\alpha}(x_{\beta_i\cap\alpha})).
\end{aligned}$$

Q.E.D.

$\mathrm{outside}(q_{\delta 0}) = 1$ holds for the top-node δ. Therefore we have the following corollary:

Corollary 3.1 *Let $\beta'_1, \ldots, \beta'_{K'}$ be δ's child nodes.*

$$\mathrm{outside}(q_{\beta'_1\delta}(x_{\beta'_1\cap\delta})) = \sum_{x_{\delta\setminus\beta'_1}} \phi_\delta(x_\delta)\prod_{i=2}^{K'}\mathrm{inside}(q_{\beta'_i\delta}(x_{\beta'_i\cap\delta})).$$

□

Theorem 3.2 in conjunction with Corollary 3.1 clearly shows that the computation of generalized outside probabilities of tabled atoms in a top-down manner that starts from the top-node δ and proceeds to lower layers in $\mathrm{Expl}(q_{\delta 0})$ is exactly the same as the distributing evidence phase of BP in T with the root node δ [Jensen, 1996; Lauritzen and Spiegelhalter, 1988; Shafer and Shenoy, 1990]. We illustrate below the computation of the generalized outside probabilities of atoms in R_{T_1} w.r.t. the junction tree T_1 in Figure 2.

$$\begin{aligned}
&\mathrm{outside}(q_{\alpha_1\gamma_1}(x_1,x_4))\\
&\quad = \textstyle\sum_{x_5} P_{DB_{T_1}}(\mathtt{msw}(\mathtt{bn}(1,[]),x_1)\wedge\\
&\qquad\qquad \mathtt{msw}(\mathtt{bn}(5,[x_1,x_4]),x_5))\\
&\quad = \textstyle\sum_{x_5} P_1(x_1)P_1(x_5\mid x_1,x_4)\\
&\mathrm{outside}(q_{\beta_1\alpha_1}(x_2,x_4))\\
&\quad = \textstyle\sum_{x_1}\mathrm{outside}(q_{\alpha_1\gamma_1}(x_1,x_4))\cdot\\
&\qquad\qquad P_{DB_{T_1}}(\mathtt{msw}(\mathtt{bn}(2,[x_1]),x_2))\\
&\quad = \textstyle\sum_{x_1,x_5} P_1(x_1)P_1(x_5\mid x_1,x_4)P_1(x_2\mid x_1).
\end{aligned}$$

Finally we confirm that since $\mathrm{inside}(q_{\delta 0}) = 1$ and every tabled atom occurs only once in the proof of $q_{\delta 0}$, the product of generalized inside-outside probabilities equals a marginal probability as follows.

$$\begin{aligned}
&\mathrm{inside}(q_{\beta_1\alpha_1}(x_2,x_4))\mathrm{outside}(q_{\beta_1\alpha_1}(x_2,x_4))\\
&\quad = \textstyle\sum_{x_3} P_1(x_3\mid x_2)P_1(x_4\mid x_3)\\
&\qquad \textstyle\sum_{x_1,x_5} P_1(x_1)P_1(x_5\mid x_1,x_4)P_1(x_2\mid x_1)\\
&\quad = P_1(x_4\mid x_2)P_1(x_2) \;=\; P_1(x_2,x_4).
\end{aligned}$$

4 Conclusion

We have proved that BP in junction trees is nothing but the generalized IO computation applied to junction trees (Theorem 3.1 and 3.2, Corollary 3.1). This equivalence is a generalization of the well-known equivalence between the forward-backward algorithm and BP applied to HMMs [Smyth *et al.*, 1997] and provides a missing link between BP and PCFGs for the first time.

The most closely related work to ours is CFDs proposed by McAllester et al. [McAllester *et al.*, 2004]. CFDs are a propositional framework for probabilistic inference of Markov random fields. They proved that a single algorithm can efficiently compute probabilities both for PCFGs and for BNs in their framework but the relationship between BP and their algorithm remains unclear. Since PRISM also generates propositional expressions (explanation graphs) from first order expressions by (tabled) search, it is an interesting future topic to relate CFDs to PRISM.

References

[Baker, 1979] J. K. Baker. Trainable grammars for speech recognition. In *Proceedings of Spring Conference of the Acoustical Society of America*, pages 547–550, 1979.

[Chavira and Darwiche, 2005] M. Chavira and A. Darwiche. Compiling bayesian networks with local structure. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, pages 1306–1312, 2005.

[Doets, 1994] K. Doets. *From Logic to Logic Programming*. The MIT Press, 1994.

[Jensen, 1996] F. V. Jensen. *An Introduction to Bayesian Networks*. UCL Press, 1996.

[Kask *et al.*, 2001] K. Kask, R. Dechter, J Larrosa, and F. Cozman. Bucket-tree elimination for automated reasoning. ICS Technical Report Technical Report No.R92, UC Irvine, 2001.

[Lafferty, 1993] J.D. Lafferty. A derivation of the Inside-Outside Algorithm from the EM algorithm. Technical Report TR-IT-0056, IBM T.J.Watson Research Center, 1993.

[Lauritzen and Spiegelhalter, 1988] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their applications to expert systems. *Journal of the Royal Statistical Society, B*, 50:157–224, 1988.

[McAllester *et al.*, 2004] D. McAllester, M. Collins, and F. Pereira. Case-factor diagrams for structured probabilistic modeling. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 382–391, Arlington, Virginia, 2004. AUAI Press.

[Pynadath and Wellman, 1996] D. V. Pynadath and M. P. Wellman. Generalized queries on probabilistic context-free grammars. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI'96)*, pages 1285–1290, 1996.

[Sato and Kameya, 2001] T. Sato and Y. Kameya. Parameter learning of logic programs for symbolic-statistical modeling. *Journal of Artificial Intelligence Research*, 15:391–454, 2001.

[Shafer and Shenoy, 1990] G.R. Shafer and P.P. Shenoy. Probability propagation. *Annals of Mathematics and Artificial Intelligence*, 2:327–352, 1990.

[Smyth *et al.*, 1997] P. Smyth, D. Heckerman, and M. Jordan. Probabilistic independence networks for hidden markov probability models. *Neural Computation*, 9(2):227–269, 1997.

[Zhou and Sato, 2003] Neng-Fa Zhou and T. Sato. Efficient Fixpoint Computation in Linear Tabling. In *Proceedings of the Fifth ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP2003)*, pages 275–283, 2003.