A Reestimation Algorithm for Probabilistic Recursive Transition Network[*]

Young S. Han, and Key-Sun Choi

Center for Artificial Intelligence
Computer Science Department
Center for Artificial Intelligence Research
Korea Advanced Institute of Science and Technology
Taejon, 305-701, Korea
yshan@csking.kaist.ac.kr, kschoi@csking.kaist.ac.kr

Abstract

Probabilistic Recursive Transition Network(PRTN) is an elevated version of RTN to model and process languages in stochastic parameters. The representation is a direct derivation from the RTN and keeps much the spirit of Hidden Markov Model at the same time. We present a reestimation algorithm for PRTN that is a variation of Inside-Outside algorithm that computes the values of the probabilistic parameters from sample sentences (parsed or unparsed).

1. Introduction

In this paper, we introduce a network representation, **Probabilistic Recursive Transition Network** that is directly derived from RTN and HMM, and present an estimation algorithm for the probabilistic parameters. PRTN is a RTN augmented with probabilities in the transitions and states and with the lexical distributions in the transitions, or is the Hidden Markov Model augmented with a stack that makes some transitions deterministic.

The parameter estimation of PRTN is developed as a variation of Inside-Outside algorithm. The Inside-Outside algorithm has been applied to SCFG recently by Jelinek (1990) and Lari (1991). The algorithm was first introduced by Baker in 1979 and is the context free language version of Forward-Backward algorithm in Hidden Markov Models. Its theoretical foundation is laid by Baum and Welch in the late 60's, which in turn is a type of the EM algorithm in statistics (Rabiner, 1989).

Kupiec (1991) introduced a trellis based estimation algorithm of Hidden SCFG that accommodates both Inside-Outside algorithm and Forward-Backward algorithm. The meaning of our work can be sought from the use of more plain topology of RTN, whereas Kupiec's work is a unified version of forward-backword and Inside-Outside algorithms. Nonetheless, the implementation of reestimation algorithm carries no more theoretical significance than the applicative efficiency and variation for differing representations since Baker first applied it to CFGs.

2. Probabilistic Recursive Transition Network

A probabilistic RTN (PRTN, hereafter) denoted by λ is a 4 tuple.

$$\lambda = (\mathcal{A}, \mathcal{B}, \Gamma, \Xi).$$

$\mathcal{A}$ is a transition matrix containing transition probabilities, and $\mathcal{B}$ is an observation matrix containing probability distribution of the words observable at each terminal transition where row and column correspond to terminal transitions and a list of words respectively. Γ specifies the types of transitions, and Ξ denotes a stack. The first two model parameters are the same as that of HMM, thus typed transitions and the existence of a stack are what distinguishes PRTN from HMM.

[*]This research is partly supported by KOSEF (Korea Science and Technology Foundation) under the title "A Study on the Building Techniques for Robust Knowledge-based Systems" from 1991 through 1994.

The stack operations are associated with transitions. According to the stack operation, transitions are classified into three types. The first type is **push transition** in which state identification is pushed into the stack. The second type is **pop transition** which is selected by the content of stack. Transitions of the third type are not committed to stack operation. The three types are also accompanied by different grammatical implication, hence grammatical categories are assigned to transitions except pop transitions. Push transitions are associated with nonterminal categories, and will be called **nonterminal transition** when it is more transparent in later discussions. In general, the grammar expressed in PRTN consists of layers. A **layer** is a fragment of network that corresponds to a nonterminal. The third type of transition is linked to the category of terminals (words), thus is named **terminal transition.** Also a table of probability distribution of words is defined on each terminal transition. In the context of HMMs, the words in the terminal transition are observations to be generated. Pop transitions represent returning of a layer to one of its possibly multiple higher layers.

The network topology of PRTN is not different from that of RTN. In a conceptual drawing of a grammar, each layer looks like an independent network. Compared with conceptual drawing of the network, an operational view provides more vivid representation in which actual paths or parses are composed. The only difference between the two is that in operational view a nonterminal transition is connected directly to the first state of the corresponding layer. In this paper, the parses or paths are assumed to be sequences of dark-headed transitions (see Fig. 1 for example).

Before we start explaining the algorithms let us define some notations. There is one start state denoted by $\mathcal{S}$, and one final state denoted by $\mathcal{F}$. Also let us call states immediately following a terminal transition **terminal state**, and states at which pop transitions are defined **pop state**. Some more notations are as follows.

- $first(l)$ returns the first state of layer l.
- $last(l)$ returns the last state of layer l.
- $layer(s)$ returns the layer state s belongs to.
- $bout(l)$ returns the states from which layer l branches out.
- $bin(l)$ returns the states to which layer l returns.
- $terminal(l)$ returns a set of terminal edges in layer l.
- $nonterminal(l)$ returns a set of nonterminal edges in layer l.
- $\vec{ij}$ denotes the edge between states i and j.
- $[i,j]$ denotes the network segment between states i and j.
- $W_{a\sim b}$ is an observation sequence covering from a_{th} to b_{th} observations.

3. Reestimation Algorithm

PRTN is a RTN with probabilistic transitions and words[1] that can be estimated from sample sentences by means of statistical techniques. we present a reestimation algorithm for obtaining the probabilities of transitions and the observation symbols (words) defined at each terminal transition. Inside-Outside algorithm provides a formal basis for estimating parameters of context free languages such that the probabilities of the observation sequences (sample sentences) are maximized. The reestimation algorithm iteratively estimates the probabilistic parameters until the probability of sample sentence(s) reaches a certain stability. The reestimation algorithm for PRTN is a variation of Inside-Outside algorithm customized for the representation. The algorithm to be discussed is defined only for well formed observation sequences.

Definition 1 An observation sequence is *well formed* if there exists at least a path that generates the sequence in the network and starts at $\mathcal{S}$ and ends at $\mathcal{F}$.

Let an observation sequence of length N denoted by

$$W = W_1 W_2 \cdots W_N .$$

We start explaining the reestimation algorithm by defining Inside-probability.

The Inside probability denoted by $P_I(i)_{s\sim t}$ of state i is the probability that a portion of $layer(i)$

[1]we do not consider probabilistic states in this paper.

$E \rightarrow T + E$
$E \rightarrow T$
$T \rightarrow F * T$
$T \rightarrow F$
$F \rightarrow (E)$
$F \rightarrow a$

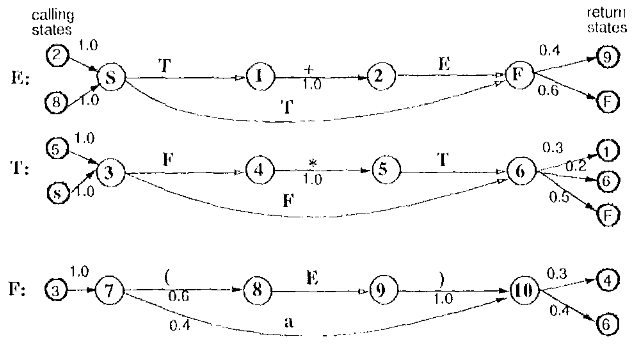

Figure 1: Illustration of PRTN. A parse is composed of dard-headed transitions.

(from state i to the last state of the layer) generates $W_{s\sim t}$. That is, it is the probability that a certain fragment of a layer generates a certain segment of an input sentence, and this can be computed by summing the probabilities of all the possible paths in the layer segment that generate the given input segment.

$$\begin{aligned} P_I(i)_{s\sim t} &= P([i,e] \rightarrow W_{s\sim t}|\lambda) \\ where \quad e &= last(layer(i)) \,. \end{aligned}$$

More constructive representation of Inside probability is then

$$\begin{aligned} P_I(i)_{s\sim t} \quad = \quad & \sum_k a_{ik} b(\overrightarrow{ik}, W_s) P_I(k)_{s+1\sim t} \;+ \\ & \sum_j \sum_{r=s}^{t} a_{ij} a_{uv} P_I(j)_{s\sim r} P_I(v)_{r+1\sim t} . \end{aligned}$$

$$\begin{aligned} where \quad & \overrightarrow{ik} \in terminal(layer(i)), \\ & \overrightarrow{ij} \in nonterminal(layer(i)), \\ & u = last(layer(j)), \\ & v \in bin(layer(j)), \\ & layer(i) = layer(v) \,. \end{aligned}$$

The paths starting at state i are classified into two cases according to the type of immediate transition from i: it can be of terminal or nonterminal type. In case of terminal, after the probability of the terminal transition is taken into account, the rest of the layer segment is responsible for the input segment short of one word just generated by the terminal transition. In case of nonterminal, first the transition probabilities (push and respective pop transitions) are multiplied, then depending on the coverage of the nonterminal transition (sublayer) the rest of the current layer is responsible for the remaining input sequence after done by the sublayer. After the last observation is made, the last state (pop state) of $layer(i)$ should be reached.

$$P_I(i)_{t+1\sim t} = \begin{cases} 1 & \text{if } i = last(layer(i)), \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 2 is the pictorial view of the Inside probability. A well formed sequence can begin only at state S, thus to be strict, $P_I(S)$ has additional product term $P(S)$ that can be computed also using Inside-Outside algorithm. Now define the Outside probability.

The Outside probability denoted by $P_O(i,j)_{s\sim t}$ is the probability that partial sequences, $W_{1\sim s-1}$ and $W_{t+1\sim N}$, are generated provided that the partial sequence, $W_{s\sim t}$, is generated by $[i,j]$ given a model, λ. This is a complementary point of Inside-probability. This time, we look at the outside of given layer segment and input segment. Assuming a given layer segment generates a given input segment, we want to compute the probability that the surrounding portion of the whole PRTN generates the rest of the input sequence.

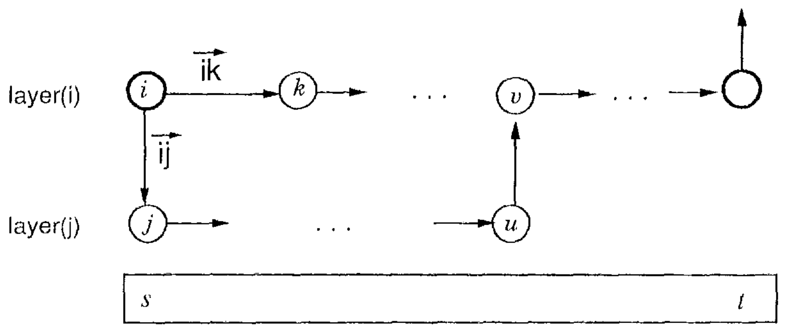

Figure 2: Illustration of Inside probability.

The Outside probability is computed first by considering the current layer consisting of two parts after excluding $[i,j]$ that are captured in Inside-probability. Beyond the current layer is simply an Outside probability with respect to the current layer.
And by definition,

$$\begin{aligned} P_O(i,j)_{s\sim t} &= P([\mathcal{S},i] \rightarrow W_{1\sim s-1},\ [j,\mathcal{F}] \rightarrow W_{t+1\sim N} \mid \lambda) \\ &= \sum_x \sum_{a=1}^{s} \sum_{b=t+1}^{N} a_{xf} a_{ey} \times P_I^q(f,i)_{a\sim s-1} P_I(j)_{t+1\sim b} P_O(x,y)_{a\sim b} \,. \end{aligned}$$

$$\begin{aligned} where \quad x &\in bout(layer(i)), \\ y &\in bin(layer(i)), \\ f &= first(layer(i)), \\ e &= last(layer(i)), \\ layer(i) &= layer(j), \\ layer(x) &= layer(y). \end{aligned}$$

x represents a state from which $layer(i)$ branches out, and y represents a state to which $layer(j)$ returns to. Every time a different combination of left and right sequences with respect to $W_{s\sim t}$ is tried in the layer states i and j belong to, the rest of remaining sequences is the Outside probability at the layer above $layer(i)$.

When there is no subsequence to the right of $W_{a\sim b}$ $(i.e., b = N)$,

$$P_O(i,j)_{a\sim N} = 1 \,.$$

Fig. 3 shows the network configuration in computing the Outside probability. $P_I^q(f,i)_{a\sim s-1}$ is the probability that sequence, $W_{a\sim s-1}$, is generated by $layer(i)$ left to state i. $P_I(j)_{t+1\sim b}$ is the probability that sequence $W_{t+1\sim b}$ is generated by $layer(i)$ right to state j. The portions of W not covered by $W_{a\sim b}$ is then left to the parent layers of $layer(i)$.

$P_I^q(f,i)_{s\sim t}$ is a slight variation of Inside probability in which $P_I(f)_{a\sim b}$'s in the Inside probability formula are replaced by $P_I^q(f,i)_{a\sim b}$. Its actual computation is done as follows:

$$P_I^q(f,i)_{s\sim t} = \begin{cases} P_I(f)_{s\sim t} & \text{if } s \le t, \\ 1 & \text{if } s > t \text{ and } f = i, \\ 0 & \text{if } s > t \text{ and } f \ne i. \end{cases}$$

It is basically the same as Inside probability except that it carries a state identification i to check the validity of stop state. If there are observations left for generation $(s \le t)$, things are done just as in computing Inside probability, ignoring i. When boundary point is reached $(s > t)$, if the last state is i, it returns 1, and 0, otherwise.

The probability of an observation sequence can be computed using Inside probability as

$$\begin{aligned} P(W|\lambda) &= P([\mathcal{S},\mathcal{F}] \rightarrow W_{1\sim N} \mid \lambda) \\ &= P_I(\mathcal{S})_{1\sim N} \,. \end{aligned}$$

Now we can derive the reestimation algorithm for $\mathcal{A}$ and $\mathcal{B}$ using the Inside and Outside probabilities. As the result of constrained maximization of Baum's auxiliary function, we have the following form of reestimation for each transition (Rabiner 1989).

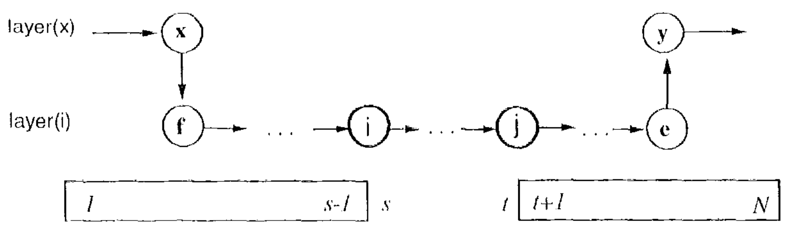

Figure 3: Illustration of Outside probability.

$$\bar{a}_{ij} = \frac{\text{expected no. of transitions from i to j}}{\text{expected no. of transitions from i}} .$$

The expected frequency is defined for each of the three types of transition. For a terminal transition,

$$E_t(\vec{ij}) = \frac{\sum_{r=1}^{N} a_{ij} b(\vec{ij}, W_r) P_O(i,j)_{r\sim r}}{P(W \mid \lambda)} .$$

For a nonterminal transition,

$$E_{nt}(\vec{ij}) = \frac{\sum_{s=1}^{N} \sum_{t=s}^{N} a_{ij} P_I(j)_{s\sim t} a_{uv} P_O(i,v)_{s\sim t}}{P(W \mid \lambda)} .$$

$$\begin{aligned} \textit{where} \quad & u = last(layer(j)),\ v \in bin(layer(j)), \\ & layer(i) = layer(v),\ layer(j) = layer(u), \\ & \vec{uv} \text{ is a pop transition} . \end{aligned}$$

For a pop transition,

$$E_{pop}(\vec{ij}) = \frac{\sum_{s=1}^{N} \sum_{t=s}^{N} a_{uv} P_I(v)_{s\sim t} a_{ij} P_O(u,j)_{s\sim t}}{P(W \mid \lambda)} .$$

$$\begin{aligned} \textit{where} \quad & u \in bout(layer(i)), \\ & j \in bin(layer(i)), \\ & v = first(layer(i)), \\ & layer(u) = layer(j), \\ & layer(v) = layer(i), \\ & \vec{uv} \text{ is a nonterminal transition} . \end{aligned}$$

Considering that transitions of terminal and nonterminal types can occur together at a state, the reestimation for terminal transitions is done as follows:

$$\bar{a}_{ij} = \frac{E_t(\vec{ij})}{\sum_k E_t(\vec{ik}) + \sum_k E_{nt}(\vec{ik})} .$$

For nonterminal transitions,

$$\bar{a}_{ij} = \frac{E_{nt}(\vec{ij})}{\sum_k E_t(\vec{ik}) + \sum_k E_{nt}(\vec{ik})} .$$

And for pop transitions, notice that only pop transitions are possible at a pop state,

$$\bar{a}_{ij} = \frac{E_{pop}(\vec{ij})}{\sum_k E_{pop}(\vec{ik})} .$$

For a terminal transition $\vec{ij}$ and an observation symbol w,

$$\bar{b}(\vec{ij}, w) = \frac{\sum_{t \text{ s.t. } W_t = w} a_{ij} b(\vec{ij}, W_t) P_O(i,j)_{t\sim t}}{\sum_{t=1}^{N} a_{ij} b(\vec{ij}, W_t) P_O(i,j)_{t\sim t}} .$$

The reestimation process continues until the probability of the observation sequences reaches a certain stability. It is not unusual to assume that the training set can be very large, and even grow indefinitely in non trivial applications in which case additive training can be tried using a smoothing technique as in (Jarre and Pieraccini 1987).

The complexity of Inside-Outside algorithm is $O(N^3)$ both in the number of states and input length (Lari 1990). The efficiency comes from the fact that the algorithm successfully exploits the context-freeness. For instance, the generation of substrings by a nonterminal is independent of the surroundings of the nonterminal, and this is how the product of the Inside and Outside probabilities works and the complexity is derived.

4. Conclusion

Recently several probabilistic parsing approaches have been suggested such as SCFG, probabilistic GLR, and probabilistic link grammar (Lafferty, 1992). Kupiec extended the reestimation algorithm for SCFG to cover non-Chomsky normal forms (Carroll, 1993). This paper further advances the trend by implanting the Inside-Outside algorithm on the plain topology of RTN which distinguishes itself from Kupiec's work.

References

[1] Baker, J. K. (1979). Trainable Grammars for Speech Recognition. *Speech Communication Papers for the 97th Meeting of the acoustical Society of America* (D.H. Klatt & J.J. Wolf, eds): 547-550.

[2] Baum, L. E. (1972). An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of a Markov Process." *Inequalities* 3: 1-8.

[3] Carroll J., and Briscoe E. (1993). Generalized probabilistic LR parsing of natural language (Corpora) with unification-based grammars. *ACL* **19** (1). 25-59.

[4] Jarre, A., and Pieraccini, R. (1987). "Some Experiments on HMM Speaker Adaptation," *Proceedings of ICASSP*, paper 29.5.

[5] John Lafferty., Daniel Sleator. and Davy Temperley. (1992). Grammatical trigrams: a probabilistic model of link grammar. In *Proceedings of AAAI Fall symposium on Probabilistic Approaches to Natural Language Processing*, Cambridge, MA. 89-97.

[6] Jelinek, F. Lafferty, J. D. and Mercer R. L. (1990). Basic Methods of Probabilistic Context Free Grammars. *IBM RC 16374*. IBM Continuous Speech Recognition Group.

[7] Kupiec, Julian (1991). A Trellis-Based Algorithm For Estimating the Parameters of a Hidden Stochastic Context-Free Grammar. *Proceedings, Speech and Natural Language Workshop*. sponsored by DARPA. Pacific Grove: 241-246.

[8] Lari, K.; and Young, S. J. (1991). "Applications of stochastic context-free grammars using the Inside-Outside algorithm." *Computer Speech and Language* 5: 237-257.

[9] Rabiner, Lawrence R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* 77 (2).