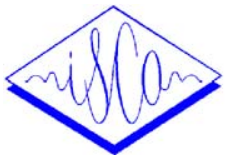

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

LEARNING OF STOCHASTIC CONTEXT-FREE GRAMMARS BY MEANS OF ESTIMATION ALGORITHMS

Joan-Andreu Sánchez *José-Miguel Benedí*

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia (Spain)
e-mail: {jandreu,jbenedi}@dsic.upv.es

ABSTRACT

The use of the Inside-Outside (IO) algorithm for the estimation of the probability distributions of Stochastic Context-Free Grammars is characterized by the use of all the derivations in the learning process. However, its application in real tasks for Language Modeling is restricted due to the time complexity per iteration and the large number of iterations that it needs to converge. Alternatively, several estimations algorithms which consider a certain subset of derivations in the estimation process have been proposed elsewhere. This set of derivations can be chosen according to structural criteria, or by selecting the k-best derivations. These alternatives are studied in this paper, and they are tested on the corpus of the Wall Street Journal processed in the Penn Treebank project.

1. INTRODUCTION

In this paper, we study the learning of Stochastic Context-Free Grammars (SCFGs) by means of estimation algorithms. One of the most well-known methods for learning SCFGs is the Inside-Outside (IO) algorithm [1, 7, 3]. The use of the IO algorithm for Language Modeling in Speech Recognition Systems for real tasks is restricted due to the time complexity per iteration and the large number of iterations needed to converge. Alternatively, an algorithm based on the Viterbi Score (VS) can be considered [7, 3]. The convergence of the VS algorithm is faster than the IO algorithm, since the VS algorithm only considers the information obtained from the best derivation. However, the obtained SCFGs are, in general, not as well-learned.

Another possibility for estimating SCFGs from only a certain subset of derivations can be explored. In order to select this subset of derivations, two alternatives have been proposed: from structural information content in bracketed corpora [8] and from the k-best derivations [11]. In the first alternative, a modification of the IO algorithm which learns SCFGs from partially bracketed corpora was proposed [8]. In the second alternative, a new algorithm for the estimation of the probability distributions of SCFGs from the k-best derivations was proposed. This algorithm considers more information than the VS algorithm, and therefore the SCFGs are, in general, better estimated.

In this work, first we explore the alternative of estimation from structural information content in the bracketed corpora, analyzing the behavior of the extension of the IO algorithm (IOb) proposed in [8] and, describing a new algorithm based on the VS algorithm (VSb). Secondly, we present the kVS algorithm [11] together with the complexity study. Finally, experiments with the part of Wall Street Journal processed in the Penn Treebank project are also reported in order to illustrate the behavior of the algorithms.

Work partially supported by the Spanish CICYT under contract TIC98/0423-C06.

2. ESTIMATION FROM A SET OF DERIVATIONS

A *Context-Free Grammar* (CFG) G is a four-tuple (N, Σ, P, S), where N is a finite set of non-terminal symbols, Σ is a finite set of terminal symbols $(N \cap \Sigma = \emptyset)$, P is a finite set of rules of the form $A \rightarrow \alpha$ ($A \in N$ and $\alpha \in (N \cup \Sigma)^+$) (we only consider grammars with no empty rules) and S is the initial symbol $(S \in N)$. A CFG in Chomsky Normal Form is a CFG in which the rules are of the form $A \rightarrow BC$ or $A \rightarrow a$ ($A, B, C \in N$ and $a \in \Sigma$). A *left-derivation* of $x \in \Sigma^+$ in G is a sequence of rules $d_x = (p_1, p_2 \ldots, p_m)$, $m \geq 1$ such that: $(S \overset{p_1}{\Rightarrow} \alpha_1 \overset{p_2}{\Rightarrow} \alpha_2 \ldots \overset{p_m}{\Rightarrow} x)$, where $\alpha_i \in (N \cup \Sigma)^+$, $1 \leq i \leq m-1$, and p_i rewrites the left-most non-terminal of α_{i-1}. The *language generated* by G is defined as $L(G) = \{x \in \Sigma^+ \mid S \overset{+}{\Rightarrow} x\}$.

A *Stochastic Context-Free Grammar* (SCFG) G_s is defined as a pair (G, q) where G is a CFG and $q : P \rightarrow]0, 1]$ is a probability function of rule application such that $\forall A \in N$: $\sum_{\alpha \in (N \cup \Sigma)^+} q(A \rightarrow \alpha) = 1$. Let d_x be a left-derivation (derivation from now on) of the string x. The expression $\mathrm{N}(A \rightarrow \alpha, d_x)$ represents the number of times that the rule $A \rightarrow \alpha$ has been used in the derivation d_x and $\mathrm{N}(A, d_x)$ is the number of times that the non-terminal A has been derived in d_x. We define

the *probability* of the derivation d_x of the string x as: $\Pr(x, d_x \mid G_s) = \prod_{(A\to\alpha)\in P} q(A \to \alpha)^{\mathrm{N}(A\to\alpha, d_x)}$. Let Δ_x be a finite set of different derivations of the string x. We define the *probability* of the string x with respect to Δ_x as: $\Pr(x, \Delta_x \mid G_s) = \sum_{d_x\in\Delta_x} \Pr(x, d_x \mid G_s)$. If Δ_x is the set of all possible derivations, then we have the *probability* of the string x and it is noted as: $\Pr(x \mid G_s) = \sum_{\forall d_x} \Pr(x, d_x \mid G_s)$. We define the *probability of the best derivation* of the string x from a set of derivations Δ_x as: $\widehat{\Pr}(x, \Delta_x \mid G_s) = \max_{d_x\in\Delta_x} \Pr(x, d_x \mid G_s)$, and we define the *best derivation*, $\widehat{d}_x$, as the argument which maximizes this function. The *language generated* by G_s is defined as $L(G_s) = \{x \in L(G) | \Pr(x \mid G_s) > 0\}$.

The probability functions associated to the rules of a SCFG can be adequately modified in order to represent the stochastic information content in a set of training strings. In such a case, it is necessary to choose a function to be optimized and a context in which to do this process. In this work, we use the context of the Growth Transformations [2] to optimize the functions that depend on the training sample. Several of the functions to be maximized will be considered.

Given an initial SCFG G_s, a training sample Ω and a finite set[1] $\Delta_\Omega = \bigcup_{x\in\Omega} \Delta_x$, the following function can be used to modify the probabilities ($\forall(A \to \alpha) \in P$):

$$q'(A \to \alpha) = \frac{\sum_{x\in\Omega} \frac{1}{\Pr(x,\Delta_x|G_s)} \sum_{d_x\in\Delta_x} \mathrm{N}(A \to \alpha, d_x)\Pr(x, d_x \mid G_s)}{\sum_{x\in\Omega} \frac{1}{\Pr(x,\Delta_x|G_s)} \sum_{d_x\in\Delta_x} \mathrm{N}(A, d_x)\Pr(x, d_x \mid G_s)}. \quad (1)$$

This transformation attempts to improve the function $\Pr(\Omega, \Delta_\Omega \mid G_s) = \prod_{x\in\Omega} \Pr(x, \Delta_x \mid G_s)$ guaranteeing that $\Pr(\Omega, \Delta_\Omega \mid G_s') \geq \Pr(\Omega, \Delta_\Omega \mid G_s)$. It can be proven that this transformation guarantees that the estimated models are consistent [10].

Transformation (1) is used in the IO algorithm when Δ_x has the maximum number of derivations of x, while transformation (1) is used in the VS algorithm when Δ_x has only the best derivation over all possible derivations. In the first algorithm, the function to be maximized is the likelihood of the sample, while in the second algorithm, the function to be maximized is the likelihood of the best parse of the sample.

Based on transformation (1), new estimation algorithms can be defined in which the set of derivations used for the estimation can be chosen according to structural criteria or by choosing the k-best derivations. Both proposals are described below.

2.1. Estimating from the k-best derivations

Based on transformation (1), an algorithm is proposed in which the k-best derivations is computed in each iteration following the strategy of the VS algorithm. This algorithm starts with an initial grammar and then an iterative process begins. In each iteration, the k-best derivations of each string in the sample are obtained. The counts of the numerator and denominator of (1) are accumulated for those rules appearing in these derivations. At the end of the iteration, transformation (1) is applied to each rule. This iterative process continues until a local maximum of the function being optimized is achieved.

There exists an efficient algorithm to compute the k-best derivations of a string which is based on a Dynamic Programming scheme [6]. The time complexity to obtain the best derivation of a string x is $O(|x|^3|P|)$. The time complexity to obtain each new derivation is in practice *approximately* proportional to the number of rules of the previous derivation times a logarithmic factor [6].

To study the time complexity of the proposed estimation algorithm, we assume that the SCFG is in Chomsky Normal Form. The set of k-best derivations for small values of k is calculated for each string x in the sample with a time complexity of *approximately* $O(|x|^3|P|)$ [6]. Therefore, the time complexity of the algorithm per iteration is $O(|\Omega|n^3|P|)$, where n is the size of the longest string in the sample.

2.2. Estimating from structural information

Another estimation algorithm can be defined from transformation (1) in which the set of derivations used in the learning process is chosen according to the structural information content in the training sample. This structural information can be incorporated in the sample by parsing the strings (manually or automatically) according to some criteria (syntactic or semantic, basically), and registering the information typically by parentheses. In the estimation process, only those derivations which are compatible with the parsing are considered as appropriate derivations. We now describe how the structural information represented by parentheses can be treated.

Informally, a partially bracketed corpus is a set of sentences which is annotated with parentheses marking constituent frontiers [8]. More precisely, a bracketed corpus Ω is a set of pairs (x, B) where x is a string and B the bracketing of x.

Given the string $x = x_1x_2\ldots x_n$, the pair of integers (i, j), $1 \leq i \leq j \leq n$ forms a span of x. A span (i, j) delimits substring $x_i \ldots x_j$.

A bracketing B of x is a finite set of spans on x, $B = \{(i, j)|1 \leq i \leq j \leq n\}$ such that every two spans (i, j), $(k, l) \in B$ accomplishes that $i \leq k \leq l \leq j$, or $k \leq i \leq j \leq l$. In such a case the spans do not overlap.

Given (x, B), any parse of x must respect the limits defined by B. The following concepts establish the conditions for a derivation of x to be compatible with B. First, we define the bracketing defined by a derivation.

Let (x, B) be a bracketed string, and let d_x be a derivation of x with the SCFG G_s. If the SCFG does not have

[1] We abuse the notation and consider this union operation to be a union between multisets which maintains the result as a multiset.

useless symbols, then every non-terminal that appears in every sentential form of the derivation generates a substring $x_i \dots x_j$ of x, $1 \leq i \leq j \leq |x|$ and defines a span (i, j). A derivation of x is compatible with B if all the spans defined by it are compatible with B.

Given a SCFG and a bracketed corpus Ω, for each bracketed string (x, B), we define the function:

$$c(i,j) = \begin{cases} 1 & \text{if } (i,j) \text{ does not overlaps any } b \in B \\ 0 & \text{otherwise} \end{cases}$$

This function filters those derivations (or partial derivations) whose parsing is not compatible with the bracketing defined on the sample.

The IO algorithm used in the estimation of the rule probabilities is modified to take advantage of the bracketing of a string by using the function $c()$ previously defined [8]. The *inside* and *outside* probabilities are adequately modified in order to consider only those partial parses which are compatible with the bracketing defined on the strings. The expressions used to modify the probabilities are the classical ones, which can be viewed as a special case of expression (1). These modifications do not affect the time complexity of the algorithm per iteration which is $O(|\Omega| n^3 |P|)$.

Another possible estimation algorithm which considers structural information content in the sample, and which is based on the Viterbi-Score algorithm can be defined. The selection of the best derivation is made from those derivations that are compatible with the bracketing defined on the sample. The counts of the rules which appear in these derivations are accumulated, and finally, expression (1) is applied. The time complexity of this new algorithm is $O(|\Omega| n^3 |P|)$.

All the above mentioned algorithms obtain a local maximum of the function which is being maximized. No theoretical relation can be established about the goodness of the models which are estimated by each algorithm. In the following section we compare the estimation algorithms in an experiment.

3. EXPERIMENTS WITH THE PENN TREEBANK CORPUS

The corpus used in the experiments was the part of the Wall Street Journal which had been processed in the Penn Treebank project[2] [5]. This corpus consists of English texts collected from the Wall Street Journal from editions of the late eighties. It contains approximately one million words. This corpus was automatically labelled, analyzed and manually checked as described in [5] (an example is shown in Figure 1). There are two kinds of labelling: a part of speech (POStag) labelling and a syntactic labelling. The size of the vocabulary is greater than 25,000 different words, the POStag vocabulary is composed of 45 labels[3] and the syntactic vocabulary is composed of 14 labels.

```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN di-
rector) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
```

Figure 1: The sentence *Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.* labelled and analyzed in the Penn Treebank project.

We decided to work only with the POStag labelling, since the vocabulary of the original corpus was too large for the experiments. The corpus was divided into sentences according to the bracketing. In this way, we obtained a corpus whose main characteristics are shown in Table 1.

Table 1: Characteristics of the Penn Treebank corpus once it was divided into sentences.

No. of senten.	Average length	Standard deviation	Min. length	Max. length
49,207	23.61	11.13	1	249

Given the time complexity of the bracketed IO algorithm, we decided not to consider the sentences with more than 15 POStags in order to reduce the computational effort. For the experiments, the corpus was divided into a training corpus (directories 00 to 19) and a test corpus (directories 20 to 24). The characteristics of these sets can be seen in Table 2. The perplexity per word was used to evaluate the goodness of the obtained model [4]. The test set perplexity[4] with 3-grams was 9.63.

An initial SCFG in Chomsky Normal Form to be estimated was constructed. This SCFG had the maximum number of rules which can be composed with 45 terminal symbols (the number of POStags) and 14 non-terminal symbols (the number of syntactic labels), which sums up

[2]Release 2 of this data set can be obtained from the Linguistic Data Consortium with Catalogue number LDC94T4B (http://www.ldc.upenn.edu/ldc/noframe.html)

[3]There are 48 labels defined in [5], however three do not appear in the corpus

[4]The values were computed with the software tool described in [9] (Release 2.04 is available in http://svr-www.eng.cam.ac.uk/∼prc14/toolkit.html).

Table 2: Characteristics of the data sets defined for the experiments when the sentences with more than 15 POStags were removed.

Data set	No. of senten.	Average length	Standard deviation
Training	9,933	10.67	3.46
Test	2,295	10.51	3.55

to $3,374$ rules. The probabilities were randomly generated and three seeds were tested. Given that the results were similar, only one of the seeds is reported.

Table 3 shows the perplexity of the final model for the algorithms considered (VS, kVS, IOb, VSb). First, observe the good results obtained with the IOb algorithm compared with the other estimation algorithms. It should be noted that given the characteristics of the initial model, this is the algorithm which uses more information, and, therefore, the convergence is much slower than for the other algorithms. Second, we can see that the kVS algorithm can obtain better results than the VS algorithm, even by using small values of k. This means that this algorithm can be an appropriate alternative to the VS algorithm.

Table 3: Test set perplexity for different algorithms. The training set was composed by all the other partitions.

Algorithm	VS	kVS ($k = 7$)	IOb	VSb
Test set per.	21.56	20.65	13.14	21.84

Taking into account that the VS, kVS and VSb algorithms are very sensitive to the initial values of the probabilities, we explored the possibility of initially estimating a few iterations using a more robust algorithm, such as the IO or the IOb algorithm, and then to continue with the other algorithm until convergence. We tested this proposal, but no significant improvement was obtained.

4. CONCLUSIONS AND FUTURE WORK

In this work, we have studied algorithms which use a subset of derivations in the estimation process to learn the distributions probabilities of a SCFG . This subset is chosen according to structural criteria or by selecting the k-best derivations. We have tested these algorithms in an experimental work.

For future work, we propose studying a way to obtain the initial grammar from the structural information content in the sample. In addition, we propose studying how to integrate these models in real tasks of Language Modeling.

5. REFERENCES

[1] J.K. Baker. Trainable grammars for speech recognition. In Klatt and Wolf, editors, *Speech Communications for the 97th Meeting of the Acoustical Society of America*, pages 31–35. Acoustical Society of America, June 1979.

[2] L.E. Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, 3:1–8, 1972.

[3] F. Casacuberta. Growth transformations for probabilistic functions of stochastic grammars. *IJPRAI*, 10(3):183–201, 1996.

[4] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[5] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[6] A. Marzal. *Cálculo de las k mejores soluciones a problemas de programación dinámica*. Ph. d. dissertation, Universidad Politécnica de Valencia, 1994.

[7] H. Ney. Stochastic grammars and pattern recognition. In P. Laface and R. De Mori, editors, *Speech Recognition and Understanding. Recent Advances*, pages 319–344. Springer-Verlag, 1992.

[8] F. Pereira and Y. Schabes. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*, pages 128–135. University of Delaware, 1992.

[9] R. Rosenfeld. The cmu statistical language modeling toolkit and its use in the 1994 arpa csr evaluation. In *ARPA Spoken Language Technology Workshop*, Austin, Texas, USA, 1995.

[10] J.A. Sánchez and J.M. Benedí. Consistency of stochastic context-free grammmars from probabilistic estimation based on growth transformation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(9):1052–1055, 1997.

[11] J.A. Sánchez and J.M. Benedí. Estimation of the probability distributions of stochastic context-free grammars from the k-best derivations. In *In 5th International Conference on Spoken Language Processing*, pages 2495–2498, Sidney, Australia, 1998.