Probabilistic Context Free Grammars

Daniel Marcu
marcu@isi.edu

1 Probabilistic Context Free Grammars (PCFG)

Definition 1 *A PCFG is a tuple (N, W, P, S, D), where*

- *N is a set of nonterminal symbols,*
- *W is a set of terminal symbols,*
- *P is a set of production rules $\alpha \to \beta$ so that $\alpha \in N$ and $\beta \in (N \cup W)^*$*
- *$S \in N$ is the start symbol*
- *D is a function that assigns a probability to each rule $0 \leq D(\alpha \to \beta) \leq 1$.*

Background Information: Probability Theory
Let ζ be a discrete event space.
P is a probability distribution over ζ iff

1. $0 \leq P(A) \leq 1, \forall A \in \zeta$
2. $\sum_{A \in \zeta} P(A) = 1$

P may be a function of some vectors of parameters Θ.
For a parameter setting, we write $P(A|\Theta)$.
The parameter space $\Omega = \{\Theta | P(A|\Theta)\}$ is a probability distribution over ζ.
Example: Let $\aleph$ be the set of all sequence $\langle w_1, w_2, \ldots, w_m \rangle$ where each $w_i \in W$.
Let τ be the set of all the parse trees that can be generated by a context free grammar G so that each parse tree spans a member of $\aleph$.
Want to define $P(x, y|\Theta_G)$, so that P is a probability distribution.

- the prob of any string $\langle w_1, w_2, \ldots, w_m \rangle$ is a number between 0 and 1.
- the sum of the prob of all strings is 1.

Problems for PCFGs

1. How can one compute the prob of tree T for a sentence $w_{1..m}$, $P(T|w_{1..m}, G)$.

2. How can one compute the prob. of a sentence $w_{1..m}$ given a grammar $P(w_{1..m}|G)$.

3. How should one associate probabilities to rules so that they define a probability distribution?

4. Assuming that one is given a corpus of parse trees, how can one estimate the probabilities of the rules?

5. Given a string and a PCFG, how can one determine the "best" parse tree?

6. Given a grammar and a corpus of sentences, how can one estimate the probability of the rules of the grammar?

1.1 Computing the probability of a tree

$P(T) = \prod_{i=1..n} P(\alpha_i \rightarrow \beta_i|\alpha_i)$ for all the nonterminals $\alpha_i \in N$ in a derivation.

1.2 Computing the probability of a sentence

$P(s) = \sum_{T \in \tau(s)} P(T)$

1.3 Associating probabilities with grammar rules

Probabilities should be associated so that

1. $\forall T \in \tau, P(T) \geq 0$	1. $\forall s \in S, P(s) \geq 0$
2. $\sum_{T \in \tau} P(T) = 1$	2. $\sum_{s \in S} P(s) = 1$

Exercise: Show that these two sets of conditions are equivalent.

Counterexample:

Let $G = (\{S\}, \{a\}, S, \{S \rightarrow a, S \rightarrow SS\}, \{p(S \rightarrow a) = 1/3, p(S \rightarrow SS) = 2/3\})$.
P(a) = 1/3.
P(aa) = 2/3 * 1/3 * 1/3 = 2/27.
P(aaa) = $(2/3)^2$ * $(1/3)^3$ * 2 = 8/243
But 1/3 + 2/27 + 8/243 + ... = 1/2 !!!
In general, if $p(S \rightarrow a) = x$ and $p(S \rightarrow SS) = 1 - x$, one can prove that

$$\sum_{n=1..\infty} P(a^n) = min(1, x/(1-x))$$

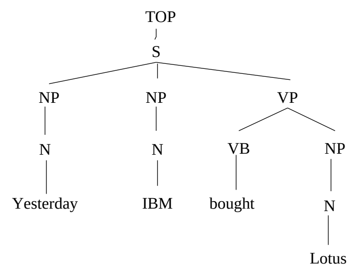
In order for the probabilities of the rules to determine a distribution, two necessary conditions must be met [Booth and Thompson, 1973]:

- For all nonterminals $\alpha \in N, \sum_{(\alpha \to \beta) \in P} P(\alpha \to \beta | \alpha) = 1$. (NOT $P(\alpha \to \beta)$; NOT $P(\alpha \to \beta | \beta)$)
- The grammar should not contain self-looping rules of probability 1 [because some probability mass would be lost in derivations that never terminate].

Example:

S = TOP
N = {TOP, S, NP, VP, VB, NNP}
W = {gave, bought, U.S., IBM, Lotus, yesterday}

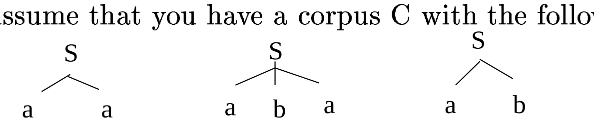
Rules	Probabilities (D)
TOP $\to$ S	1.0
S $\to$ NP VP	0.8
S $\to$ NP NP VP	0.2
VP $\to$ VB NP	0.6
VP $\to$ VB NP NP	0.4
NP $\to$ N	1.0
VB $\to$ gave	0.6
VB $\to$ bought	0.4
N $\to$ Lotus	0.8
N $\to$ U.S.	0.1
N $\to$ IBM	0.1



Probability(T) =
P(TOP -> S | S) x
P(S -> NP NP VP | S) x
P(NP -> N | NP) x
x ... x
P(N -> Lotus | N).

1.4 Estimating the probabilities of the rules of a grammar from a corpus of parse trees

Assume that you have a corpus C with the following parse trees:



9 occurrences 2 occurrences 4 occurrences

Let $G = (S, S, a, b, S \to aa, S \to aba, S \to ab, D = ?)$

Assume that $p(S \to aa|S) = p_1, p(S \to aba|S) = p_2, p(S \to ab|S) = p_3$. Hence $\Theta = (p_1, p_2, p_3)$ are the parameters of the model.

Question: What is the best grammar (Θ) that characterizes this corpus?

We consider that the best grammar is that that maximizes the likelihood of the corpus C.

$$\begin{cases} L(C|\Theta) = \prod_{i=1..15} P(T_i|\Theta) = p_1^9 \times p_2^2 \times p_3^4 \\ p_1 + p_2 + p_3 = 1 \end{cases}$$

Background Information: The technique of Lagrangian multipliers
Suppose that x is an n dimensional vector and that $f(x)$ is a real valued function on x. Also suppose that we want to determine the maximum of $f(x)$ subject to some constraint $c - g(x) = 0$. Let's call this maximum x^*.
The technique of Lagrange is to form a new function L, called the Lagrangian, by adding a scalar λ multiplied by the constraint to the original function, i.e., $L(x, \lambda) = f(x) + \lambda(c - g(x))$, and to find an *unconstrained* maximum over x and a minimum over λ for $L(x, \lambda)$. Let's denote this solution with $\tilde{x}, \tilde{\lambda}$.
Lagrange has shown that $f(\tilde{x}) = f(x^*) = L(\tilde{x}, \tilde{\lambda})$.

$$\begin{cases} L^{log}(C|\Theta) = 9log(p_1) + 2log(p_2) + 4log(p_3) \\ p_1 + p_2 + p_3 = 1 \end{cases}$$

Want to maximize $Lagrangian(L^{log}) = 9log(p_1) + 2log(p_2) + 4log(p_3) - \lambda_1(p_1 + p_2 + p_3 - 1)$. The partial derivatives must be 0.

$9/p_1 - \lambda_1 = 0$, $2/p_2 - \lambda_1 = 0$, $4/p_3 - \lambda_1 = 0$.
$p_1 = 9/\lambda_1$; $p_2 = 2/\lambda_1$; $p_3 = 4/\lambda_1$;
$(9 + 2 + 4)/\lambda_1 = 1$; $\lambda_1 = 1/15$;
$p_1 = 9/15; p_2 = 2/15; p_3 = 4/15$.

Consider now the general case of a PCFG $G = (S, N, W, P, D)$ and a corpus of n trees, $T_1, \ldots, T_n$. Each T_i contains r_i rules $\alpha_{ij} \to \beta_{ij}, 1 \leq j \leq r_i$.

The likelihood of the corpus is

$$\begin{aligned} L(corpus) &= \prod_{i=1..n} P(T_i) \\ &= \prod_{i=1..n} \prod_{j=1..r_i} P(\alpha_{ij} \to \beta_{ij}|\alpha_{ij}) \\ &= \prod_{(\alpha \to \beta) \in P} P(\alpha \to \beta|\alpha)^{Count(\alpha \to \beta)} \end{aligned}$$

We are interested to determine the parameters $P(\alpha \to \beta|\alpha)$ that maximize $L(Corpus)$.

Theorem 1 *Assume $\Theta = \{p_1, p_2, \ldots, p_n\}$ is a combination of m multinomial distributions $\Omega_1, \Omega_2, \ldots, \Omega_m$. Each Ω_i is a subset of the integers $\{1, 2, \ldots, n\}$ such that the Ω_s form a partition of $\{1, 2, \ldots, n\}$. Assume $\{p_i|i \in \Omega_j\}$ be the parameters of the jth multinomial, with the constraint that $\sum_{i \in \Omega_j} p_i = 1$. Let Ω^i be the multinomial that contains p_i.*

Assume that the likelihood of the data can be written

$$L(X|\Theta) = \prod_{i \in \Omega_1} p_i^{C(i,X)} \prod_{i \in \Omega_2} p_i^{C(i,X)} \cdots \prod_{i \in \Omega_m} p_i^{C(i,X)}$$

where $C(i, X)$ is the count of the event which corresponds to p_i in sample X.

If these conditions are satisfied, then maximizing $L(X|\Theta)$ subject to the constraints $\sum_{i \in \Omega_j} p_i = 1$ gives maximum likelihood estimates for each p_i as

$$\hat{p}_{i\,ML} = \frac{C(i, X)}{\sum_{j \in \Omega^i} Count(j, X)}$$

In the case of our grammar $G = (S, N, W, P, D)$, the parameters $\Theta = \{p_1, p_2, \ldots, p_n\}$ are given by the probabilities associated with each rule. Assume that you order $p_1, p_2, \ldots, p_n$ so that $p_1, p_2, \ldots p_{i1}$ correspond to the rules headed by nonterminal N_1, $p_{i1}, \ldots, p_{i2}$ to the rules headed by nonterminal N_2, and so on. Then $p_1, p_2, \ldots p_{i1}$ give the multinomial distribution Ω_{N_1}, $p_{i1}, \ldots, p_{i2}$ give the multinomial distribution Ω_{N_2}, and so on. In addition, since we want the parameters to yield a probabily distribution, we need to enforce the constraints $\sum_{i \in \Omega_{N_j}} p_i = 1$.

According to the theorem, if $\beta(\alpha) = \{\beta | (\alpha \to \beta) \in P\}$, the maximum likelihood estimates of the parameters are given by

$$\hat{P}(\alpha \to \beta | \alpha) = \frac{Count(\alpha \to \beta)}{\sum_{\gamma \in \beta(\alpha)} Count(\alpha \to \gamma)} = \frac{Count(\alpha \to \beta)}{Count(\alpha)}$$

1.5 Parsing with PCFGs

Assume that $G = (S, N, W, P, D)$ is in Chomsky Normal Form ($A \to BC; A \to a$).

$$T_{best}(s) = argmax_{T \in T(s)} P(T)$$

Notation:

- $p[i, j, k]$ - the maximum probability for a constituent labeled k spanning words $i..j$.
- $b[i, j, k] = m, k_1, k_2$ - backpointer to the rule used to derive $p[i, j, k] (i \leq m \leq j)$.

\# Initialization
for all i, j, k
    $p[i, j, k] = 0$
\# Base case
for $i = 1..n$
    for $k = 1..G$
        if $k \to w_i \in G$
            $p[i, i, k] = P(k \to w_i | k)$
\# Recursive case. Bottom up reconstruction for $s = 2..n$
    for $i = 1..(n - s + 1)$
        $j = i + s - 1$
        for $m = i..(j - 1)$
            for $k = 1..G$
                if $(k \to k_1 k_2 \in G \wedge$
                    $p[i, m, k_1] > 0 \wedge p[m + 1, j, k_2] > 0)$
                    $prob = p[i, m, k_1] \times p[m + 1, j, k_2] \times P(k \to k_1 k_2 | k)$
                    if $(prob > p[i, j, k])$
                        $p[i, j, k] = prob$
                        $b[i, j, k] = \{m, k_1, k_2\}$
end all for loops

Parsing is still $O(n^3 |G|)$.

Practical issues

For very large grammars, this algorithm may be too expensive.

Solution: BEAM SEARCH — for each span $[i, j]$ keep in memory only the nonterminals k_i for which $p[i, j, k_i] > max(p[i, j, k])/B$.

Performance

On sentences < 100 words, 70% labeled recall and 74% labeled precision [Charniak, 1997].

References

[Booth and Thompson, 1973] T.L. Booth and R.A. Thompson. Applying probability measures to abstract languages. *IEEE Transactions on Computers*, C–22(5):442–450, 1973.

[Charniak, 1997] Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI'97)*, pages 598–603, Providence, Rhode Island, July 27–31 1997.