Stochastic Context-Free Grammars

Mark W. Andrews

11th April 2004

1 Introduction: Strings, Grammars & Formal Languages

Stochastic context-free grammars, or SCFGs, are generative systems of stochastic languages. In other words, a SCFG specifies a probability distribution over the set of all possible strings that are concatenated from a finite alphabet $\mathcal{V}$.

Definition 1 An **alphabet** $\mathcal{V}$ is a finite set of symbols. It denoted here by

$$\mathcal{V} = \{w_0, w_1, w_2, w_3 \ldots w_V\}.$$

Definition 2 The kth **power** of an alphabet $\mathcal{V}$, denoted $\mathcal{V}^k$, is the set of strings of length k made from the elements of $\mathcal{V}$.

Example 1 For the alphabet $\mathcal{V} = \{0, 1\}$, where $|\mathcal{V}| = 2$. $\mathcal{V}^1$ will be set of strings, made from the elements of $\mathcal{V}$, that are of length 1, or $\mathcal{V}^1 = \{0, 1\}$. Likewise, $\mathcal{V}^2 = \{00, 01, 10, 11\}$, $\mathcal{V}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$, etc.

Definition 3 $\mathcal{V}^*$, or **Kleene closure** of $\mathcal{V}$, is the union of all powers of the alphabet, or

$$\mathcal{V}^* = \mathcal{V}^1 \cup \mathcal{V}^2 \cup \mathcal{V}^3 \cup \mathcal{V}^4 \ldots.$$

Definition 4 A **formal language** L is defined as $L \subseteq \mathcal{V}^*$. In other words, L is a (possibly infinite) set of strings, each formed by concatenation from a finite set of symbols $\mathcal{V}$.

This may be illustrated by way of examples.

Example 2 L_0 is a language with five member strings concatenated from $\mathcal{V} = \{0, 1\}$,

$$L_0 = \{10, 01, 110, 101, 011\}.$$

Example 3 L_1 is a language where each member string conforms to the expression 01^i0, with $i \in \mathbb{N}$,

$$L_1 = \{00, 010, 0110, 01110, 011110 \ldots\}.$$

Example 4 L_2 is a language where each member strings conforms to the form $0^i 1^i$,

$$L_2 = \{\varepsilon, 01, 0011, 000111, 00001111, \ldots\}.$$

(The empty string ε is also a member of this language. It conforms to $0^i 1^i$ for the case of $i = 0$.)

Example 5 L_3 is a language where each member conforms to $0^i 1^i 0^i$, with $i \in \mathbb{N}$,

$$L_3 = \{\varepsilon, 010, 001100, 000111000, 000011110000, \ldots\}$$

Remark 1 The set of all possible languages, $\{L \colon L \subseteq \mathcal{V}^*\}$, is uncountably infinite, and only a strict subset of this set can be generated or recognized. Such languages are known as the **computable** languages, and their complement are the **uncomputable** languages. A number of generative systems have been shown to be *universal* in the sense that they can generate all computable languages. These universal systems include Turing machines (Turing 1936), Church and Kleene's λ-calculus (Kleene 1950), Church's recursive-functions (Church 1936), Post's tag-systems (Post 1936) and phrase-structure grammars (Chomsky 1956).

Definition 5 A **phrase-structure grammar** $\mathcal{G}$ is specified by a 4-tuple $\langle \mathcal{V}_{NT}, \mathcal{V}_T, P, S \rangle$, where $\mathcal{V}_{NT} = \{N^0, N^1, N^n\}$ is a set of non-terminal symbols, $\mathcal{V}_T = \{w_0, w_1, w_2, w_3 \ldots\ w_V\}$ is a set of terminal symbols (i.e. simply $\mathcal{V}$ from Definition 1), P is a set of production rules and $N^0 \in \mathcal{V}_{NT}$ is a start state. The production rules P take the general form

$$(\mathcal{V}_{NT} \cup \mathcal{V}_T)^* \rightarrow (\mathcal{V}_{NT} \cup \mathcal{V}_T)^*,$$

where $(\mathcal{V}_{NT} \cup \mathcal{V}_T)^*$ is the set of all possible subsets of $(\mathcal{V}_{NT} \cup \mathcal{V}_T)$. The language generated by a phrase-structure grammar is obtained by starting with the start symbol N^0, and applying the production rules.

Often phrase-structure grammars are classified into four types: Type-0, Type-1, Type-2, Type-3. These are defined in terms of the restrictions, if any, that they place on the nature of the production rules.

Definition 6 A **Type-0** grammar is a phrase-structure grammar with no restrictions on the form of its production rules. In other words, the production rules can take the general form

$$(\mathcal{V}_{NT} \cup \mathcal{V}_T)^* \rightarrow (\mathcal{V}_{NT} \cup \mathcal{V}_T)^*.$$

Type-0 languages are also known as **recursively enumerable languages**.

Definition 7 A **Type-1** grammar is a phrase-structure grammar grammars with production rules of the restricted form

$$\alpha N \beta \rightarrow \alpha \gamma \beta,$$

with $\alpha, \beta \in (\mathcal{V}_{NT} \cup \mathcal{V}_T)^*$, $N \in \mathcal{V}_{NT}$ and $\gamma \in (\mathcal{V}_{NT} \cup \mathcal{V}_T)^+$. $(\mathcal{V}_{NT} \cup \mathcal{V}_T)^+$ is the set of all possible subsets of $(\mathcal{V}_{NT} \cup \mathcal{V}_T)$ with the exclusion of the empty string ϵ. Type-1 grammars are also known as **context-sensitive grammars**.

Example 6 The language $L_3 = 0^i 1^i 0^i$, shown in Example 5 above is a context-sensitive language, or a language generated by a Type-1 language. It can be generated by the following phrase-structure grammar: Given N^0,

$$\begin{aligned} N^0 &\to 0N^1N^00 \quad \text{or} \quad 010, \\ N^10 &\to 0N^1, \\ N^11 &\to 11. \end{aligned}$$

Definition 8 A **Type-2** grammar is a phrase-structure grammar whose productions take the following form,

$$N \to \zeta,$$

with $N \in \mathcal{V}_{NT}$ and $\zeta \in (\mathcal{V}_{NT} \cup \mathcal{V}_T)^*$. Type-2 grammars are also known as **context-free grammars**.

Example 7 The language $L_2 = 0^n 1^n$, shown in Example 4 is a context-free language. It may be generated by the Type-2 grammar: Given N^0,

$$\begin{aligned} N^0 &\to 01, \\ N^0 &\to 0N^01. \end{aligned}$$

Definition 9 A **Type-3** grammar is a phrase-structure grammar with production rules of the form

$$N^i \to wN^j,$$

where $N^i, N^j \in \mathcal{V}_{NT}$ and $w \in \mathcal{V}_T$. Type-3 grammars are also known as **regular grammars**, or **finite-state** grammar.

Example 8 The language $L_1 = 01^i0$, shown in Example 3, is a regular language. It may be generated by the rewriting rules: Given N^0,

$$\begin{aligned} N^0 &\to 0N^1, \\ N^1 &\to 1N^1 \quad \text{or} \quad 10. \end{aligned}$$

Definition 10 The **Chomsky hierarchy** of phrase-structure grammars is a hierarchy of the four grammar types. where each numbered grammar type is a strict superset of the grammar types that have higher numbers. In other words,

$$\text{Type-0} \supsetneq \text{Type-1} \supsetneq \text{Type-2} \supsetneq \text{Type-3}.$$

2 Stochastic Grammars

Definition 11 A **stochastic grammar** is a phrase-structure grammar with probabilities on each production rule. For any set of symbols on the LHS of a production, the number of possible ways it can be written must sum to 1.

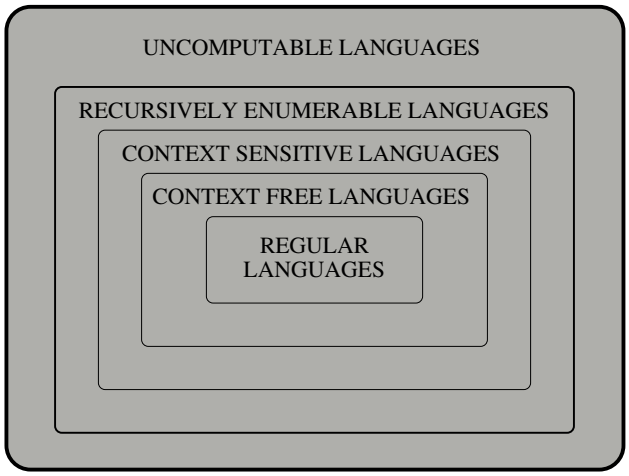

Figure 1: The Chomsky Hierarchy of Languages. This is explained in Definition 10. The space of all possible languages, i.e $\{L\colon L \subseteq \mathcal{V}^*\}$, is shown divided into a series of subsets. The languages in the hierarchy are recursively enumerable, context-sensitive, context-free and regular languages. Each lower language is a strict subset of each higher one.

Definition 12 A **stochastic regular grammar** is a regular grammar such that for each rule $N^i \to w^j N^k$

$$\sum_{\{j,k\}} \mathrm{P}(N^i \to w^j N^k) = 1.$$

Remark 2 A stochastic regular, or Type-3, grammar is identical to a HMM.

Remark 3 It is generally that human languages can not be Type-3 languages, (Chomsky 1956, Chomsky & Miller 1963). Type-3, or regular grammars, can not generate arbitrary numbers of center-embeddings. These types of constructions are, however, present in all human languages. In English, we have sentence such as

The man yelled furiously.

The man who chased the boy yelled furiously.

The man who chased the boy who stole the apple-pie yelled furiously.

The man who chased the boy who stole the apple-pie that was baked on Friday yelled furiously.

Given that this is the case, human languages must be at least Type-2, or context-free. These languages are the lowest type that will accommodate embedded structures like this.

Definition 13 A **stochastic context-free grammar** is a context-free grammar such that for each rule $N^i \to \zeta^j$

$$\sum_{\{j\}} \mathrm{P}(N^i \to \zeta^j) = 1.$$

Definition 14 A context-free grammar is in **Chomsky normal form** if its productions only the the following forms

$$N^i \to N^j N^k,$$
$$N^i \to w^j$$

Remark 4 It has been suggested that human languages are appropriately characterized as subsets of the Type-1 languages, the context-sensitive languages (Savitch 1987). Although context-free grammatical constructions (e.g. nested embeddings) occur routinely in human languages, there are certain constructions that demand greater generative power. The evidence for this comes primarily by way of the cross-serial dependencies found in certain human languages. These constructions are similar to those occurring in English sentences that use the term *respectively*. For example, *Mark, Peter and Zoubin went swimming, running and jogging, respectively*. In this sentence, the nth noun is coupled to the nth verb. Cross-serial dependencies occur most notably, however, in the verb-clusters of Swiss-German and Dutch. For example, the Dutch sentence *Jan Piet Marie de kinderen zag helpen laten zwemmen*, literally translates into English as *Jan Piet Marie the children see help make swim*.

Definition 15 In a SCFG, a non-terminal N^i is said to **dominate** $w_a, \ldots, w_b$ if recursive application of the grammar's rewritting rules, starting at N^i, will lead to the string $w_a, \ldots, w_b$. This property is denoted by

$$N^i \Rightarrow w_a, \ldots, w_b.$$

2.1 Properties of Stochastic Context-Free Grammars

In the case of a regular grammar, for a string $w_0, \ldots w_n$,

$$\mathrm{P}(w_0, \ldots w_n) = \sum_{\{N^0 \ldots N^n\}} \mathrm{P}(w_0, \ldots w_n, N^0 \ldots N^n). \tag{1}$$

In this case, the set of possible sequences of nonterminals $N^0 \ldots N^n$ is straightforward to enumerate. In a SCFG, for a string $w_0, \ldots w_n$,

$$\mathrm{P}(w_0, \ldots w_n) = \sum_{t \in \mathcal{T}} \mathrm{P}(w_0, \ldots w_n, t) \tag{2}$$

where $\mathcal{T}$ is the set of all parse trees for $w_0, \ldots w_n$, and $t \in \mathcal{T}$ is an individual tree. It is more challenging to enumerate the number of possible trees that give rise to a string. The situation benefits from the conditional independences inherent in the tree structures. In particular,

$$\mathrm{P}(N^i \to \zeta^j | \text{ ancestor node of } N^i) = \mathrm{P}(N^i \to \zeta^j),$$

$$\mathrm{P}(N^i \Rightarrow w_a \ldots w_b | w_i \notin w_a \ldots w_b) = \mathrm{P}(N^i \Rightarrow w_a \ldots w_b),$$
$$\forall k, \quad \mathrm{P}(N^i \Rightarrow w_{a+k} \ldots w_{b+k}) = \mathrm{P}(N^i \Rightarrow w_a \ldots w_b).$$

3 Grammar Induction & Learning

Given a data-set of strings $\mathcal{D}$, and a set of possible grammars $\mathcal{G}_0, \mathcal{G}_1, \mathcal{G}_2, \ldots$, with parameters $\theta_0, \theta_1, \theta_2, \ldots$ respectively, the probability of each grammar given the observed data is

$$\overbrace{\mathrm{P}(\mathcal{G}_i|\mathcal{D})}^{\text{Posterior}} = \frac{\overbrace{\mathrm{P}(\mathcal{D}|\mathcal{G}_i)}^{\text{Evidence}}\ \overbrace{\mathrm{P}(\mathcal{G}_i)}^{\text{Prior on } \mathcal{G}_i}}{\sum_{\mathcal{G}_i} \mathrm{P}(\mathcal{D}|\mathcal{G}_i)\mathrm{P}(\mathcal{G}_i)}, \tag{3}$$

where

$$\overbrace{\mathrm{P}(\mathcal{D}|\mathcal{G}_i)}^{\text{Evidence}} = \int d\theta_i\ \mathrm{P}(\overbrace{\mathcal{D}|\theta_i, \mathcal{G}_i}^{\text{Likelihood}})\overbrace{\mathrm{P}(\theta_i|\mathcal{G}_i)}^{\text{Prior on } \theta}. \tag{4}$$

The problem of grammar learning grammars is generally described as

a) **model induction**: choosing the grammar $\mathcal{G}_i$ that is the most-likely generative model of the observed data $\mathcal{D}$,

b) **parameter estimation**: choosing the most probable parameters θ_i for that $\mathcal{G}_i$.

4 Conclusion

References

Chomsky, N. (1956), 'Three models for the description of language', *Institute of radio engineers transactions on information theory* **2**, 113–124.

Chomsky, N. & Miller, G. A. (1963), Introduction to the formal analysis of natural languages, *in* R. D. Luce, R. R. Bush & E. Galanter, eds, 'Handbook of mathematical psychology', Vol. 2, John Wiley and Sons, Inc., New York and London, pp. 269–321.

Church, A. (1936), 'An unsolvable problem of elementary number theory', *American Journal of Mathematics* **58**, 345–363.

Kleene, S. (1950), *Introduction to Metamathematics*, Van Nostrand, Princeton, NJ.

Post, E. (1936), 'Finite combinatory processes-formulation', *The Journal of Symbolic Logic* **1**, 103–105.

Savitch, W. J. (1987), Context-sensitive grammar and natural language syntax, *in* W. J. Savitch, E. Bach, W. Marsh & G. Safran-Naveh, eds, 'The Formal Complexity of Natural Language', Reidel, Dordrecht, pp. 358–368.

Turing, A. (1936), 'On computable numbers, with an application to the entscheidungsproblem', *Proceedings of the London Mathematical Society* **42**, 230–265.