Insights into the Problem of Syntactic Category Acquisition

Jason L. Hutchens

Abstract— **In this paper we briefly present an insight into the problem of syntactic category acquisition which suggests that techniques based on the clustering of distribution vectors are misguided, and that better results are attainable by searching for sub-simplices in the high-dimensional data which corresponds to the prediction vectors of simple, low-order stochastic language models.**

Keywords— **Syntactic category acquisition, stochastic language modelling, grammatical inference, pattern recognition.**

I. Introduction

THE *UpWrite* is a generalised process for the automatic construction of hierarchical representations of data, developed by Michael Alder as an alternative to King Sun Fu's programme of syntactic pattern recognition [1]. Application of the UpWrite to stochastic language modelling resulted in an insight into the problem of syntactic category acquisition [2]. In this paper we present that insight, following a brief overview of the traditional, distribution based approaches of syntactic category acquisition.

II. Syntactic Category Acquisition

Syntactic categories of symbols occur in many different kinds of data, including natural language text, and the automatic extraction of these categories is potentially of very great use. The traditional method of syntactic category acquisition from a corpus of data proceeds as follows:

1. Construct a *distribution vector* for each of the symbols which is being classified;
2. Cluster the space of distribution vectors using methods of numerical taxonomy [3] in order to determine the syntactic categories;
3. Assign each symbol in the alphabet to some number of these syntactic categories.

This distributional approach was common in linguistics prior to the Chomskian revolution, with Zellig Harris pioneering the field [4]. Computer programs which implement this technique generally define a symbol's environment as a local context consisting of the two symbols which precede it and the two symbols which follow it. Distribution vectors are formed by collecting statistics about the relative frequencies of contexts observed for each symbol in the alphabet. Syntactic categories are then created from symbols which occur in similar contexts, according to the results of clustering these distribution vectors.

The work of George Kiss [5], Hinrich Schütze [6] and Redington, Finch and Chater [7] use similar classification techniques, and suffer from similar drawbacks as a result, due to various ambiguities which exist in the data, and which aren't accounted for by the algorithm.

J. Hutchens is with Artificial Intelligence. E-mail: `hutch@a-i.com`

III. An Important Insight

Consider the phrase-structure grammar shown in figure 1, which generates a symbol sequence over a lowest level alphabet of four symbols. We would like to recover the three syntactic categories $(1) \mapsto [\mathtt{A}|\mathtt{B}]$, $(2) \mapsto [\mathtt{B}|\mathtt{C}]$ and $(3) \mapsto [\mathtt{C}|\mathtt{D}]$.

#	$\mapsto$	(1)(4)(2)(5)(3)(6)
1	$\mapsto$	A|B
2	$\mapsto$	B|C
3	$\mapsto$	C|D
4	$\mapsto$	(1)|(3)
5	$\mapsto$	(1)|(2)
6	$\mapsto$	(3)|(2)

Sample: `ACBCDBAABBCDABBACCAACBDBBBBBDCABCCDCB` ...

Fig. 1. A phrase-structure grammar which generates a sequence of symbols which exhibit several kinds of ambiguity.

The first thing to note is that these syntactic categories are ambiguous in that the symbols B and C belong to two syntactic categories; this is presumably a feature of many types of data, and ambiguity of this kind obviously occurs in natural languages. A second form of ambiguity is that any given syntactic category may be followed by exactly two others, with equal probability, due to the fact that broader syntactic categories, such as (5), exist in the grammar. For example, $P((1)|(2)) = \frac{1}{2}$ and $P((2)|(2)) = \frac{1}{2}$. This second type of ambiguity also exists in natural languages—consider, for example, the fact that an adjective may be followed by either an adjective or a noun in a sequence of English words.

This second form of ambiguity, *existing at the level of the syntactic categories*, means that the observations which we actually make at the lowest level representation of the data (which is the level at which we must necessarily begin) will consist of a convex combination of some unknown number of syntactic categories.[1] For example, $P(\mathtt{B}|(2)) = \frac{1}{2}P(\mathtt{B} \in (1)) + \frac{1}{2}P(\mathtt{B} \in (2))$.

Let us continue this exposition by considering the nature of the space of probability distributions. Each element of a probability distribution is non-negative, and the sum of all elements is unity. Therefore, a probability distribution represented as a vector in $\mathbb{R}^{|\mathcal{A}|}$, where $|\mathcal{A}|$ is the cardinality of the alphabet, will lie on the $(|\mathcal{A}|-1)$–*simplex*.

[1] Here we are assuming that observations take the form of the predictions, expressed as probability distributions over the alphabet, of some stochastic language model, such as a low-order Markov model.

Consider the predictions made by a stochastic language model about a string of symbols taken from a binary alphabet. If we consider the predictions to be vectors in $\mathbb{R}^2$, we will find that all of the vectors lie on the the simplex corresponding to the line segment which has $(0,1)$ and $(1,0)$ as its end-points, as shown in figure 2. Similarly, for a ternary alphabet, the predictions made by a stochastic language model will lie in the region of space corresponding to the equilateral triangle which has $(0,0,1)$, $(0,1,0)$ and $(0,0,1)$ as its vertices, as in figure 3. In general, over the alphabet $\mathcal{A}$, all predictions lie in the region of space corresponding to the $(|\mathcal{A}|-1)$–simplex, in $\mathbb{R}^{|\mathcal{A}|}$.

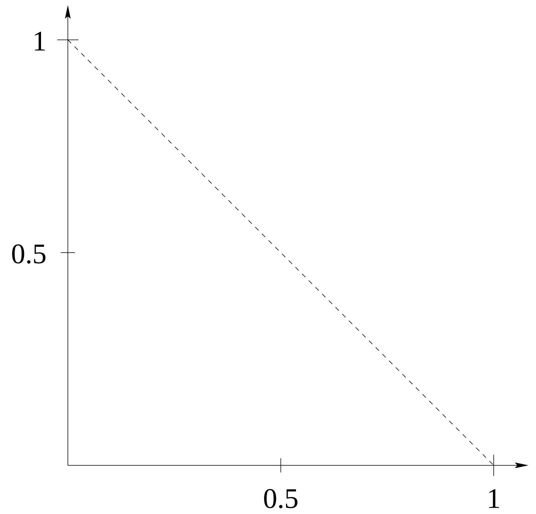

Fig. 2. A simplex in $\mathbb{R}^2$ is the line segment connecting $(0,1)$ and $(1,0)$.

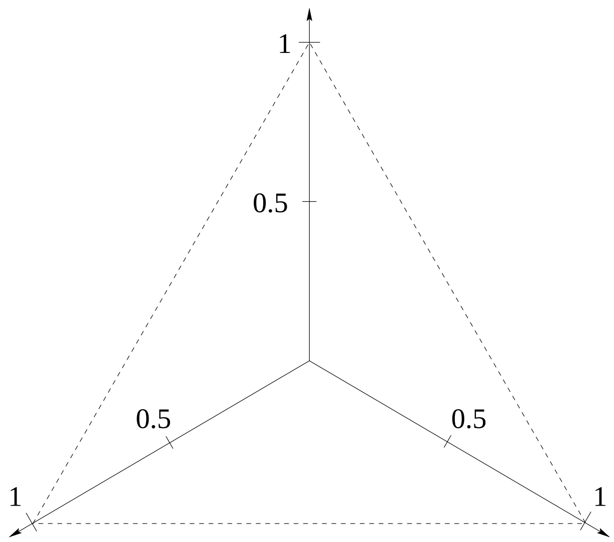

Fig. 3. A simplex in $\mathbb{R}^3$ is the region of space corresponding to the equilateral triangle which has $(0,0,1)$, $(0,1,0)$ and $(1,0,0)$ as its vertices.

We are now in a position to consider how the predictions made by a stochastic language model inferred from the lowest-level representation of the data may be distributed within the space. We do this by recalling that these observed probability distributions will be convex combinations of some unknown number of probability distributions at a higher level of abstraction. This means that each observed probability distribution will lie somewhere within the simplex which has as its vertices the k higher-level probability distributions which the lower level probability distribution is a convex combination of. The exact location of the observed probability distribution within this simplex is determined by the coefficients of the convex combination, and these are equivalent to a probability distribution over the higher-level alphabet.

Another way of thinking about this is that the observed probability distributions are the *DownWritten* versions of higher-level probability distributions. For example, the higher-level probability $P(\mathtt{B}|(2))$ is DownWritten to the lower-level probability $\frac{1}{2}P(\mathtt{B} \in (1)) + \frac{1}{2}P(\mathtt{B} \in (2)) = \frac{1}{2}$, as we have previously shown. We assume, of course, that each syntactic category corresponds to a probability distribution over the lower-level alphabet, and this is true in the case where the syntactic categories are stochastic.

It is therefore apparent that the observed probability distributions are not noisy versions of a single underlying syntactic category—as methods based on clustering the observations assume—they are noisy convex combinations of some unknown number of syntactic categories. The evidence we observe is therefore rather indirect; instead of looking for clusters of points, we should be looking for the set of points which define the vertices of some unknown number of k–simplices, for various k. Obviously there are many candidate point sets, with the unit vectors of the $(|\mathcal{A}|-1)$–simplex being a possible (and uninteresting) solution. Therefore, we should search for a solution according to the principle of Occam's razor; we want to find the vertices which describe the observed distribution of vectors well, while at the same time minimizing the combined description length of these vertices and the data relative to the k-simplices which they define.[2]

As an example of the sort of data we have been talking about, we show in figure 4 a plot of the vectors which correspond to the predictions made by a 2^{nd}–order Markov model over the data defined by the grammar of figure 1.[3] In this diagram, the four vertices of the $(|\mathcal{A}|-1)$–simplex are denoted by crosses, and connected with solid lines. These vertices define a tetrahedron in $\mathbb{R}^4$, and all possible probability distributions fall within this simplex. The probability distributions corresponding to the three syntactic categories of the grammar are denoted by circles, and it is obvious that the sixteen vectors[4] which represent the predictions made by the 2^{nd}–order Markov model all fall on

[2]This should be possible, because knowledge of the k-simplices reduces the degrees of freedom of the observed data.

[3]This plot is a projection from $\mathbb{R}^4$ onto $\mathbb{R}^2$, as produced by the `xgobi` program.

[4]There are sixteen distinct vectors corresponding to the predictions made after the sixteen possible contexts observed by the 2^{nd}–order Markov model. Note that two of these vectors lie very near the syntactic categories [B|C] and [A|B], and are therefore difficult to distinguish in the diagram.

the simplex defined by these three vectors. An algorithm which searched for these vertices in the manner prescribed above would successfully recover the three syntactic categories, while any standard clustering algorithm would not.[5]

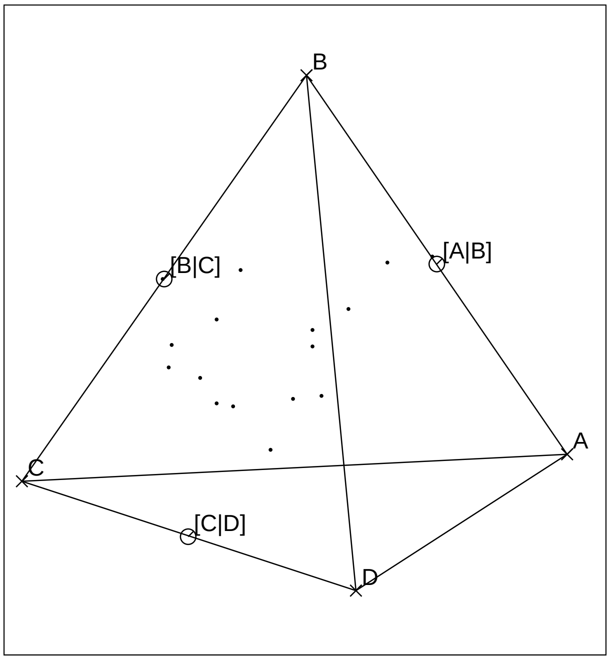

Fig. 4. The sixteen vectors representing predictions made by a 2^{nd}–order Markov model over data generated by the grammar of figure 1 lie on the 2–simplex which is defined by the three vectors which represent the syntactic category, and this, in turn, lies within the 3–simplex in $\mathbb{R}^4$.

Increasing the context available to the stochastic language model will serve to disambiguate the predictions it makes to an extent, and syntactic categories will therefore be more readily obtained by clustering the predictions made by higher order Markov models. This has the disadvantage, however, of limiting the number of observations made per context, and the sparse data problem raises its ugly head. A more intelligent technique for finding syntactic categories is required, and we hope that the insight presented in this paper marks the genesis of such a technique.

IV. Summary and Conclusion

In this paper we have shown that the predictions made by stochastic language models are convex combinations of stochastic syntactic categories which exist, undiscovered, in the data the model was inferred from. Traditional distribution based approaches for finding syntactic categories fail to take account of this fact. Looking for sub-simplices of distribution vectors may prove to be more successful than merely finding clusters of them.

[5]The problem is compounded by the fact that the bottom-most syntactic category in the diagram is separated from the vectors which correspond to predictions by a significant amount.

References

[1] Michael D. Alder, "Inference of syntax for point sets," in *Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies and Hybrid Systems*, E. S. Gelsema and L. N. Kanal, Eds. June 1994, number 16 in Machine Intelligence and Pattern Recognition, pp. 45–58, Elsevier Science B.V.

[2] Jason L. Hutchens, *The UpWrite Predictor: A General Grammatical Inference Engine for Symbolic Time Series with Applications in Natural Language Acquisition and Data Compression*, Ph.D. thesis, Department of Electrical & Electronic Engineering, The University of Western Australia, Australia 6907, 1999.

[3] Peter H.A. Sneath and Robert R. Sokal, *Numerical Taxonomy*, W.H. Freeman and Company, 1973.

[4] Zellig S. Harris, "Distributional structure," in *Papers on Syntax*, Henry Hiz, Ed., pp. 3–22. Kluwer Boston, 1981.

[5] G.R. Kiss, "Grammatical word classes: A learning process and its simulation," *Psychology of Learning and Motivation*, vol. 7, pp. 1–41, 1973.

[6] Hinrich Schütze, "Part-of-speech induction from scratch," in *Proceedings of the 31^{st} Annual Meeting of the Association for Computational Linguistics*, 1993, pp. 251–258.

[7] Martin Redington, Nick Chater, and Steven Finch, "Distributional information: A powerful cue for acquiring syntactic categories," *Cognitive Science*, vol. 22, no. 4, pp. 425–469, 1998.