
Linguistic Categories as Basins of Curvature

Whitney Tabor
Mind Articulation Project
MIT Building 20C-228
Cambridge, MA 02139
tabor@mit.edu

DRAFT: Please do not quote.

Abstract

Linguistic grammars do a good job of elucidating the lexical categories and phrasal units of the languages they model. Certain recurrent Connectionist networks can model similar data but it is not easy to discern the abstract structures of the resulting representations. Hierarchical clustering is helpful in this regard, but it often produces implausible clusters and there seems to be no principled way of deciding which clusters are relevant. A more promising approach is to examine the curvature of the hidden space image in the output space. Because the relative frequency distributions of lexical items tend to be highly categorical in nature, with probabilistic variation in only a few dimensions, they generally lie on the linear surfaces of the output space. Consequently, categories are associated with regions of low curvature. I show that category-extraction based on this principle can produce a more revealing analysis than hierarchical clustering in certain cases.

1. Introduction.

Linguistic grammars express high-level structural characteristics of the languages they model in a particularly transparent manner. But Connectionist models of the same phenomena are often difficult to interpret.

A good example is Elman 1990 and 1991's experiments with training a Simple Recurrent Network to predict English-like word sequences. Elman found that a network trained on the task of predicting next-words generated by a simple context-free grammar learned to approximate the word-to-word transition probabilities associated with the grammar. In this sense, it learned to encode much of the syntactic structure of the grammar. As a way of gaining insight into higher-level properties of the network's representation, Elman performed a hierarchical clustering analysis on a sample of the hidden unit states. Happily he found that there were clusters corresponding to many of the categorical structures that linguists find explanatory (e.g., nouns, animate nouns, verbs, transitive-verbs, intransitive verbs). But unfortunately, not all the clusters mapped nicely onto interpretable concepts. Moreover, there seems to be no theoretical method available of deciding which clusters reflect fundamental features of the prediction architecture and which are spurious creations of the exhaustive clustering algorithm.

Other studies have also yielded partially coherent results. Servan-Schreiber et al. 1991 train a recurrent network on a next-symbol prediction task where the symbols are generated by a probabilistic finite state grammar. They find that hierarchical clustering provides direct identification of the generating grammar states only special cases. Giles et al. 1992 also train a recurrent network on the output of a finite state grammar. They sometimes succeed in reconstructing the states of the grammar by quantizing the activation states of the hidden units and forming classes of elements that associated with the same hidden space cubicles.

Although these clustering and discretization techniques can be revealing, they focus only indirectly on the structure of the network representation by studying its interaction with the sometimes noisy, and usually incomplete data. Here, I describe an alternative technique which permits direct characterization of critical features of the network structure. This technique focuses on the curvature of the hidden image in the output space.

When output activations model probabilities, they lie in the interval $[0,1]$ so the output space is a bounded simplex with linear sides. Typically, linguistic elements are categorically associated with a small number of behaviors, and show probabilistic alternation among these behaviors.¹ Consequently, if the task is to predict the behaviors associated with each element, then the target probability distributions lie on surfaces and edges of the output simplex, which themselves lie in a variety of orientations. Moreover, each linguistic class tends to consist of many members whose behaviors are categorically similar or identical, but which show essentially random differences in the associated probabilities.² Thus, the categories tend to correspond to regions on the linear surfaces of the simplex. Consider a network with a relatively low-dimensional hidden unit representation that is trained to map to these outputs. The image of the hidden space in the output space is a smooth connected manifold. In order to fit the data, the network must flex this manifold

¹For example, the verb *report* has a Noun Phrase (NP) complement (e.g., *reported the scores*) approximately 64% of the time and a Sentential (Sbar) Complement (e.g., *reported that the geese were returning*) approximately 25% of the time (based on the Penn Treebank *Wall Street Journal* sample).

²For example, (NP, Sbar) rates for *mention* are (52%, 24%), for *know*, (29%, 48%), for *indicate* (22%, 67%).

so that it passes through the differently oriented linear regions associated with the categories. Consequently, the fitted hidden manifold will tend to be linear in regions associated with categories and curved in intermediate regions. This fact may be used to reconstruct the categories from the trained network state. As a step toward developing this technique for use on real linguistic corpora, I examine its performance on artificially generated corpora where it is easy to perceive the high-level structure of the generating model.

2. Motivation from historical work.

This approach to analyzing network grammar models is motivated by work on the historical evolution of linguistic categories. *Grammaticalization* (see Hopper and Traugott 1993) is a common kind of language change in which a member of one of the large, meaning-laden open classes (e.g., nouns, verbs) evolves into a member of one of the small, semantically abstract grammatical markers (e.g., prepositions, auxiliary verbs, tense/aspect/negation markers, conjunctions). Examples include the French negation marker *pas* which derives from a noun meaning ‘step’, the German preposition *wegen* ‘on account of’ which derives from a noun meaning ‘way, path’, the English future auxiliary marker *going to/gonna* (e.g. *It’s gonna be sunny tomorrow.*) which derives from the motion verb *going to* (e.g. *We are going to Alabama.*).

Processes of grammaticalization typically take place gradually, with subtle changes in the frequencies of the various uses of evolving forms preceding more dramatic changes in their categorical behavior (Tabor 1994). In a historical text based study of the *be going to* development, I find that the trajectory of this item in probability space is well-modeled as a trajectory along the hidden unit manifold of a network trained to model the general characteristics of the linguistic categories involved (Tabor 1995). During the transition phase (17th through 19th centuries), *be going to* behaves as though it is traversing one of the curved, intermediate regions of the manifold.

Given the fact that standard grammar models interpolate only linearly, and that it is the nonlinear structure of the network model that generates the successful predictions, it is desirable to find a way of characterizing the structure of these nonlinearities. In particular, it is desirable to be able to say, in terms of the structure of the network representation, when an element has changed enough quantitatively that it should start exhibiting novel categorical behaviors.

3. A simple feedforward example.

I first describe a simple feedforward example. Consider the probabilistic mapping shown in Table 1. Intuitively, the inputs fall into two classes: those for which the output is primarily A, and those for which the output is primarily B. However, the water is somewhat muddied by the fact that there are a number of shared behaviors among the inputs.

The mapping of Table 1 is intended as an idealized picture of a typical linguistic situation. For example, one can think of the inputs as verbs and the outputs as

Table 1: A mapping that generates two classes of elements. (The inputs are listed in the left-hand column, the outputs across the top. Each numerical entry gives the probability that the corresponding input will map to the corresponding output.)

	A	X	Y	B
A0	1.00	0.00	0.00	0.00
AX	0.75	0.25	0.00	0.00
AY	0.75	0.00	0.25	0.00
All	0.50	0.20	0.20	0.10
B0	0.00	0.00	0.00	1.00
BX	0.00	0.25	0.00	0.75
BY	0.00	0.00	0.25	0.75
Ball	0.10	0.20	0.20	0.50

types of complements that cooccur with the verbs. The A class could correspond to verbs that take primarily Adjective Phrase complements (e.g. *grew uneasy, waxed poetic, felt mischievous*), but sometimes take Noun Phrase, Prepositional Phrase, or Sentential complements. The B behavior could correspond to verbs that take primarily Sentence Complements (e.g. *remarked that she knew all along, said that he would be happy to apologize*), but sometimes take one of the other three types.

I trained a feedforward network using data generated by the mapping of Table 1. Each input was uniquely coded as an indexical bit vector (a vector with a value of 1 on some dimension and 0 everywhere else). The network had two hidden units with fixed sigmoid activation functions, and four output units with softmax activation. Since the outputs were distributed multinomially, training by the delta rule (with backpropagation of error) caused the output units to converge on the expectations of the outputs given the inputs (Rumelhart *et al.* 1995). The mapping is very simple in this case and the network had no trouble learning it on repeated trials.

I used a numerical method to estimate curvature of the hidden manifold. The manifold is defined by the function,

$$(1) \quad f(\vec{x}) = g(W\vec{x} + \vec{b})$$

where \vec{x} is the vector of hidden unit activations, W is the matrix of hidden \rightarrow output weights, \vec{b} is the vector of output biases and g is the softmax activation function. The best affine approximation to the hidden manifold at \vec{x}_0 is given by

$$(2) \quad A(\vec{x}) = f(\vec{x}_0) + f'(\vec{x}_0)(\vec{x} - \vec{x}_0)$$

where $f'(\vec{x}_0)$ is the Jacobian of f at \vec{x}_0 . An easy way of estimating the relative magnitude of the curvature at \vec{x}_0 is to examine the maximum distance of this affine surface from the manifold at some small distance from \vec{x}_0 . Thus we can define,

(3)

$$\begin{aligned}
C(\vec{x}_0, \epsilon) &= \max_{|\vec{\epsilon}|=\epsilon} |A(\vec{x}_0 + \vec{\epsilon}) - f(\vec{x}_0 + \vec{\epsilon})| \\
&= \max_{|\vec{\epsilon}|=\epsilon} |f(\vec{x}_0) - f(\vec{x}_0 + \vec{\epsilon}) + f'(\vec{x}_0) \vec{\epsilon}|
\end{aligned}$$

I estimated C by examining evenly spaced points on the ball of radius ϵ around \vec{x}_0 in the hidden unit space.

Figure 1: Contour plot of hidden manifold curvature values. The diagonal, lobed ridge separates the basin corresponding to class A from the basin corresponding to class B. [Horizontal axis = Hidden Unit 1, Vertical axis = Hidden Unit 2]

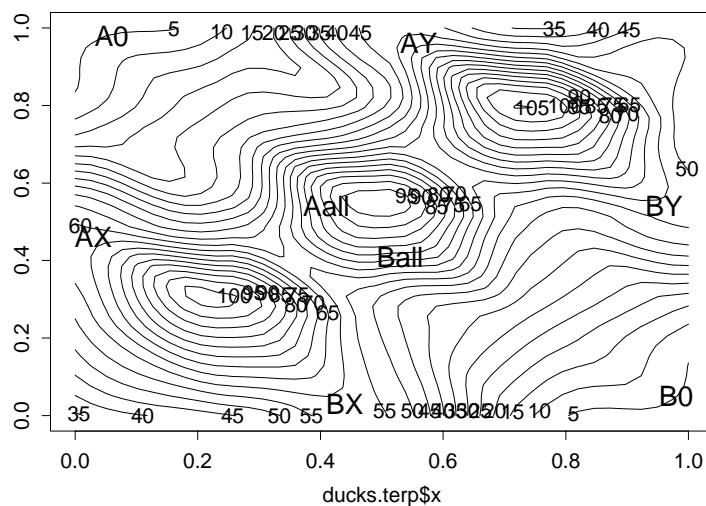
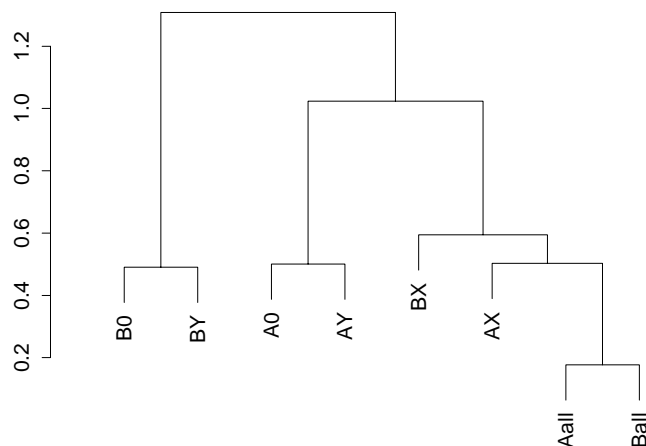


Figure 1 shows a contour plot of these curvature estimates for the trained feedforward network described above. The letter symbols on the figure show the hidden unit locations corresponding to the different inputs. As hoped, the region is divided into two basins corresponding to the two categories. There is a ridge of high curvature running along the middle of the space which may be interpreted as the general-case dividing line between the categories.

In this case, a hierarchical clustering analysis of the hidden unit states (Figure 2) produces undesirable results: there is no cluster corresponding to either category A or B.

This section has illustrated how basins of curvature correspond to intuitively sensible divisions in a very simple example. In the next section, I examine the performance of the procedure with a recurrent network trained on a more linguistically realistic task.

Figure 2: Hierarchical clustering analysis with undesirable results.



4. A recurrent network example.

I used the probabilistic finite state grammar in Figure 3 to generate a stream of words, which were assigned localist input representations and presented in order to the network. As in Elman 1990 and 1991, the task was to predict on the output layer what word was coming next at each point. The network architecture was the same as in the preceding simulation except that there were 16 input and output units (one for each word) and recurrent connections in the hidden layer. The recurrent connections are necessary in order to learn the context dependency associated with the adverbs. I did not train on the complete gradient but unfolded three steps in time (see Rumelhart, Hinton and Williams 1986).

The net learned the task easily on many successive runs. There are three major states: the states of being most likely to generate a **Subject** noun, a **Verb**, and an **Object** noun, respectively. Each of these is complicated somewhat by the possibility of intervening adverbs and the possibility of using a verb intransitively. There are two minor states associated with the adverbs, which cannot occur in sequence, but otherwise generate close approximations of the three major states.

Figure 4 maps the curvature of the hidden manifold. Note that there are three basins which correspond to the three major states. The words **Subject**, **Verb** and **Object** show the average locations of elements which generated the three corresponding states. In each case, although the particular hidden states are scattered, they all lie within the appropriate basin. In this simulation, it is evident by inspection that a hierarchical clustering of the hidden units yields the three major categories in the first two bifurcations, but again, there seems to be no principled reason for

Figure 3: A grammar for generating sentences with optional objects and scrambling adverbs.

1.00 S : Nsubj VP'
 0.60 VP' : VP
 0.20 VP' : VP Adv
 0.20 VP' : Adv VP

 0.80 VP : V Nobj
 0.20 VP : V

 0.25 Nsubj : Lorrie, Rudolph, Jasper, Florence

 0.25 V : eats, knows, chooses, wins

 0.25 Nobj : cactus, fish, houses, despair

 0.25 Adv : slowly, thoughtfully, vigorously, quietly

stopping at three. The basin analysis, by contrast, yields only three groupings. In effect, the network has assigned lower priority to distinctions beyond this grouping. In this sense, the basin analysis shows that the network is making a categorical contrast between systematicity in the signal which is treated as significant (the major grammatical states) and systematicity in the signal that is treated as noise (the minor adverb states) (cf. MacKay 1992).

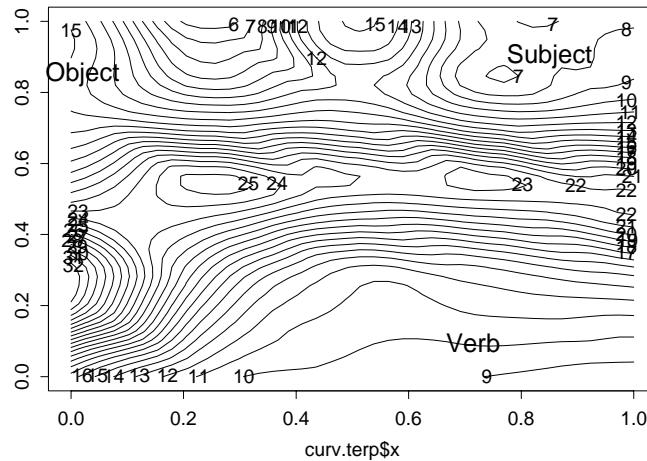
5. Conclusion

I noted that a network trained to predict the highly restricted probability distributions of linguistic items develops a representation in which basins of curvature correspond to categories. This result is useful since previous structure-extracting mechanisms like hierarchical clustering are not explicit about which clusters correspond to representationally important categories. The problem with such techniques is that they do not provide a clear theory of the distinction between contrasts which the network treats as fundamental to the prediction task and contrasts which are being sidelined for their lesser predictive value. Developing a theory of this difference along the lines suggested here may facilitate interpretation of Connectionist models at the level of the abstract, conceptually unifying structures which seem to play a central role in natural language representation.

Acknowledgements

This research has been supported in part by postdoctoral fellowship funding to the Center for the Sciences of Language (NIH-NIDCD 5T32DC0035-04).

Figure 4: The three curvature basins in the hidden unit manifold correspond to the three primary states of the generating finite state grammar.



References

- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14:179–211.
- Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7:195–225.
- Giles, C. L., Miller, C. B., Chen, D., Chen, H. H., Sun, G. Z., and Lee, Y. C. (1992). Learning and extracting finite state automata with second-order recurrent neural networks. *Neural Computation*, 4:393–405.
- Hopper, P. J. and Traugott, E. C. (1993). *Grammaticalization*. Cambridge University Press, Cambridge, England.
- MacKay, D. J. C. (1992). Bayesian methods for adaptive models. Ph.D. Dissertation, California Institute of Technology.
- Rumelhart, D., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation: The basic theory. In *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum Associates.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors, *Parallel Distributed Processing, Volume I*, pages 318–362. MIT Press.
- Servan-Schreiber, D., Cleeremans, A., and McClelland, J. L. (1991). Graded state machines: The representation of temporal contingencies in simple recurrent networks. *Machine Learning*, 7:161–193.
- Tabor, W. (1994). Syntactic innovation: A connectionist model. Ph.D. dissertation, Stanford University.

- Tabor, W. (1995). Lexical change as nonlinear interpolation. In Moore, J. D. and Lehman, J. F., editors, *Proceedings of the 17th Annual Cognitive Science Conference*. Lawrence Erlbaum Associates.