A Continuum of Induction Methods for Learning Probability Distributions with Generalization

Dekai Wu*
Computer Science Division
University of California at Berkeley
Berkeley, CA 94720 U.S.A.
dekai@cs.berkeley.edu

Abstract

Probabilistic models of pattern completion have several advantages, namely, ability to handle arbitrary conceptual representations including compositional structures, and explicitness of distributional assumptions. However, a gap in the theory of induction of priors has hindered probabilistic modeling of cognitive generalization biases. We propose a family of methods parameterized along a value γ that controls the degree to which the probability distribution being induced generalizes from the training set. The extremes of the γ-continuum correspond to relative frequency methods and extreme maximum entropy methods. The methods apply to a wide range of pattern representations including simple feature vectors as well as frame-like feature DAGs.

Introduction

The motivations for this work arise from the shortcomings of existing theoretical frameworks in two fields:

- Many neural network pattern completion models have the desirable characteristic of being inherently biased to generalize from training data. Two drawbacks, however, are: (a) frequently there are no clearly specified desiderata on the nature of statistical distributions to be learned by a neural net, and (b) we are currently unable to efficiently represent compositional structures as feature vectors.[1]

- Probabilistic and statistical inductive models, being symbolic, can easily handle compositional structures. However, there is a lack of models that can be biased to generalize from training data; specifically, the most common methods for inducing prior probability distributions—relative frequency priors and maximum entropy priors—are inadequate.

To address this gap we propose a family of inductive methods, called the γ-continuum, that can be thought of in several ways:

- From the associationist point of view, a functional specification for a class of pattern completion models.

- From the learning point of view, a non-Bayesian inductive learning method for a probabilistic inference engine. An inductive bias is determined by a set of *abstractive relations*.

- From the probability theory point of view, a method for generating priors from a training set. The method incorporates an a priori *abstractive bias* that causes the model to make generalizations.

Our driving application is probabilistic pattern completion to support integrated natural language parsing and semantic interpretation, where the patterns combine lexical, syntactic, and semantic structures. In a probabilistic pattern completion model, the input is an abstract or partial pattern, and the task is to select the most probable complete pattern.

The proposed methods are more flexible than neural networks with respect to representation constraints; concepts need not be represented as feature vectors, but only need to satisfy a weaker semi-lattice constraint, explained below. Limited forms of compositional conceptual structure are permitted. At the same time, the nature of the probability distributions that can be learned is clearly formulated, and these distributions are better for modeling generalization from a training set than either relative frequency or maximum entropy priors.

Our proposal fills a gap in the existing theory of probability distribution induction. However, it is not intended that the distributions generated by our methods necessarily be evaluable by computationally tractable means. The methods are information-theoretic functional specifications, for which different approximation heuristics may be appropriate depending upon the domain.

Pattern Structure and the Abstraction Space

We shall only consider examples where the patterns (instances) are encoded using feature-vector and feature-DAG (frame-like) representations, though the internal structure of patterns is of no consequence to the inductive methods and many other representations could be used as well. The set

*This paper has benefitted greatly from helpful discussions with Terry Regier and Steve Omohundro, and I am grateful to Marti Hearst for implementing code to generate the induced distributions. Thanks to Nigel Ward for proofreading, and also to Robert Wilensky, Jerome Feldman, and the members of the BAIR and L_0 seminars. This research was sponsored in part by the Defense Advanced Research Projects Agency (DoD), monitored by the Space and Naval Warfare Systems Command under N00039-88-C-0292, the Office of Naval Research under contract N00014-89-J-3205, and the Sloan Foundation under grant 86-10-3.

[1] A number of recent proposals employ recurrent nets to achieve "dynamic compositionality" that can sequentially "expand out" compositional structures (e.g., Pollack 1989, 1990), but there is little consensus as to the limits of such approaches.

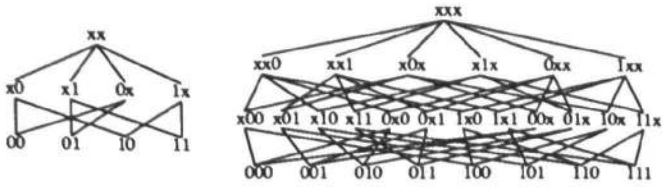

Figure 1: Feature abstraction semi-lattices.

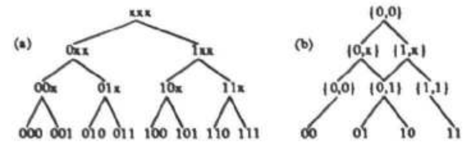

Figure 2: Semi-lattices for (a) incomplete feature abstraction, (b) position-insensitive feature abstraction.

of possible patterns must be finite (though arbitrarily large) and forms the space of simple events. To perform pattern completion using probabilistic inference, we need to know the probability distribution over these events.

Compound events constitute abstractions over groups of patterns. Note that all patterns are defined to be simple events; compound events or abstractions are not proper patterns, but partial or incomplete patterns. This usage should not be confused with the conventional AI use of the notion of abstraction as an epistemological relationship between two concepts.

The shape of the abstraction hierarchy is determined by a set of abstractive relations. Figure 1 shows, for 2- and 3-bit vector patterns, the hierarchy determined by a feature-abstractive relation that substitutes "x" or "don't care" bits for feature values. The leaf nodes are complete patterns (simple events); the internal nodes are incomplete patterns (compound events).

In a pattern completion task, the input is an internal node representing an abstract or partial pattern. The task of completing the pattern corresponds to selecting the most probable leaf node (complete pattern) under the internal node. (The term "simple event" is somewhat counterintuitive when speaking of complete patterns, which are more fleshed out than incomplete ones.)

Minimal constraints are imposed on the shape of the abstraction space. In fact, the only constraint is that the abstractive relations must determine a *semi-lattice* hierarchy, meaning that for any two concepts there must be a unique least upper bound (most specific common ancestor). Figure 2 shows other useful examples of abstractive relations.

The sorts of patterns that motivated development of the γ-continuum are more complex than feature-vectors. These patterns, which derive from semantic network and predicate logic languages, can be represented as *feature-DAGs*, and allow compositional structures and variable unification. Figure 3 shows two (complete) patterns that demonstrate how feature-DAG representations can be used (the details are unimportant here). An example of a feature-DAG representing plan decomposition is shown in (a). For our parsing and interpretation application, a sample feature-DAG for the nominal compound *weekend guest* is shown in (b). Details on how we map unification-grammar structures into feature-DAGs (for a simpler probability model) are given in an earlier paper (Wu 1990).

(a) (b)

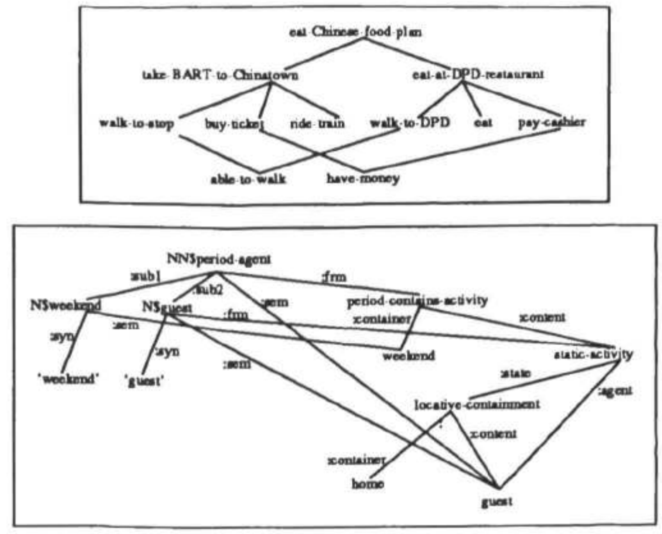

Figure 3: Feature DAGs (see text).

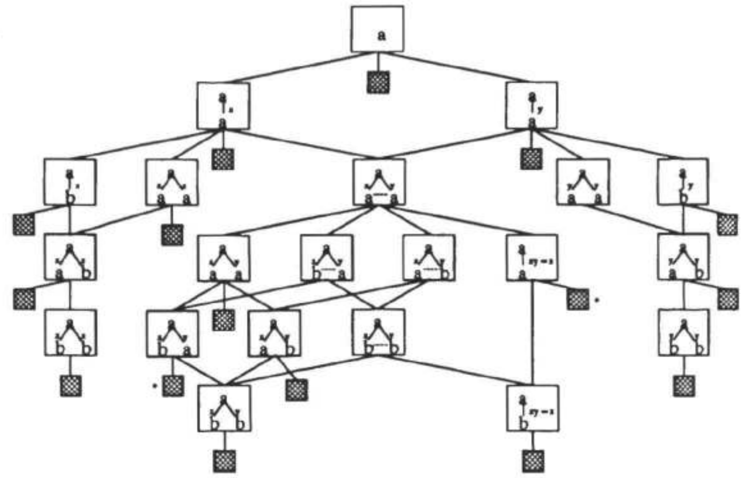

Figure 4: Semi-lattice for a feature-DAG pattern space. The leaf nodes (shaded) are the simple events. Here each feature-DAG is restricted to depth 2 and branch factor 2; there are only two concepts a and b, and a is superordinate to b; and there are two primitive roles x and y that combine to form a third composite role z.

Given these sorts of feature-DAG patterns, Figure 4 shows the semi-lattice determined by the abstractive relations we use for our parsing and semantic interpretation model, making some simplifying restrictions. The abstractive relations determine how to generate all the ancestors of any pattern. The four relations used here are: superordinate concept substitution (an ancestor can be generated by replacing any concept with a superordinate concept), sub-DAG partition (an ancestor can be generated by extracting any partition that is itself a DAG), concept unification option (an ancestor can be generated by adding an option to unify any two compatible concepts), and role de-unification option (an ancestor can be generated by adding an option not to unify the primitive roles comprising a composite role). These abstractive relations make it possible to represent such things as the conditional probability of two roles or fillers being unified, something feature-vector models have difficulty with.

Inductive Methods for Probabilistic Models

The problem of inducing probability distributions from a finite sample, or training set, has a long history. In the past, much debate about the validity of various proposals has arisen from different interpretations of probability theory (see Weatherford 1982; Hacking 1975; Mortimer 1988; Cheeseman 1985). Probability theory, as a mathematical framework, can legitimately be appropriated for different purposes, so long as the interpretation is made clear. Three of the main schools are subjective probabilities, which denote degrees of belief; relative frequency probabilities, which denote real-world physical properties; and logical probabilities, which are purely logical relations. The acquisition of priors is a problem that plagues all probabilistic inference mechanisms, including the widely used Bayesian networks (Pearl 1988; natural language interpretation is done by Goldman & Charniak 1990), but little if any work has attempted to interpret the priors as a model of the *a priori* cognitive biases that give rise to generalization tendencies.

Assume we have a probabilistic pattern completion engine. What should its priors be, i.e., what is a legitimate source of initial probabilities? Any set of priors incorporates biases; there is no such thing as absolutely uninformative priors. (Equiprobability among all events is deceptive: splitting any one event into two causes all other probabilities to be revised for no logically justifiable reason.)

However, the fact that priors incorporate a bias is a plus rather than a negative, given that our purpose is to model human inductive generalization. The important thing is just to match the model's bias as closely to a human's as possible.[2] This puts our use of probability in the subjectivist school, related to logical probabilities but outside the relative frequentist school. We now examine why two of the most commonly used methods for establishing priors are not suitable for our purpose.

There are two main problems with setting the prior distribution equal to the relative frequency distribution in the training set. The first problem holds for both subjective and logical probability models: any event not in the training set is assigned zero probability. For example, in our nominal compound interpretation domain, many nominal compounds like *weekend guest* are novel constructions one would not necessarily expect in a training set. Nonetheless, a nonzero probability should be assigned to the best interpretation. The major contribution of Carnap's (1952, 1962) classic work on logical probability is a solution to the zero-probability problem, called the λ-continuum of inductive methods. This is a family of methods for inducing a prior distribution from a sample (training set), parameterized by λ. If $\lambda = 0$ the priors are exactly the relative frequencies, but if $\lambda > 0$ there are no nonzero prior probabilities. At $\lambda = \infty$ equiprobable priors are assigned to all simple events, and there is no sensitivity to the sample.

The second problem holds for subjective probability models: using the relative frequency distribution from the training set permits no generalization. Yet human learners generalize. Neural network research has demonstrated, for a number of different neural models, plausible ways in which generalization biases can be inherent, e.g., restricting hidden layer sizes. This is actually a stronger version of the first problem; the reason we do not want zero probabilities assigned to novel events is that some generalization ought to occur and thus give nonzero probabilities to novel events. None of Carnap's methods perform generalization: an event in the training set never raises the probability for other similar events. The probability for the best interpretation of *weekend guest* should not only be nonzero, but in fact should be greater than that of any other interpretation, because of its similarity to other events that *are* in the training set such as, say, *holiday visitor*.

Of the methods for inducing priors that allow generalization, maximum entropy has been the most popular method (Cheeseman 1987; Jaynes 1979). Given some set of probabilities for compound events (joint probabilities), the probabilities for simple events are computed by choosing the distribution that maximizes an entropy measure

$$H = -\sum_{i=1}^{C} P_i log P_i$$

while still satisfying the given joint probability constraints. In other words, what maximum entropy does is fill in probabilities to complete the joint distribution, given constraints on the values for some of the probabilities. There are information-theoretic arguments that this method minimizes the amount of information assumed. Maximum entropy methods do not specify whether joint probabilities are relative frequencies, but this is usually assumed.

The problem with using maximum entropy methods for generalization is that they do not specify how to choose which compound events to assign probabilities to. Training sets contain simple events, not compound events. The relative frequency distribution for the simple events fully determines the joint distribution for compound events—there is no room for making generalizations. In order to get generalizations, some of the simple events' probabilities must be discarded (as well as some of the compound events' probabilities, for even more generalization) and then recomputed by maximum entropy. (This is generalization because, for example, if for some compound event, maximum entropy replaces all its simple events' probabilities with equal probabilities, and the simple events originally had different training set frequencies, effectively a single generalization about all the simple events comprising the compound event is made.) Maximum entropy does not specify which probabilities to discard, and depending on this, different generalizations will be made. In the extreme, if all relative frequencies are discarded, maximum entropy makes all simple events equiprobable; this is extreme over-generalization because it generalizes every training event to all other simple events.

What we propose is a continuum of methods that vary according to a parameter γ that controls how much generalization occurs. The extreme ends of the continuum turn out

[2] It is beyond our present scope to offer methodological guidelines for matching biases.

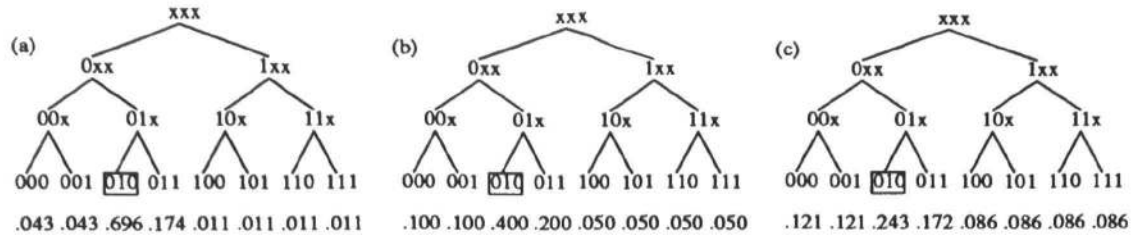

Figure 5: Effect of a training instance 010 for (a) $\gamma = 0.5$, (b) $\gamma = 1$, and (c) $\gamma = 2$.

to be the same as Carnap's λ-continuum. However, where λ dictates the degree of sensitivity to the training set, γ dictates the degree of generalization from the training set. At $\gamma = 0$ no generalization is done and the priors are the relative frequencies, and at $\gamma = \infty$ we get the maximum entropy over-generalization extreme.

The γ-continuum of Methods

Denote the set of concepts or simple events by $Q = \{q_1, q_2, \ldots, q_C\}$, and let X be a random variable with values ranging over Q. Given a training vector $\mathbf{T} = (t_1, t_2, \ldots, t_N)$ where $t_i \in Q$,

$$P_i \stackrel{\text{def}}{=} Pr(X = q_i) \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} norm\left[2^{-d(q_i, t_n)/\gamma}\right]$$

where *norm* means normalization to 1 as follows:

$$P_i \stackrel{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} \frac{2^{-d(q_i, t_n)/\gamma}}{\left[\sum_{k=1}^{C} 2^{-d(q_k, t_n)/\gamma}\right]}$$

and $d(q_a, q_b)$ is the *logical distance* between two simple events. The logical distance derives from the bias given by the abstractive relations. It is defined in terms of the *logical class cardinality*

$$lcc(q_a, q_b) \stackrel{\text{def}}{=} |leaves(lub(q_a, q_b))|.$$

The logical distance is then

$$d(q_a, q_b) \stackrel{\text{def}}{=} log_2(lcc(q_a, q_b)) = log_2|leaves(lub(q_a, q_b))|.$$

For example, consider the simple binary-tree space from Figure 2(a). The least upper bound $lub(000, 001)$ is 00x, which has only the two leaves 000 and 001. Thus the logical class cardinality $lcc(000, 001) = 2$, and the logical distance $d(000, 001) = 1$. Similarly, the logical distance $d(000, 100) = log_2 8 = 3$. (The logical distance between a node and itself is always $log_2 1 = 0$.) In the general case, logical distances for semi-lattices are usually non-integers; in Figure 4 the logical distance between the two leaves marked with asterisks is $log_2 6$.

Intuitively, logical distances encode an *a priori* semantic distance metric from the built-in inductive bias set up by the abstractive relations. In relative frequency methods, each time a simple event occurs in the training set, its frequency is incremented by 1. We can view this as adding one "unit of count" to the simple event. The γ-continuum methods instead distribute the "unit" among all simple events, in a proportion that depends on logical distance. For each training instance t_n that is a simple event q_j, if the proportion of the "unit" given to the simple event is u_j, then the proportion u_i given to any other simple event q_i satisfies the constraint

$$\frac{u_i}{u_j} = 2^{-d(q_i, q_j)/\gamma}.$$

Let us first examine the extreme-case behavior. At $\gamma = 0$, $u_i/u_j = 0$ and so $u_i = 0$ for all $i \neq j$ and $u_j = 1$, thus degenerating into the relative frequency method. At $\gamma = \infty$, $u_i/u_j = 1$ and so $u_i = u_j$ for all i, thus incrementing every simple event equally, regardless of what the training instance is.

Now consider again the simple binary-tree space, and examine the effect of a single training instance 010 assuming $\gamma = 1$ as in Figure 5(b). The greatest proportion is $u_{010} = 0.4$, followed by a lesser proportion for the closest simple event $u_{011} = 0.2$, a still lesser proportion for $u_{000} = u_{001} = 0.1$, and finally $u_{100} = u_{101} = u_{110} = u_{111} = 0.05$. Figures 5(a) and (c) show how the value of γ controls the degree to which the "unit" is "smeared" toward progressively dissimilar families of events; the more "smear", the more generalization.

As a slightly more complex example, consider again the pattern space of Figure 4. A training set containing 100 instances was used. Figure 6 compares generalization behavior for three different values of γ. In (a), $\gamma = 0$ and so the distribution is exactly the relative frequency of the training instances. The non-uniform smoothing of the distribution in (b) and (c) shows the effect of the abstractive bias.

The joint probability for any compound event is just the sum of all its simple events' probability. If a compound event is comprised of a set of simple events $\{s_1, s_2, \ldots, s_r\}$ where $s_i \in Q$, then

$$Pr(X \in \{s_1, s_2, \ldots, s_r\}) = \sum_{i=1}^{r} \left[\frac{1}{N} \sum_{n=1}^{N} \frac{2^{-d(s_i, t_n)/\gamma}}{\left[\sum_{k=1}^{C} 2^{-d(s_k, t_n)/\gamma}\right]} \right]$$
$$= \frac{1}{N} \sum_{n=1}^{N} \frac{\left[\sum_{i=1}^{r} 2^{-d(q_i, t_n)/\gamma}\right]}{\left[\sum_{k=1}^{C} 2^{-d(q_k, t_n)/\gamma}\right]}.$$

Conclusion

The lack of theoretical tools has hampered the study of how *a priori* biases—especially abstractive biases—in a pattern completor's conceptual representation system affect the tendency to generalize. Generalization is necessary when the size of the training sample is small compared to the size of the domain, a condition that almost always obtains in the real

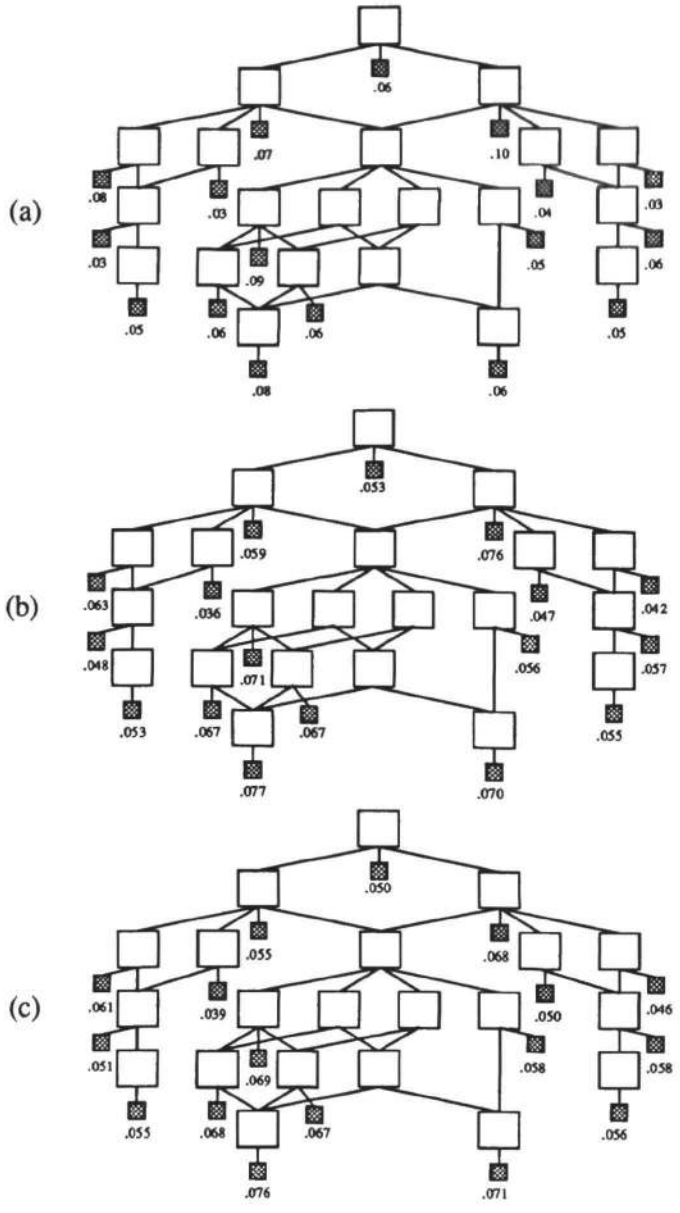

Figure 6: Generalization behavior for (a) $\gamma = 0$, (b) $\gamma = 0.8$, and (c) $\gamma = 1$.

world, in particular for human language acquisition and interpretation. Although neural networks do perform generalization because of their representational biases, the network compression loses so much information that it is difficult to tell what distribution is being learned; and moreover such networks do not (yet) handle compositional structures effectively. The family of methods we have proposed provides a declarative means to model different abstractive biases, and makes all induced distributions explicit.

The investigator must still determine empirically what concept space and abstractive relations are best for modeling cognitive biases in given domains. Moreover, for particular domains and abstractive relations, only empirical tests will tell what values of γ are useful. We are currently testing these methods for parsing and interpreting a corpus of nominal compounds, using an abstractive bias deriving from semantic network taxonomies.

Another future direction is the logical distance metric. Weighting schemes can be added to the logical structure to provide more a flexible modeling tool. Also, there are other possible logical distance metrics that possess the same essential characteristics.

Because of the size of the event space for practical domains, heuristic approximation methods are needed to evaluate these distributions. Whether heuristics can be used depends on the types of abstractive relations. In the case of the abstractive relations we use for parsing and interpretation, we are investigating various greedy algorithms including parallel intersection search techniques like marker passing (Wu 1989). Also, we are studying whether existing neural networks or other statistically-based models of generalization can function as heuristic approximation methods for certain types of abstractive relations.

References

Carnap, R. (1952). *The Continuum of Inductive Methods.* University of Chicago Press, Chicago.

Carnap, R. (1962). *The Logical Foundations of Probability.* University of Chicago Press, Chicago.

Cheeseman, P. (1985). In defense of probability. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pp. 1002–1009.

Cheeseman, P. (1987). A method of computing maximum entropy probability values for expert systems. In R. C. Smith & G. J. Erickson, editors, *Maximum-Entropy and Bayesian Spectral Analysis and Estimation Problems*, pp. 229–240. D. Reidel, Dordrecht, Holland. Revised proceedings of the Third Maximum Entropy Workshop, Laramie, WY, 1983.

Goldman, R. P. & E. Charniak (1990). A probabilistic approach to text understanding. Technical Report CS-90-13, Brown Univ., Providence, RI.

Hacking, I. (1975). *The Emergence of Probability.* Cambridge University Press, London.

Jaynes, E. T. (1979). Where do we stand on maximum entropy. In R. D. Levine & M. Tribus, editors, *The Maximum Entropy Formalism.* MIT Press, Cambridge, MA.

Mortimer, H. (1988). *The Logic of Induction.* Ellis Horwood, Chichester, England.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA.

Pollack, J. B. (1989). Implications of recursive auto associative memories. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, pp. 527–536. Morgan Kaufmann, San Mateo.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, 46:77–105.

Weatherford, R. (1982). *Philosophical Foundations of Probability Theory.* Routledge & Kegan Paul, London.

Wu, D. (1989). A probabilistic approach to marker propagation. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 574–580, Detroit, MI. Morgan Kaufmann.

Wu, D. (1990). Probabilistic unification-based integration of syntactic and semantic preferences for nominal compounds. In *Proceedings of the Thirteenth International Conference on Computational Linguistics*, Helsinki.