Machine Learning, 29, 91–101 (1997)

Learning with Probabilistic Representations

PAT LANGLEY* langley@rtna.daimlerbenz.com
Intelligent Systems Laboratory, Daimler-Benz Research & Technology Center, 1510 Page Mill Road, Palo Alto, CA 94304

GREGORY M. PROVAN gmprovan@rsc.rockwell.com
Rockwell Science Center, 1049 Camino dos Rios, Thousand Oaks, CA 91360

PADHRAIC SMYTH** smyth@ics.uci.edu
Department of Information & Computer Science, University of California, Irvine, CA 92697

1. Introduction and motivation

Machine learning cannot occur without some means to represent the learned knowledge. Researchers have long recognized the influence of representational choices, and the major paradigms in machine learning are organized not around induction algorithms or performance elements as much as around representational classes. Major examples include logical representations, which encode knowledge as rule sets or as univariate decision trees, neural networks, which instead use nodes connected by weighted links, and instance-based approaches, which store specific training cases in memory.

In the late 1980s, work on *probabilistic* representations also started to appear in the machine learning literature. This representational framework had a number of attractions, including a clean probabilistic semantics and the ability to explicitly describe degrees of certainty. This general approach attracted only a moderate amount of attention until recent years, when progress on Bayesian belief networks led to enough activity in the area to justify this special issue on the topic of probabilistic learning.

Representing uncertainty has a long and sometimes chequered history in artificial intelligence. Early work on knowledge-based systems, such as MYCIN and PROSPECTOR, modeled uncertainty explicitly and incorporated approximations to Bayesian inference. However, subsequent years saw probabilistic approaches largely ignored in AI and machine learning, until Pearl (1988) clearly demonstrated that probabilistic representations are less about *numbers* than about *structure*. He showed that a graphical notation, which lets one specify the independence assumption of a probabilistic model, has clear advantages for probabilistic inference. Only recently have researchers realized that Pearl's ideas (and related work in statistics) have profound implications for *learning*, but, as the papers in this issue show, belief networks and their associated graphical notation now play a prominent role in work on probabilistic induction.

A fundamental contribution of such probabilistic independence networks to learning is the notion that the complexity of a probabilistic representation is roughly inversely proportional to the number of independence assumptions it makes. The process of model building corresponds to searching among models by trading off complexity and fit to achieve good

[0]Also affiliated with the Institute for the Study of Learning and Expertise.
[0]Also affiliated with the Jet Propulsion Laboratory.

generalization accuracy. One approach is for the modeler to set the network structure a priori by making specific independence assumptions, thereby limiting model complexity, as assumed by most authors in this issue. A more general, but more difficult, approach is to place constraints on the type of independence structure allowed and let the algorithm consider structures within this constrained space, as in the paper by Friedman, Geiger, and Goldszmidt.

In this editorial, we briefly review two intertwined themes: that independence structure is a key factor in learning with probabilistic representations and that probabilistic formalisms inherit substantial characteristics from their non-probabilistic counterparts. The organization reflects the view that one can describe any approach to learning in terms of its:

- representation of learned knowledge;
- the manner in which it uses that knowledge;
- the fitness function used to direct search for knowledge structures; and
- the search method that uses this function to characterize a given training set.

For example, a specific learning algorithm might represent knowledge as a belief network, use a probabilistic inference algorithm to make predictions with that network, invoke likelihood of the joint probability density as its fitness function, and employ a greedy search method. In the following pages, we address each of these issues. In addition, we summarize the seven papers in the special issue and suggest some directions for future work on learning with probabilistic representations.

2. Classes of probabilistic representation

Despite the common motivations among learning researchers interested in probabilistic formalisms, the community has explored not a single probabilistic representation but many, typically relying on different forms of independence assumptions and following different variations on deterministic learning frameworks.

For example, one of the simplest probabilistic schemes is the *naive Bayesian classifier*, which is closely related to the one-layer perceptron in its number of parameters and its representational power. From an independence viewpoint, it makes the strong assumption that the attributes $A_i, i = 1, ..., n$ are conditionally independent of each other if the class variable is known. Briefly, for each discrete (symbolic) attribute value v_j assigned to attribute A_i and for class C_k, naive Bayes stores estimates for the conditional probability $p(v_j|C_k)$ of that value given the class, along with the probability $P(C_k)$ for each class. For continuous (numeric) attributes, one can either discretize their values into ranges or store some continuous distribution, like the normal, in terms of its mean and variance. The conditional independence assumption on the attributes is questionable for many domains but still useful, since it means few parameters and efficient processing. The naive Bayesian classifier was the earliest probabilistic framework to appear in the machine learning literature (e.g., Clark & Niblett, 1989; Kononenko, 1991); in this issue, Domingos and Pazzani discuss this approach, and the reasons for its success, in some detail.

Another class of representations, the *probabilistic concept hierarchy*, stores knowledge in a structure similar to a multivariate decision tree. Each node in such a hierarchy has two or more children, which correspond to specialized subclasses of their parent. The description at each node is similar to that in a naive Bayesian classifier, storing a conditional probability distribution for each attribute given the child. As with decision trees, a probabilistic concept hierarchy recursively partitions the instance space into subregions, with higher-level nodes describing more general classes and lower levels more specific ones. Most work in this framework also assumes conditional independence of attributes to some degree, but the multi-level nature gives much more representational power than naive Bayes. Although this issue includes no examples of work on probabilistic concept hierarchies, they have been represented in the machine learning literature for some time. Fisher (1987), Hanson, Stutz, and Cheeseman (1991), and Jordan and Jacobs (1993) report examples of this approach to probabilistic representation and learning.

Much recent work has focused on *belief networks*, which represent knowledge in a directed acyclic graph in which nodes correspond to attributes and links may indicate dependencies between attributes. Thus, they are similar in spirit to multilayer neural networks, an analogy that Binder, Koller, Russell, and Kanazawa explore in this issue. Stored with each node in a belief network is a table of conditional probabilities that, for each value of this attribute and for each combination of values for its parent attributes, specifies an estimate for the probability that this attribute's value will occur given the parent combination. Again, for numeric attributes one can discretize the values or use some distribution like the normal. To a first approximation, the absence of a link between two attributes in a belief network indicates the conditional independence of those attributes given their parents. An extreme case occurs with the naive Bayesian classifier, where the independence assumption leads to links only from the class attribute to the predictive attributes. Cooper and Herskovits (1992) report the earliest use of belief networks in the machine learning literature; in this issue, the papers by Binder et al., Chickering and Heckerman, Dasgupta, and Friedman et al. all address induction in Bayesian belief networks.

Much of the work on learning with probabilistic representations has focused on classification and inference tasks in which simple attribute-value formalisms are sufficient. However, domains like natural language and molecular biology involve sequential or temporal data, and researchers have used more powerful representational schemes in response.

The simplest probabilistic formalism for dealing with sequential data, *N grams*, simply stores the estimated joint probability for each possible sequence of N symbols. One can predict the overall probability of an extended sequence by combining the probabilities associated with its subsequences and one can learn these probabilistic chunks by simply counting the number of times that each occurs in training sequences. One can view the independence structure of an N-gram model as an $N-1^{th}$ order Markov assumption, which states that the current state is independent of previous symbols given the $N-1$ symbols that immediately preceded it. Although this approach has intuitive appeal, the number of parameters scales exponentially in N, thus leading to reliance on relatively simple 2-gram and 3-gram models in practice. Another response, which both Rissanen and Langdon (1981) and Ron, Singer, and Tishby (1994) have investigated, involves more flexible methods that store N grams of varying lengths.

Markov models constitute a more sophisticated approach to sequential and temporal domains, in that they are effectively a probabilistic variant on finite-state machines. They represent knowledge as a set of states, each with an associated symbol, and links between those states, each with an associated probability of transition. In an extension to this framework, *hidden Markov models*, the states themselves are unobservable variables that generate observable symbols from a set of such symbols, each with a distinct probability. This framework embodies some strong independence assumptions, in particular that each state depends directly only on the previous state, and each observable depends directly only on the current state. These assumptions let one use a model's structure and parameters to compute the probability that a given sequence of symbols will occur and to estimate the symbol and transition probabilities from a set of training sequences. One can also view a hidden Markov model as a type of belief network (Smyth, Heckerman, & Jordan, 1997), in which the hidden Markov structure is expressed as a long "chain" of successive states, with one link from each state to each observed variable.[1] Hidden Markov models have been used successfully in speech understanding, molecular biology, and other domains. In this issue, Ghahramani and Jordan describe an extension of this approach that factors hidden states into multiple variables.

Another formalism for sequential knowledge, *stochastic context-free grammars*, is a direct extension of the nonprobabilistic framework of context-free grammars. These represent knowledge as a set of rewrite rules, with a single nonterminal (unobservable) symbol on the left-hand side and one or more symbols on the right-hand side. One generates sequences of terminal (observable) symbols by starting with the root symbol, selecting a rule with it in the left-hand side, replacing this symbol with those in the right-hand side, and continuing to expand symbols until the sequence contains only terminals. The stochastic version associates a probability with each rewrite rule and assumes that this probability does not depend on the rules used to generate the symbol in its left-hand side. This assumption lets a stochastic context-free grammar compute the overall probability of any generated sequence of terminal symbols, and also to estimate the probabilities on each rewrite rule from a set of training sequences. The formalism has been widely used in statistical approaches to natural-language processing (e.g., Charniak, 1993); in this issue, Abe and Mamitsuka use an extension of the framework to learn about protein structure.

3. Performance elements and measures

Now that we have discussed representational issues, we can consider some approaches to using learned probabilistic knowledge. Having learned some probabilistic description, a system must still use that knowledge in some fashion, and one feature that distinguishes probabilistic representations and performance elements from logical ones is their reliance on the idea of *evidence combination*. Logical formalisms combine features in terms of conjunctions, disjunctions, or other relations, but they do so in an all-or-none manner.

In contrast, probabilistic methods treat the presence or absence of each feature as evidence, which they combine to determine the overall probability of some class or inference. They share this characteristic with connectionist methods and nearest-neighbor methods, but not with decision lists, univariate decision trees, or other logical approaches. The Bayesian philosophy also recommends combining evidence from different learned hypotheses through

weighted voting, but many systems use single learned probabilistic descriptions for reasons of inferential efficiency or comprehensibility.

Although some early work on belief networks emphasized their ability to acquire known network structures, recent efforts (including the papers in this issue) focus on their ability to improve along some *performance measure* that is closely tied to the performance task. If the goal is to classify new cases, then classification error or accuracy is the natural criterion, and a number of papers in this issue use that metric. If the aim is more flexible inference, because one does not know in advance the attributes present in each instance, then a more appropriate criterion is cross entropy, which measures the similarity of the observed and predicted probability distributions. Classification error is closely associated with the task of *supervised* learning, whereas cross entropy and related measures are associated with the task of *unsupervised* learning.[2]

We can reformulate these learning tasks and performance measures in probabilistic terms. Given a set of variables $X_1, \ldots, X_d$, supervised learning aims to learn conditional densities, e.g., $p(X_1|X_2, \ldots, X_d)$, where X_1 is the variable to be predicted. If X_1 takes on symbolic values, then the performance task is classification; if X_1 takes on real values, then the problem is regression. Accurate estimation of the full conditional density is sufficient but not always necessary for accurate prediction, as Domingos and Pazzani clarify in their article in this issue. Unsupervised learning, sometimes called *density estimation*, deals with the problem of inducing the full joint density function $p(X_1, \ldots, X_d)$, since none of the variables have preferred status over the others.

The above measures cut across different methodological goals. Most work on probabilistic learning, and most papers in this issue, take an experimental approach, finding the average accuracy or cross entropy when training and testing specific induction algorithms on real-world or synthetic data. However, one can also carry out formal analyses, as the papers by Dasgupta and by Domingos and Pazzani in this issue show. And probabilistic frameworks have also received attention in psychological circles, where they are evaluated in terms of their ability to match human learning behavior (e.g., Anderson & Matessa, 1992; Fisher & Langley, 1990).

The important point is that, although probabilistic learning methods are unique on some important dimensions, they are subject to the same performance measures and methodological criteria as other approaches to induction. This allows direct comparisons between probabilistic algorithms and more traditional ones, as the papers by Domingos and Pazzani, Friedman et al., and Binder et al. reveal.

4. Fitness functions

Algorithms that learn probabilistic descriptions require some way to select from among a large set of candidate descriptions. Such descriptions can vary in both their structure and in the probabilistic parameters that instantiate that structure. This leads naturally to two formulations of the learning problem: some approaches assume a given structure and focus on determining the best parameters, whereas others also select among alternative structures.

Within each framework, most probabilistic methods aim to find a description that optimizes some 'fitness' function. One common function is the likelihood of the data given the model, which is the probability of the training cases conditioned on the hypothesized struc-

ture and its parameters. In this issue, the papers by Abe and Mamitsuka, by Binder et al., and by Ghahramani and Jordan all incorporate such a *maximum likelihood* approach. This framework makes sense when the structure is given, but it can introduce problems when learning the structure, since more complex models will always have higher likelihood. In such cases, the metric does not indicate which model will generalize the best to new data.

Another common fitness function, designed to address this issue, measures the probability of each model given the observed data using

$$p(M|data) \propto p(data|M)p(M) \quad ,$$

where $p(data|M)$ is the *marginal likelihood* and $p(M)$ is the prior probability of the model. Approaches differ in how they calculate these two components. For example, one method calculates the first term by integrating over all possible parameter values,

$$p(M|data) = \int p(data|\theta, M)p(\theta|M)d\theta \quad ,$$

where $p(\theta|M)$ is the prior probability of the parameter values. However, this integral is often intractable when there are hidden variables, which has led some researchers to explore approximations. As Chickering and Heckerman note in this issue, one common approach is to use the maximum likelihood estimate for θ and some simple penalty term for $p(M)$. As Friedman et al. point out in their paper, one can view fitness functions like minimum description length as deriving from such approximations.

Although Bayesian methods like the above are often used in work on learning probabilistic structures, other approaches have also seen use. One technique, *cross validation*, directly estimates the expected performance of alternative hypotheses on data held out from the training set. For instance, Singh and Provan (1995) have used this measure to evaluate candidate structures for belief networks. Some techniques instead attempt to optimize information-theoretic measures over hypothesized structures. For example, Fisher (1987) has used *category utility*, which balances the predictiveness of variables against their predictability, to select among alternative hierarchies of probabilistic concepts.

5. The search process

The final component of a probabilistic learning algorithm specifies how to maximize the given fitness function over the set of hypotheses for a particular training set. Typically, this involves carrying out *search* through a space of hypotheses. A few approaches, such as naive Bayes and N grams, are constrained enough that this is not necessary; the collection of simple statistics produces the best parameter settings for their given structures. Similar methods are sufficient for arbitrary belief networks, provided their structures are specified, there are no hidden variables, and the data have no attributes with missing values. However, in other situations, some form of search is needed to find acceptable structures or parameters.

Probabilistic learning methods that assume a given structure need only search through the space of parameter values for that structure. Such methods typically carry out some form of gradient descent, in which the current parameter settings and the training data are used iteratively to generate new parameter settings, with the process continuing until it reaches

some halting condition. One popular technique of this sort, known as the *expectation maximization* or *EM* algorithm, operates (roughly speaking) by averaging over missing or unknown values to determine parameters of the current hypothesis (the 'expectation' step) and then computing revised parameter settings that give better results on the fitness function (the 'maximization' step).

In this issue, Chickering and Heckerman use EM to find parameter settings for belief networks with one hidden variable, whereas Jordan and Jacobs (1993) have used it for probabilistic concept hierarchies with fixed structure. Two other papers in the issue, by Ghahramani and Jordan and by Abe and Mamitsuka, discuss the forward-backward and inside-outside algorithms, which are variations on EM designed for sequential domains. There are also approaches other than EM for finding probabilistic parameters; in this issue, Binder et al. describe another gradient-descent method, based on an analogy with back-propagation in neural networks, that finds good parameter values in a belief network with hidden variables.

However, methods of this sort are not sufficient when the learning task includes selecting among different candidate structures. When there exist only a few alternatives, the system can examine all structures exhaustively, as do Chickering and Heckerman in their paper. More frequently, the size of the hypothesis space makes this intractable,[3] and most induction systems carry out more constrained forms of search through the space of structures.

For offline learning tasks, in which all training data are available at once, the most common approach is greedy search, in which one places some partial ordering on the space, selects the best-scoring hypothesis on each step, and continues until no improvement occurs. In this issue, Friedman et al. describe a greedy method that searches from simpler structures to more complex ones, while Cooper and Herskovits (1992) use a similar scheme to learn less constrained belief networks. The search process is directed by the fitness function (or some approximation of it) that the system aims to optimize.

For online learning, in which training cases become available one at a time, the typical response is incremental hill climbing. In this approach, each new instance can lead the system to modify the hypothesized knowledge structure, provided this change improves the fitness function. Because decisions must be based on fewer cases, different orders of training cases can produce different results. Fisher (1987) has used incremental hill climbing to learn hierarchies of probabilistic concepts, whereas Stolcke and Omohundro (1994) have used this approach to induce the structure of stochastic context-free grammars.

These search methods should sound familiar to researchers from other induction paradigms. Greedy and hill-climbing search occupy central roles in methods for rule learning, decision-tree induction, and neural networks, so it should come as no surprise that they have also proven useful for learning with probabilistic descriptions. Thus, the main difference between the probabilistic framework and others lies not in their search methods, but rather in their representation of learned knowledge and the manner in which they use that knowledge in new situations.

6. Papers in the issue

The papers in this special issue reveal the variety of representations that are possible within the probabilistic framework. We have organized the issue by representational complexity,

with papers on simple representations coming first and those on more complex notations later. This ordering highlights an important issue: there exists a tradeoff, which a number of papers discuss explicitly, between representational power and the complexity of learning and classification.

Thus, the initial paper, by Domingos and Pazzani, examines the naive or simple Bayesian classifier. As we have mentioned, this method makes strong independence assumptions, yet typically fares very well compared to more sophisticated approaches to supervised induction. The authors give further evidence for its competitive behavior and also show that this is not because the commonly used data sets lack attribute dependencies. Instead, they prove that, under certain conditions, naive Bayes learns an optimal classifier even when significant dependencies are present. Their basic analysis, which they back with additional experiments, revolves around the tradeoff between bias and variance. They emphasize the difference between regression tasks (for which squared loss functions make sense) and classification tasks (for which zero-one loss functions are appropriate). The naive Bayesian classifier excels at the latter because it very often assigns cases to the right class, even though its precise probability estimates are off.

The second article, by Friedman et al., describes another approach, tree augmented naive Bayes (TAN), that adopts a representational position midway between naive Bayes and a full Bayesian belief network. In particular, their hypothesis space includes networks in which each predictive attribute has no more than two parents – the class attribute and one other feature. Their learning algorithm carries out a greedy search through the space of such structures, using a technique that runs in time that is a polynomial function of the number of attributes; their classification method is similarly efficient compared to those for arbitrary belief networks. In contrast, learning belief networks with unconstrained structures has complexity exponential in the number of variables. The authors demonstrate that their learning method performs significantly better than naive Bayes over a wide range of data sets, and that it also fares better than more sophisticated techniques for creating unrestricted belief networks.

In the third paper, Dasgupta reports on a theoretical investigation into the complexity of learning the probability parameters for fixed-structure belief networks. In particular, he presents a PAC (probably approximately correct) analysis that establishes bounds on the sample complexity of inducing belief networks with and without hidden nodes. One very interesting finding is that, given the complexity of learning a given network, the sample complexity does not increase drastically when one uses hidden units, which is important because such hidden nodes can make the representation more compact and easier to understand. These results are timely, in that only recently has the experimental community started to focus its energies on learning belief networks with hidden nodes.

After this, Chickering and Heckerman address a related task: approximations to the Bayesian estimation of a belief network with one hidden variable. This problem is equivalent to finding clusters in unsupervised training data, where the hidden attribute corresponds to the cluster variable. The authors review a variety of methods that estimate the parameters of such models from the data, but that vary in their simplifying assumptions and their computational cost. Chickering and Heckerman report experimental results with these techniques on both natural and synthetic data, comparing them to an expensive but accurate Monte Carlo method. They find that most algorithms give accurate parameters when used

for model selection but not for model averaging, that most approximations are sensitive to the prior probabilities, and that some methods are considerably more efficient than others.

Binder, Koller, Russell, and Kanazawa also deal with learning the probabilities for belief networks; however, they focus on networks with known but arbitrary structure and with many missing attributes. They describe a gradient-descent algorithm for estimating the entries in conditional probability tables, similar in spirit to backpropagation for neural networks but with a probabilistic semantics. Experimental studies with two handcrafted network structures revealed that their method's learning rate (the number of instances needed to reach low error) was much faster than for a network with no hidden terms, and that its asymptotic error was less than for a comparable neural network with weights learned through backpropagation. The authors also extend their approach to parameterized representations, networks with noisy-OR nodes, networks that describe temporal processes, and ones that include continuous variables.

The sixth paper, by Ghahramani and Jordan, reviews approaches to learning hidden Markov models and describes an approach that factors the hidden state variable into multiple terms. Their method uses the expectation-maximization algorithm to infer the model's parameters from training data; however, because exact calculation of the expectation step is intractable, they develop both Monte Carlo and variational approximations for this subtask. Ghahramani and Jordan report experiments on both synthetic data (generated from a handcrafted factorial Markov model) and natural sequences that describe Bach's chorales. Their results suggest that the variational approximation provides a good tradeoff between accuracy and computational efficiency, recommending its use over the other methods that they considered.

The final article, by Abe and Mamitsuka, deals with an even more sophisticated class of representations. They review the notion of stochastic context-free grammars, which bear the same relation to hidden Markov models as do context-free grammars to finite-state machines. They note that this formalism cannot handle languages with certain long-distance relations, so they describe a more general class, stochastic tree-augmented grammars, with even more expressive power. They also report methods for probabilistic parsing with such grammars and an extension of the inside-outside algorithm for estimating rule probabilities from training strings. Experiments on the difficult task of predicting protein structure suggest that this approach can learn regularities that have eluded previous methods.

7. Directions for future research

Although the papers in this issue reflect the substantial progress that has occurred in developing and understanding algorithms for probabilistic learning, there remain many open issues that call for more work in the area. For example, the literature on probabilistic methods still makes insufficient contact with research on other approaches to induction, and future studies should try to understand the conditions under which probabilistic techniques fare better than the alternatives and vice versa. Some papers in the issue, such as those by Domingos and Pazzani and by Friedman et al., have made a good start in this direction, but we need more research along these lines, especially theoretical analyses, to understand better the factors that affect learning behavior.

Perhaps because research on probabilistic induction is a relatively recent phenomenon, some necessary steps remain before such methods are ready for real-world application. These include developing techniques for dealing with missing data, automatically introducing latent (hidden) variables into the learned structure, and selecting relevant features from among many alternatives. The papers by Binder et al. and by Chickering and Heckerman address the first of these issues, whereas Singh and Provan (1995) have addressed the third, but there remains a need for more work in these areas. Moreover, most methods for probabilistic induction have high computational costs, which means that their application to domains with large data sets must await more efficient variations.

Another important need, which the probabilistic induction community shares with the rest of machine learning, is to move beyond simple classification and prediction tasks to more complex domains. Probabilistic representations are appropriate for encoding knowledge about natural language (e.g., Charniak, 1993), problem solving (Jones & VanLehn, 1994), motor skills (Iba & Gennari, 1991), and vision (Geman & Geman, 1984), yet most researchers avoid these topics and focus instead on simple inference tasks. An increased concern with such problems would encourage effort on more expressive probabilistic representations that move beyond "flat" formalisms to ones that explicitly encode spatial and temporal relations in a hierarchical manner.

A central theme in traditional machine learning has been the role of knowledge in constraining the induction process, and research on probabilistic learning should also give more attention to this issue. The probabilistic framework provides a natural means of encoding such knowledge (e.g., about the structure of a belief network) in terms of prior probability distributions but, in practice, this scheme has been used mainly to indicate only generic biases, such as a preference for simpler models. However, belief networks and other probabilistic representations lend themselves naturally to techniques for *theory revision* (e.g., Ourston & Mooney, 1990; Towell, Shavlik, & Noordeweier, 1990), in which the learner starts its search from some existing model rather than from scratch, and this approach seems likely to give much better results in domains where knowledge is available.

Despite the challenges that remain, work on learning with probabilistic representations appears in a vigorous state. As the papers in this issue reveal, researchers are exploring a variety of representations within the general probabilistic framework, addressing a number of challenging problems, and carrying out careful evaluations of their approaches. These features bode well for this emerging paradigm, and we expect that future work on probabilistic learning will continue to reflect this encouraging trend.

Notes

1. Note that the structure of this belief network is not the same as the finite-state graph often shown for hidden Markov models.
2. For sequential tasks, one defines analogous measures that take order into account.
3. The number of possible network structures is exponential in the number of network nodes.

References

Anderson, J. R., & Matessa, M. (1992). Explorations of an incremental, Bayesian algorithm for categorization. *Machine Learning*, *9*, 275–308.

Charniak, E. (1993). *Statistical language learning*. Cambridge, MA: MIT Press.

Hanson, R., Stutz, J., & Cheeseman, P. (1991). Bayesian classification with correlation and inheritance. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence* (pp. 692–698). Sydney: Morgan Kaufmann.

Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, *3*, 261–284.

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*, 309–347.

Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*, 139–172. Reprinted in J. W. Shavlik & T. G. Dietterich (Eds.) (1990), *Readings in machine learning*. San Francisco: Morgan Kaufmann.

Fisher, D. H., & Langley, P. (1990). The structure and formation of natural categories. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 26). Cambridge, MA: Academic Press.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–742.

Iba, W. & Gennari, J. H. (1991). Learning to recognize movements. In D. H. Fisher, M. J. Pazzani, & P. Langley (Eds.), *Concept formation: Knowledge and experience in unsupervised learning*. San Francisco: Morgan Kaufmann.

Jones, R. M., & VanLehn, K. (1994). Acquisition of children's addition strategies: A model of impasse-free, knowledge-level learning. *Machine Learning*, *16*, 11–36.

Jordan, M. I., & Jacobs, R. A. (1993). Supervised learning and divide-and-conquer: A statistical approach. *Proceedings of the Tenth International Conference on Machine Learning* (pp. 159–166). Amherst, MA: Morgan Kaufmann.

Kononenko, I. (1991). Semi-naive Bayesian classifier. *Proceedings of the Sixth European Working Session on Learning* (pp. 206–219). Porto, Portugal: Pittman.

Ourston, D., & Mooney, R. (1990). Changing the rules: A comprehensive approach to theory refinement. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 815–820). Boston: AAAI Press.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Francisco: Morgan Kaufmann.

Ron, D., Singer, Y., & Tishby, N. (1994). The power of amnesia. In J. D. Cowan, G. Tesauro, & J. Alspector (Eds.) *Advances in Neural Information Processing Systems 6*. San Francsico: Morgan Kaufmann.

Rissanen, J., & Langdon, G. G. (1981). Universal coding and modeling. *IEEE Transactions on Information Theory*, *27*, 12–23.

Singh, M., & Provan, G. M. (1995). A comparison of induction algorithms for selective and non-selective Bayesian classifiers. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 497–505). Lake Tahoe, CA: Morgan Kaufmann.

Smyth, P., Heckerman, D., & Jordan, M. I. (1997). Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, *9*, 227–269.

Stolcke, A., & Omohundro, S. (1994). Inducing probabilistic grammars by Bayesian model merging. *Proceedings of the Second International Conference on Grammatical Inference and Applications* (pp. 106–118). Alicante, Spain: Springer-Verlag.

Towell, G., Shavlik, J., & Noordeweier, M. O. (1990). Refinement of approximate domain theories by knowledge-based neural networks. *Proceedings of the Eighth National Conference on Artificial Intelligence* (pp. 861–866). Boston: AAAI Press.