The Use of Relevance to Evaluate Learning Biases

Marie desJardins
SRI International
333 Ravenswood Ave.
Menlo Park CA 94025
marie@erg.sri.com

Abstract

This paper describes Probabilistic Bias Evaluation (PBE), a method for evaluating learning biases by formally analyzing the tradeoff between the expected accuracy and complexity of alternative biases. Intelligent agents must filter out irrelevant aspects of the environment, in order to minimize the costs of learning. In PBE, probabilistic background knowledge about relevance is used to compute expected accuracy; and the complexity of a bias is used to estimate the costs of learning. These are combined into a single value that can be used to select the best bias: one which maximizes predictive accuracy while minimizing computation cost.[1]

Introduction

Probabilistic Bias Evaluation (PBE) is a method for analytically evaluating alternative biases for inductive learning. Inductive learning requires that a description of a concept be formed, given a set of training instances for that concept. A concept description is a mapping from the attribute values of an object to its concept class, or outcome. (For the purposes of this paper, attributes are considered to be nominal-valued.) A set of attributes and a description language using those attributes define a concept description space containing the set of mappings that the agent will consider.

Probabilistic background knowledge is used in PBE to determine a maximally relevant set of attributes. Selecting only these attributes, rather than using all known attributes, reduces the computational complexity of learning. On the other hand, using fewer attributes can cause poor learning performance. PBE provides a formal, quantitative analysis of this tradeoff between accuracy and complexity.

Declarative Bias

Bias refers to a restriction on, or preference within, the space of concept descriptions considered by a learning system. A strong, correct bias is extremely useful to have because it allows a learner to converge quickly to a good concept description. How to find a good learning bias has been an open research question since Mitchell (1980) first introduced the concept of bias. PBE considers only the set of attributes used as the domain of the concept mapping; for the remainder of this paper, *bias* will be used to refer to a set of attributes.

Russell and Grosof (1987) showed that for deterministic learning problems, certain types of bias can be represented as declarative background knowledge in the form of determinations. A *determination*, as defined in Davies and Russell (1987), represents a dependency between relational schemata. P determines Q ($P \succ Q$) if all objects that share the same value for P also share the same value for Q. Formally,

$$P(x,y) \succ Q(x,z) \quad \text{iff} \quad \forall wyz[P(w,y) \wedge Q(w,z) \rightarrow \forall x[P(x,y) \rightarrow Q(x,z)]] \qquad (1)$$

Expressing declarative bias by using determinations is straightforward, because learning consists of applying deductive logic to the bias and observations to yield consistent rules. However, determinations are rarely available in environments containing uncertainty, where the concepts to be learned may be nondeterministic. Such domains require a probabilistic form of background knowledge, and a way of using this knowledge to impose a bias on learning.

Probabilistic Bias Evaluation

The best bias is not simply the one that defines a description space that is the most likely to contain the "correct" concept description, since any superset of a given bias will always be better—or at least as good—by this measure. A useful bias evaluation metric should make a tradeoff between accuracy and simplicity, so that an attribute that would significantly increase the size of the concept description space, while yielding only slightly better concept descriptions, will not be considered adequately relevant to be included in the bias.

[1]This work is partially supported under the ARPA/Rome Laboratory Planning Initiative, contract number F30602-93-C-0071.

The consequences of selecting a larger description space are twofold. First, it will take more observations to converge on a good concept description. Second, searching a larger space takes more computational time for each observation. The impact on the system's overall performance depends on how much the agent discounts future rewards and on the cost of time in the environment.

In this paper, *time* should be taken to refer to the number of observations made, unless otherwise indicated. The analysis in PBE only considers the number of observations, not the computational time per observation. However, these measures are related in that they both depend on the size of the bias; thus, computational time per observation will be indirectly minimized.

The value that PBE assigns to each potential bias is a measure of that bias's expected behavior in the long run. The bias value is the expected discounted future accuracy of the predictions that would be made if the bias were used for learning. *Uniformities,* which represent probabilistic knowledge about the distribution of outcomes (O) given biases, or attribute sets (A), are used to derive the expected accuracy of the best concept description. This expected accuracy is combined with a learning curve, yielding the expected accuracy of concept descriptions over time. A time-preference function is then used to find the overall expected discounted accuracy, which is the value of the bias. The steps in this process are described in the following sections. A more detailed description of PBE can be found in desJardins (1994).

Probabilistic Background Knowledge

Determinations are *weak knowledge*, in that they do not specify what the function mapping the inputs P to the output Q is, only that one exists. So, for example, knowing that $\text{Species}(x,s) \succ \text{Color}(x,c)$ does not allow one to predict the color of an individual of a previously unobserved species. But after observing one individual of species s_1 whose color is known to be c_1, one can immediately form a rule that says $\forall x[\text{Species}(x,s_1) \rightarrow \text{Color}(x,c_1)]$. The latter sort of rules—which enable individual predictions to be made—will be referred to as *strong knowledge.*

Probabilistic background knowledge about relevance is represented in PBE by using a form of weak knowledge called *uniformities*, which are a probabilistic version of determinations. $U(O|A)$ (read "the uniformity of O given A") is the probability that two random individuals that have the same values of the attributes A will have the same value of the outcome O. Roughly, $U(O|A)$ is the degree to which O can be predicted, given A. Like a determination, $U(O|A)$ does not specify what the most common value of O will be for any given value of A. Formally,

$$U(O|A) = P(O(x) = O(y)|A(x) = A(y)) \quad (2)$$

Since a uniformity is simply a probability statement about two independent random variables, x and y, a system can learn and reason with uniformities as it would with other forms of probabilistic knowledge. Initial uniformities are provided by the system designer or by domain experts. The initial uniformities can be updated as the system acquires experience, and new uniformities can be learned by generalizing strong knowledge (Russell 1986).

In our framework, then, *relevance* is a probabilistic notion. Specifically, an attribute or set of attributes A is relevant for predicting the value of an output attribute O if it increases the marginal uniformity of O: i.e., if

$$U(O|A) > U(O) \quad (3)$$

Note, however, that we are not just interested in choosing all relevant attributes, but the attributes that are sufficiently relevant to be worth exploring, given the costs of using those attributes for learning. The goal of the PBE method is to provide a formal framework for making this tradeoff.

Expected Accuracy

To find the expected accuracy of predictions over time, PBE first computes the expected accuracy of the best concept description in the space defined by A. A simple prediction task is assumed: every time a new example (specifying values of A) is observed, the agent must predict a value of O. If the most likely outcome is always predicted (maximizing expected accuracy), the expected accuracy of the best concept description is $\hat{p}$, as given below.

The distribution of O, given A, is assumed to satisfy the following two conditions:

1. The distribution is unimodal; that is, for each value of A, there is one value of O, $\hat{o}$, that occurs most often.
2. The other values of O occur equally often.

Suppose that O can take on n different values, $o_1, \ldots, o_n$. Assumption 1 says that given A, some $\hat{o}$ has the highest probability. Without loss of generality, assume that this is o_1; its probability is $\hat{p}$:

$$P(O(x) = o_1|A) = \hat{p} \quad (4)$$

Assumption 2 says that the remaining values of O have equal probability. If there are n values of O,

$$P(O(x) = o_1|A) = \frac{1-\hat{p}}{n-1}, \quad i = 2, \ldots, n \quad (5)$$

The definition of uniformity (Equation 2) states that

$$U(O|A) = P(O(x) = O(y)|A(x) = A(y)) \quad (6)$$

Substituting the probabilities from Equations 4 and 5, and solving for $\hat{p}$ in terms of $U(O|A)$, gives

$$\hat{p} = \frac{1 + \sqrt{1 - n + n(n-1)U(O|A)}}{n} \quad (7)$$

Learning Curves

Results from computational learning theory suggest that the number of examples $\hat{m}$ needed to learn a concept is proportional to the Vapnik-Chervonenkis (V-C) dimension of the bias (Blumer *et al.* 1986). In other words, $\hat{m} = cd$, where d is the V-C dimension of the bias, and c is a constant that depends on the error rate. (For any particular learning algorithm, c is best approximated empirically.)

The V-C dimension d for a decision tree, for example, is equal to the size of the space of possible examples, and

$$\hat{m} = cd = c \prod_{i=1}^{a} n_i \tag{8}$$

where a is the number of attributes in the bias, and n_i is the number of values of attribute i.

This paper makes the simplifying assumption that prior to finding the best concept description ($t < \hat{m}$), predictions are no better than random guessing; after this, they are as good as they are expected to get (i.e., the probability of a correct prediction is $\hat{p}$). Then the accuracy of predictions as a function of time will be

$$q(t) = \begin{cases} \frac{1}{n} & \text{if } t < \hat{m} \\ \hat{p} & \text{otherwise.} \end{cases} \tag{9}$$

The actual learning curve will, of course, be smoother; however, for relatively small, simple concept description spaces, this curve appears to be a reasonable approximation. For more accurate results, though, a better learning curve will be needed; unless results from computational learning theory can be applied, a learning curve should be approximated empirically.

Time Preference Functions

The effect of the passage of time on the value of predictions depends on a variety of factors, including the amount of computation time available, what the prediction task is, the cost of making errors, the life expectancy of the agent, how fast the environment is changing, and how many other tasks need to be accomplished simultaneously with this learning task.

The effects of these various factors are modeled in PBE with a time-preference function $\mathcal{T}(t)$. Time-preference functions are used in decision analysis (Holtzman 1989) to indicate the degree to which the importance of reward changes over time (i.e., the degree of discounting of future rewards). If an intelligent agent's prediction task involves making a single prediction at time t_0, for example, only the accuracy of the agent's prediction at that moment matters: earlier and later performance is irrelevant. In this case, the time-preference function is zero at all points except for a spike at time t_0.

A reasonable time-preference function for a simple autonomous agent constantly performing predictions in a dynamic environment is γ^t, based on a constant discount rate γ, close to 1. Intuitively, using $\mathcal{T}(t) = \gamma^t$ means that accurate predictions in the distant future are exponentially less important than near-term accuracy; but any correct prediction, no matter how distant, has some positive value. The closer γ is to 1, the more heavily long-term accuracy is counted. The value of γ will depend on the particular environment in which the agent finds itself, and should be determined experimentally.

Expected Value of Biases

Combining the bias's accuracy over time with the time-preference function $\mathcal{T}(t)$, and integrating over time, yield a general equation for the value of a bias:

$$V = \int_1^\infty q(t)\,\mathcal{T}(t)\,dt \tag{10}$$

Using the simplified learning curve from Equation 9 and letting $\hat{m} = cd$ and $\mathcal{T}(t) = \gamma^t$ gives

$$V = \int_1^{cd} \frac{\gamma^t}{n}\,dt + \int_{cd}^\infty \gamma^t\,\hat{p}\,dt \tag{11}$$

$$= \left[\frac{\gamma^t}{n \ln \gamma}\right]_1^{cd} + \left[\frac{\hat{p}\,\gamma^t}{\ln \gamma}\right]_{cd}^\infty \tag{12}$$

$$= \frac{-1}{\ln \gamma}\left[\gamma^{cd}(\hat{p} - \frac{1}{n}) + \frac{\gamma}{n}\right] \tag{13}$$

Notice that γ and n are constant for a given learning task; therefore, the only term that will differ among candidate biases is $\gamma^{cd}(\hat{p} - 1/n)$. Intuitively, if γ is large, so that the agent is willing to wait for accurate predictions, d has less influence on the value (in the extreme case, $\gamma = 1$ and $\gamma^{cd} = 1$, regardless of d). As $\hat{p}$ grows, the bias becomes more predictive, and the value of the bias increases.

Results

The effects of the cost associated with larger attribute sets were measured using ID*, a probabilistic, incremental version of ID3 based on Quinlan (1986) and Utgoff (1988). (ID* and the procedure used for these tests are described in detail by desJardins [1992].) We ran ID* in a synthetic learning domain, using various subsets of the full attribute set as the learning bias; the results are summarized here.

In each test run, 100 training examples were generated, and ID* was used to build four sets of decision trees, with different subsets of the attributes. The average predictive quality n (number of test examples classified correctly) is shown as a function of the number of training examples seen in Figure 1. (The attribute names are arbitrary and used only to aid in the exposition.)

Figure 2 shows the expected accuracy $\hat{p}$ (computed from the uniformities for the synthetic domain); the expected number of correct predictions on a test set

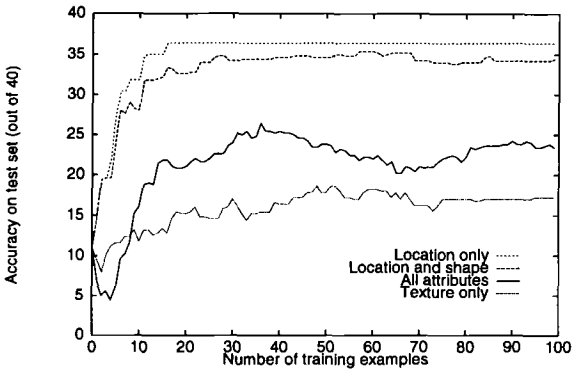

Figure 1: Results of learning with four different biases

Attributes	$\hat{p}$	E(n)	n
All	.98	39.2	23.4
Location	.90	36.0	36.4
Location and shape	.95	38.0	34.4
Texture	.50	20.0	17.2

Figure 2: Expected and actual accuracy

of forty examples; and the actual number of correct predictions on the test set after 100 training examples.

The smaller biases performed approximately as well as expected, given the 100 training examples, but the larger sample size needed for convergence in larger spaces is clearly hindering the performance of the learning algorithm. Presumably, given enough training examples (and computation time), the bias using all of the attributes would converge on nearly perfect prediction accuracy, but the marginal amount of accuracy gained over the predictions made by the smaller biases is unlikely to be significant for many learning tasks. Considering that this is a relatively simple domain, the degree to which learning is impaired when the full attribute set is used is rather surprising.

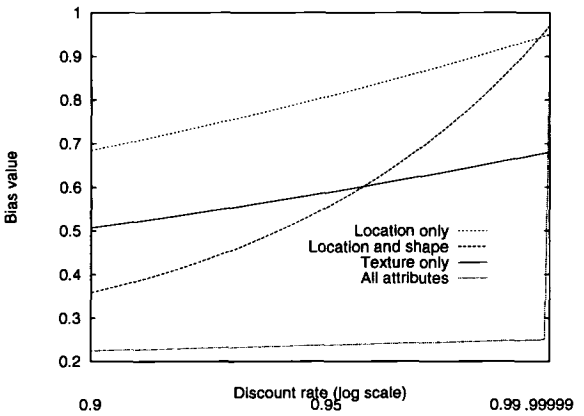

Figure 3: Relative bias values

The relative bias values for the domain are shown in Figure 3. Location is the best choice unless γ is very high (location and shape outperform location only when γ is around 0.999). Using all of the attributes is not worthwhile unless γ is extremely close to 1.

Conclusions

In realistic learning domains, there will be large numbers of irrelevant and marginally relevant attributes. Intelligent agents must filter out the aspects of the environment that are not important for the learning task at hand, and they must do this in a way that is sensitive to the context of learning. PBE provides a formal framework for evaluating and selecting appropriate and useful learning biases.

References

Blumer, A.; Ehrenfeucht, A.; Haussler, D.; and Warmuth, M. K. 1986. Classifying learnable geometric concepts with the Vapnik-Chervonenkis dimension. In *Proc. 18th ACM Symposium on Theory of Computation*, 273–282.

Davies, T., and Russell, S. 1987. A logical approach to reasoning by analogy. Technical Report Note 385, AI Center, SRI International.

desJardins, M. 1992. *PAGODA: A Model for Autonomous Learning in Probabilistic Domains.* Ph.D. Dissertation, UC Berkeley. (Available as UCB CS Dept. Technical Report 92/678).

desJardins, M. 1994. Evaluation of learning biases using probabilistic domain knowledge. In *Computational Learning Theory and Natural Learning Systems, vol. 2.* The MIT Press. 95–112.

Holtzman, S. 1989. *Intelligent Decision Systems.* Addison-Wesley.

Mitchell, T. 1980. The need for biases in learning generalizations. Technical Report CBM-TR-117, Rutgers University.

Quinlan, R. 1986. The effect of noise on concept learning. In Michalski, R.; Carbonell, J.; and Mitchell, T., eds., *Machine Learning II.* Morgan Kaufman. 149–166.

Russell, S. J., and Grosof, B. N. 1987. A declarative approach to bias in concept learning. In *AAAI*, 505–510.

Russell, S. J. 1986. *Analogical and Inductive Reasoning.* Ph.D. Dissertation, Stanford University.

Utgoff, P. E. 1988. ID5: An incremental ID3. In *Machine Learning Conference*, 107–120.