LEARNING CONCEPT DESCRIPTIONS FROM EXAMPLES WITH ERRORS

Jakub Segen

AT&T Bell Laboratories,
Holmdel, N.J. 07733

ABSTRACT

This paper presents a scheme for learning complex descriptions, such as logic formulas, from examples with errors. The basis for learning is provided by a selection criterion which minimizes a combined measure of discrepancy of a description with training data, and complexity of a description. Learning rules for two types of descriptors are derived: one for finding descriptors with good average discrimination over a set of concepts, second for selecting the best descriptor for a specific concept. Once these descriptors are found, an unknown instance can be identified by a search using the descriptors of the first type for a fast screening of candidate concepts, and the second for the final selection of the closest concept.

1. Introduction

While the majority of the AI work on learning concentrates in error free domains, there is an acknowledged need for learning techniques directed towards noisy data [Dietterich and Michalski, 1983], [Mitchell, 1982]. A problem of a major importance in learning from data with errors is the choice of the preference criterion for ranking competing descriptions. The criteria such as maximum likelihood, minimum error, or minimum estimated entropy which are generally used for inference from noisy data, suffice for inferring simple parametric models, but are not well suited to learning in rich spaces of symbolic descriptions used in AI. These criteria minimize the discrepancy between a description and the training observations. If the language used to form descriptions is sufficiently rich to express the training data, such criteria will rank a description that exactly matches the training observations as better or equal to any other description. For example, if the space of descriptions includes predicate calculus expressions, a concept A represented in the training set by three instances, whose parameter "length" assumes values 4.6, 5.2, and 5.7, might generate a description: *(length(A) - 4.6 OR length(A) • 5.2 OR length(A) - 5.7).* Any errors in the training data will be represented in such an overspecified description along with possible regularities. A version of this problem known as the "curse of dimensionality" appears even with simple vector models when the number of dimensions is not specified, [Kanal, 1974].

One way of preventing the inference process from generating overspecified descriptions is to include some measure of description complexity in the preference criterion, to bias it towards simple descriptions. This idea is well known in philosophy of science (Occam's razor), and various measures of complexity (or simplicity) were proposed in AI literature [Michalski and Stepp, 1983], [Michalski, 1983], [Mitchell, 1980], [Buchanan and Mitchell, 1978]. A specific question is the trade-off between the complexity of a description and the discrepancy with data. A criterion objectively combining these two measures by relating both to Kolmogorov'g complexity [Kolmogorov, 1968], was introduced in [Segen, 1980] and called minimal representation criterion.

In this paper we apply the minimal representation criterion to derive general rules for learning concept descriptions from noisy training data. These rules can be used to learn symbolic descriptors, such as logic formulas, as well as parametric models. In Section 2 of this paper we summarize the minimal representation criterion, in Section 3 we apply it to derive selection rules for two types of descriptors: concept specific descriptors and globally useful system descriptors, and to decide which descriptors should be used with default values. In Section 4 we show how to apply both types of descriptors to classify instances using bottom-up and top-down strategies.

2. Minimal Representation Criterion

Consider the problem of rinding a program for a Turing machine, to generate a given finite sequence of observations. While there are infinitely many programs for any such sequence, it seems reasonable to chose the shortest program since it represents the least commitment and minimum redundancy. If we treat a program to be a randomly generated binary sequence with O's and I's having equal probability, then the shortest program is also the most probable one. The problem of selecting a probability model *P(y)* from a sequence of observations $y^n = y_1, y_2, \ldots, y_n$ can be recast as a case of the above problem by establishing an isomorphism between the class of probability distributions and a subset of programs for a Turing machine [Segen, 1980]. Selecting the shortest program in this subset corresponds to finding a probability distribution minimizing the expression

$$Q(P(y), y^n) = S(P(y)) - \log P(y^n) \qquad (1)$$

where *S(P(y))* is the number of bits needed to specify the probability distribution *P(y).* All the logarithms used in this paper are in the base 2. The above criterion for estimating the probability distribution has been called the *minimal representation criterion.* Its main difference from the maximum likelihood criterion (equivalent to seeking a minimum of $-\log P(y^n)$) comes from the term $S(P(y))$, which is a measure of a complexity of the specification of the probability distribution *P(y).* Including it in the criterion in effect penalizes more complex distributions.

Properties of the minimal representation criterion were treated formally in [Segen, 1980]. It has been applied to discover patterns in a continuous signal and in a symbol sequence, and to such problems as selecting the number of clusters.

3. Choosing Concept Descriptors

The problem of selecting a single descriptor for each concept can be stated as follows: Given is a training set T, consisting of a set of instances $X_1, X_2 \cdots X_k$ and a concept assignment for each of the instances: $X_k \rightarrow C_i$. Also given is a space F of functions, which we call descriptor functions or *descriptors,* defined on the domain of instances. A descriptor can be any computable function with a probability distribution defined on its range of values. For each of the concepts $C_1, C_2 \ldots C_m$ we want to select a descriptor $f \in F$ that is most

helpful in deciding whether an instance with unknown concept assignment should be assigned to this concept.

We approach the descriptor selection indirectly, as a problem of estimating the conditional probability $P(C_i|X)$ while its form is restricted to:

$$P(C_i|X) = P(C_i|f(X)) \tag{2}$$

Descriptor selection is a part of the task of finding the estimate of this form, for which we will use the minimal representation criterion. If the instances and their concept assignments in the training set are independent, we can write the logarithm of the probability of the concept assignments given in the training set T as

$$\log P(T) = \sum_{i=1}^{m} \sum_{j=1}^{n(i)} \log P(C_i|f(X_{ij})) \tag{3}$$

where $X_{i1}, X_{i2} \cdots X_{in(i)}$ are the instances assigned to concept C_i. Choice of the best descriptor for a given concept C_i can be treated as a single concept problem. In this case:

$$\log P(T) = \sum_{j=1}^{n(i)} \log P(C_i|f(X_{ij})) + \sum_{j=1}^{n-n(i)} \log P(\sim C_i|f(X_{\sim ij})) \tag{4}$$

Using Bayes's formula, equation (4) can be written as:

$$\log P(T) = \sum_{j=1}^{n(i)} \log \left[\frac{P(f(X_{ij})|C_i)}{P(f(X_{ij}))} \cdot P(C_i) \right] + \sum_{j=1}^{n-n(i)} \log \left[\frac{P(f(X_{\sim ij})|\sim C_i)}{P(f(X_{\sim ij}))} \cdot (1-P(C_i)) \right] \tag{5}$$

where

$$P(f(X)) = P(f(X)|C_i))\ P(C_i) + P(f(X)|\sim C_i)(1-P(C_i))$$

Applying the minimal representation criterion, we should minimize:

$$Q(T,P(T)) = S(P(f(X)|C_i)) + S(P(f(X)|\sim C_i)) + S(P(C_i)) + S(f) - \log P(T) \tag{6}$$

with respect to f, $P(f(X)|C_i)$, $P(f(X)|\sim C_i)$ and $P(C_i)$. Selection of the best descriptor is a part of the minimization task. Computation of $S(P)$ for parametric probability models was discussed in [Segen, 1980]. Evaluation of $S(f)$ is based on a representation of the descriptor. For example, if a descriptor is a concatenation of k primitive operations, we need $S(f) = (k+1)*\log(n+1)$ bits to express it, where n is the number of primitive operations available. The best descriptor can be found by searching the space F in order of increasing descriptor complexity, and this search will always terminate [Segen, 1980]. If descriptors are binary valued then (6) is minimized for each descriptor by replacing probabilities in (5) with corresponding frequencies, which makes the learning of logic formulas particularly simple.

A descriptor f that results in a higher value of $Q(T,P(T))$ than a "no descriptor" case, i.e. the case when $P(C_i|X) = P(C_i)$, will not be selected even if there are no other competing descriptors. We will call such a descriptor not *informative* with respect to C_i. The best estimate of its conditional probability $P(f(x)|C_i)$ is the $P(f(x))$, which can be used as a *default* distribution for C_i.

Once the best single descriptor for a concept is found we can search for additional descriptors to form a *conjunctive description*. If descriptors $f_1, f_2 \cdots f_k$ are independent for C_i and $C_{\sim i}$ we can write:

$$P(C_i|f_1,f_2,..f_k) = \frac{P(f_1|C_i)\cdot P(f_2|C_i) \cdots P(f_k|C_i)}{\sum_{i=1}^{n} P(C_i)\cdot \prod_{j=1}^{k} P(f_j|C_i)} \tag{7}$$

Using this form in (4) to compute $P(T)$ we can search for descriptors incrementally, i.e., having found descriptors $f_1,f_2,...f_v$ we can find a descriptor f_{v+1} minimizing the criterion while using the previously found distributions for $f_1,f_2 ...f_v$. This process terminates when an addition of another descriptor does not decrease the value of $Q(T,P(T))$. Notice, that we do not assume that all descriptors are independent, but search for the best set of descriptors using (7) as a constraint. Also, the inconsistency pointed out in [Pendault, Zucker, and Muresan, 1981] does not appear since there are only two disjoint sets.

Descriptors selected in the above process are useful for a specific concept and we call them *c–descriptors*. In addition we can search for globally useful descriptors that discriminate among many concepts at once. They will be called system descriptors or *s–descriptors*. System descriptors can be selected and rated using the expression (1) with:

$$\log P(T) = \sum_{i=1}^{m} \sum_{j=1}^{n(i)} \log \frac{P(f(X_{ij})|C_i)\cdot P(C_i)}{P(f(X_{ij}))} \tag{8}$$

Here we include the conditional probabilities for all the concepts, so descriptors are rated according to their average discriminating power. We can select and order some number, or all informative s-descriptors. All the s-descriptors that are informative with respect to a concept C_i will form a set SD_i. This set along with the values of $P(f|C_i)$ can be considered a probabilistic version of frame.

4. Using Descriptors to Classify Instances

A direct approach to classifying an unlabeled instance is to compute the probability $P(C_i|X)$ for every known concept using c-descriptors, and select the concept with the highest probability. However, if there are many concepts this strategy can be expensive. In such a case, we can use a data-driven search based on s-descriptors to select quickly several most likely concepts, and then evaluate only the c-descriptors of the selected concepts. Let $f_1, f_2, \ldots f_k$ be the results of applying any k s-descriptors to the instance X. We can roughly evaluate the probability $P(C_i|f_1,f_2, \ldots f_k)$ by assuming their conditional independence.

$$P(C_i|f_1,f_2, \ldots f_k) \propto P(C_i)\cdot \prod_{j=1}^{k} P(f_j|C_i) \tag{9}$$

We do not need to normalize the right hand side, since we are interested only in order imposed by the probability, not its absolute value. This proportion will not be affected if the right hand side is divided by $P(f_1)*P(f_2)* \ldots *P(f_k)$, for all i (this value is not equal to $P(f_1,f_2, \ldots f_k)$, since we assume only conditional, and not marginal independence).

$$P(C_i|f_1,f_2, \ldots f_k) \propto P(C_i)\cdot \prod_{j=1}^{k} \frac{P(f_j|C_i)}{P(f_j)} \tag{10}$$

Now, since $\frac{P(f_m|C_i)}{P(f_m)} = 1$ by default when the s-descriptor f_m is not a member of the set SD_i, we can eliminate all such factors from

the product. Also, the order imposed by the above proportion will be preserved if we take a logarithm of the right hand side. We will call the result an *evidence* towards the concept C_i based on the facts $f_1, f_2, \ldots f_k$.

$$EV(C_i|f_1,f_2,\ldots f_k) = \log P(C_i) + \sum_{f_j \in SD_i} \log \frac{P(f_j|C_i)}{P(f_j)} \quad (11)$$

The evidence provides the same ordering of concepts as the probability but its much less expensive to compute, since evaluation of a descriptor changes the evidence only for a subset of all concepts. This subset is known a priori for each descriptor and it can be much smaller than the set of all concepts known to the system. Therefore, we can provide links from each s-descriptor to the affected concepts and update only the evidence for these concepts after evaluating the descriptor. An expression similar to (11), but without the above feature, was presented in [Charniak, 1983]. The decision for switching from evidence accumulation to evaluating s-descriptors can come in two ways: either evidence for some concept reaches a given threshold, or after evaluating some number of s-descriptors the concepts are sorted and tested in order of decreasing evidence.

If the range of a descriptor f_m is a small set of discrete values $v_{m1}, v_{m2}, \ldots$, we can set a weighted link from each outcome v_{ml} to each concept C_i for which f_m is informative, with the weight of the link equal to $\log(P(v_{ml}|C_i) / P(v_{ml}))$. A firing outcome simply adds link weights to the evidence of corresponding concepts. Such an organization clearly resembles models of neural nets, and it can be carried out in a parallel architecture such as Thistle [Fahlman, Hinton, and Sejnowski, 1983].

5. Concluding Remarks

The methods proposed here for learning of descriptors apply to both parametric models and logic formulas. They are particularly simple for predicate descriptors since their probability can be estimated as frequency. While we have not discussed domain specific descriptor generators, many of the generation schemes presented in AI literature [Dietterich and Michalski, 1983] [Cohen and Feigenbaum, 1982], [Michalski, 1983] are compatible with the methods of this paper. A side result that might become important for large systems is the automatic assignment of a default status to some descriptors. The most important direction for future work lies in developing incremental learning strategies, needed for both time and storage efficiency.

REFERENCES

[1] T. G. Dietterich and R. S. Michalski, "A Comparative Review of Selected Methods for Learning Structural Descriptions," Machine Learning, Michalski R. S., Carbonell, J. G., and Mitchell, T. M. (Eds.), Tioga, Palo Alto, 1983.

[2] T. M. Mitchell, "Generalization as Search," Artificial Intelligence, Vol. 18, No. 2, pp. 203-226, March 1982.

[3] L. Kanal, "Patterns in Pattern Recognition," IEEE Trans. Inform. Theory, Vol. IT-20, pp. 697-722, Nov. 1974.

[4] R. S. Michalski and R. Stepp, "Learning from Observations: Conceptual Clustering," Machine Learning, Michalski R. S., Carbonell, J. G., and Mitchell, T. M. (Eds.), Tioga, Palo Alto, 1983.

[5] R. S. Michalski "A Theory and Methodology of Inductive Learning," Machine Learning, Michalski R. S., Carbonell, J. G., and Mitchell, T. M. (Eds.), Tioga, Palo Alto, 1983.

[6] T. M. Mitchell, The Need for Biases in Learning Generalizations," Rutgers University, CS Dept. Report, May 1980.

[7] B. G. Buchanan and T. M. Mitchell, "Model Directed Learning of Production Rules," in Waterman and Hayes-Roth (ed.) "Pattern Directed Inference Systems," Academic Press, New York, 1978.

[8] A. N. Kolmogorov, "On the Logical Basis of Information Theory and Probability Theory," IEEE Trans. Inform. Theory, IT-14, pp. 662-664, 1968.

[9] J. Segen, "Pattern-Directed Signal Analysis," PhD Thesis, Carnegie-Mellon Univ., Pittsburgh, 1980.

[10] E. P. D. Pendault, S. W. Zucker, and L. V. Muresan, "On the Independence Assumption Underlying Subjective Bayesian Updating," Artificial Intelligence, 16(2) pp.213-222, 1981.

[11] E. Charniak, The Bayesian Basis of Common Sense Medical Diagnosis," Proc. AAAI-83, pp.70-73, 1983.

[12] S. E. Fahlman, G. E. Hinton, and T. J. Sejnowski, "Massively Parallel Architectures for AI: NETL, Thistle, and Boltzmann Machines," Proc. AAAI-83, pp. 109-113, 1983.

[13] P. R. Cohen and E. A. Feigenbaum (Eds.), The Handbook of Artificial Intelligence," HeurisTech Press, Stanford, ch. 14, 1982.