A Statistical Approach to Decision Tree Modeling

Michael I. Jordan
Department of Brain and Cognitive Sciences
Massachusetts Institute of Technology
Cambridge, MA 02139
jordan@psyche.mit.edu

Abstract

A statistical approach to decision tree modeling is described. In this approach, each decision in the tree is modeled parametrically as is the process by which an output is generated from an input and a sequence of decisions. The resulting model yields a likelihood measure of goodness of fit, allowing ML and MAP estimation techniques to be utilized. An efficient algorithm is presented to estimate the parameters in the tree. The model selection problem is presented and several alternative proposals are considered. A hidden Markov version of the tree is described for data sequences that have temporal dependencies.

1 INTRODUCTION

Decision tree algorithms have been studied throughout machine learning and statistics as a nonparametric approach to data modeling (Breiman, et al., 1984; Quinlan, 1993). Decision tree methodology is often contrasted with classical parametric statistical methodology, which requires the formulation of an explicit probabilistic model of the data generation process. The parametric approach is often viewed with suspicion, as being at worst an arbitrary imposition of the modeler's assumptions on the data and at best an inflexible approach to modeling data. In this paper, I show that these criticisms, while not without merit, are misdirected. I describe a statistical approach to decision tree modeling that allows classical parametric statistical ideas to be put to good use in the context of decision trees. The problem of decision tree induction is treated as a problem in parameter estimation and model selection.

There are a number of advantages to formulating decision tree problems within a statistical framework. These advantages include the ability to make use of optimization algorithms that take advantage of properties of likelihood functions, the ability to use theoretical results from likelihood theory and Bayesian theory in analyzing the performance of the tree, and the ability to generalize the basic decision tree architecture in interesting and well-motivated directions. Moreover, there is a clear potential for realizing performance gains from the statistical approach. Consider a linear regression problem (Figure 1(a)). In regression analysis, the randomness inherent in the data induces randomness in the estimates of the parameters for the slope and intercept of the regression line. The variance in these estimates is affected by the spread of the data on the x-axis, a phenomenon referred to as *leverage* (Draper & Smith, 1981). Leverage is quadratic; that is, data points that are furthest from the mean x value have a substantially greater influence on the variance than data points near the mean. Consider now a piecewise linear fit as would be performed by a decision tree that has chosen a splitting point

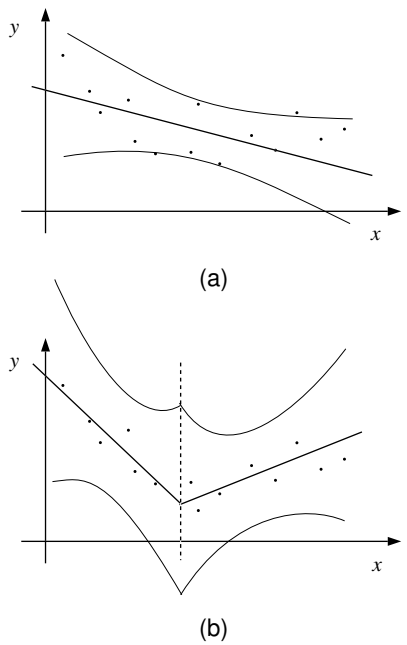

Figure 1: (a) Error estimates for a regression line are based on the variances of the parameter estimates. (b) Piecewise linear regression. The variance is larger because of the decreased leverage. (NB: These graphs are hand drawn and are merely meant to be suggestive.)

somewhere on the x-axis (Figure 1(b)). Each separate regression slope has substantially greater variance than the slope of the global regression, partly because of the decrease in the number of points contributing to each separate regression, but also because of the loss of leverage. One way to control this increase in variance is to allow points to have a certain amount of influence across a split. Allowing this influence will of course increase the bias of the parameter estimates; however, because there is uncertainty in the choice of the best splitting point, data points on one side of the split may in fact provide useful information about the parameters on the other side of the split. Due to the quadratic nature of leverage, a small increase in bias may be more than justified by a larger decrease in variance. This tradeoff can yield improved prediction performance. A statistical approach to decision tree modeling takes advantage of this tradeoff between bias and variance in a natural way. As described below, the basic idea is to utilize a statistical model for each decision in the decision tree. These statistical models smooth across the splits to the extent allowed by the data.

The remainder of this paper presents an overview of a statistical approach to decision tree modeling. Many of the ideas presented here are described in more detail in Jordan and Jacobs (1994).

2 PROBABILITY MODELS FOR DECISION TREES

A likelihood-based approach to decision tree induction requires a probabilistic model of the process by which data are generated. For a given input $\boldsymbol{x}$, we assume that a sequence of probabilistic decisions are taken that result in the generation of a corresponding output $\boldsymbol{y}$. We do not require that this sequence of decisions have a direct correspondence to a process in reality, rather the decisions may simply represent an abstract set of "twenty questions" that specify, with increasing precision, the location of the conditional mean of $\boldsymbol{y}$ on a nonlinear manifold that relates inputs to mean outputs.

We consider regression models in which $\boldsymbol{y}$ is a real-valued vector and classification models in which $\boldsymbol{y}$ is either a binary scalar or a binary vector with a single non-zero component. In either case the goal is to formulate a conditional probability density of the form $P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is a parameter vector. Maximizing a product of N such densities with respect to $\boldsymbol{\phi}$ (where N is the sample size) yields a maximum likelihood estimate of $\boldsymbol{\phi}$. Bayesian maximum a posterior estimation can be handled by incorporating a prior on the parameter vector. In a later section, we consider a Markov model in which the likelihood of a data sequence is not simply the product of N independent densities.

A probabilistic model of a decision tree involves a sequence of probabilistic decisions, each conditional on the input $\boldsymbol{x}$ and conditional on previous decisions. As shown in Figure 2, we model the first decision by utilizing a set of random decision variables $\{\omega_i\}$, where the probabilities

$$P(\omega_i|\boldsymbol{x}, \boldsymbol{\eta})$$

sum to one and depend both on $\boldsymbol{x}$ and on a parameter vector

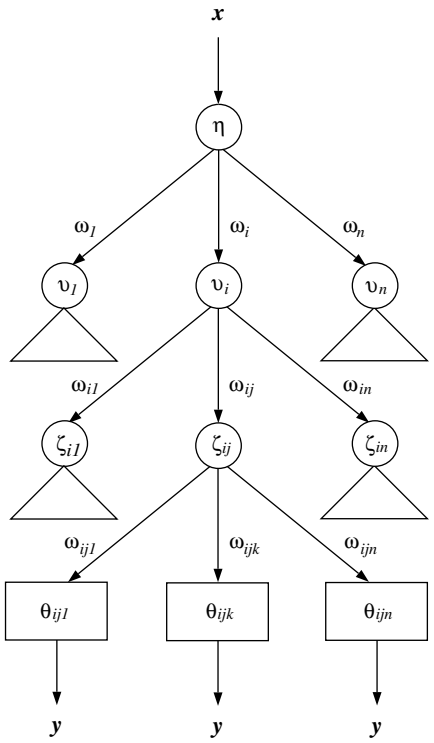

Figure 2: A probabilistic decision tree. The decisions ω are modeled by parametric statistical models having parameters $\{\boldsymbol{\eta}, \boldsymbol{\nu}_i, \boldsymbol{\zeta}_{ij}, \ldots\}$. At the leaves of the tree, an output $\boldsymbol{y}$ is generated via a statistical model parameterized by parameters $\boldsymbol{\theta}$. The total probability of $\boldsymbol{y}$ given $\boldsymbol{x}$ is a mixture of the path probabilities.

$\boldsymbol{\eta}$. The second decision is conditional on the first and is modeled by a set of probabilities

$$P(\omega_{ij}|\boldsymbol{x}, \omega_i, \boldsymbol{\nu}_i)$$

where $\boldsymbol{\nu}_i$ are additional parameters. A sequence of decisions $\omega_i, \omega_{ij}, \ldots, \omega_{ij\cdots k}$ terminates at a leaf node which contains a local probability model for generating $\boldsymbol{y}$ from $\boldsymbol{x}$. This model is of the form

$$P(\boldsymbol{y}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots, \boldsymbol{\theta}_{ij\cdots k})$$

where $\boldsymbol{\theta}_{ij\cdots k}$ is a parameter vector for the local probability model at leaf node $i, j, \ldots, k$.

A particular vector $\boldsymbol{y}$ can be generated via multiple paths through the tree. To obtain the total conditional probability of $\boldsymbol{y}$ given $\boldsymbol{x}$ we sum over all such paths to obtain the following probability model

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\phi}) \quad = \quad & \sum_i P(\omega_i|\boldsymbol{x}, \boldsymbol{\eta}) \sum_j P(\omega_{ij}|\boldsymbol{x}, \omega_i, \boldsymbol{\nu}_i) \\ \cdots \quad & \sum_k P(\omega_{ij\cdots k}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots, \boldsymbol{\xi}_{ij\cdots k'}) \\ & P(\boldsymbol{y}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots, \boldsymbol{\theta}_{ij\cdots k}) \end{aligned} \tag{1}$$

where $\boldsymbol{\phi}$ is the collection of all the parameters in the tree and k' is the index preceding the index k in the sequence

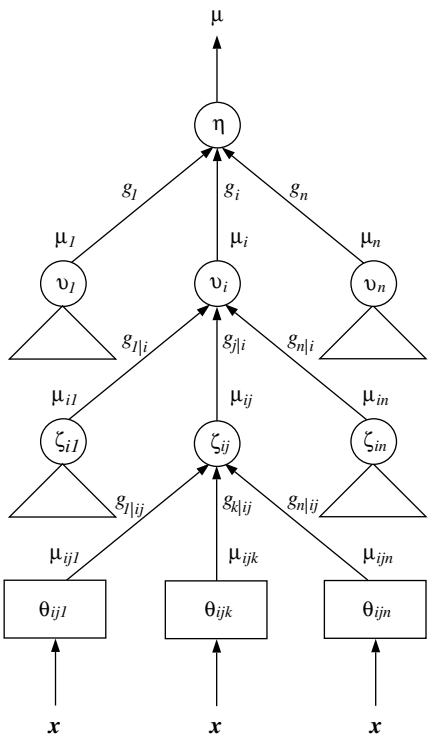

Figure 3: Propagating the conditional means in the decision tree.

$i, j, \ldots, k$. This probability model is a hierarchical mixture density in which both the mixing proportions and the mixture components are parametric densities conditional on $\boldsymbol{x}$.[1] To utilize this model for estimation, we maximize the log likelihood

$$l(\boldsymbol{\phi}, \mathcal{X}) = \sum_p P(\boldsymbol{y}^{(p)}|\boldsymbol{x}^{(p)}, \boldsymbol{\phi})$$

where $\mathcal{X}$ is the training set: $\mathcal{X} = \{(\boldsymbol{x}^{(p)}, \boldsymbol{y}^{(p)})\}_{p=1}^N$.

Equation 1 gives the conditional density of $\boldsymbol{y}$ for each $\boldsymbol{x}$. Moments of this conditional density are readily obtained. For example, conditional means can be obtained by an recursion upward in the tree. Letting $\boldsymbol{\mu}_{ij\cdots k}$ denote $E(\boldsymbol{y}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots, \boldsymbol{\theta}_{ij\cdots k})$ we have

$$\boldsymbol{\mu}_{ij\cdots k'} = \sum_k g_{k|ij\cdots k'} \boldsymbol{\mu}_{ij\cdots k}$$

where the symbol $g_{k|ij\cdots k'}$ is used to denote the probability $P(\omega_{ij\cdots k}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots, \boldsymbol{\xi}_{ij\cdots k'})$. As shown in Figure 3, these conditional means are propagated upwards, yielding

$$\boldsymbol{\mu} = E(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\phi})$$

at the top of the tree.

[1] Jordan and Jacobs (1994) refer to this model as the "hierarchical mixture of experts" (HME) model, where the term "expert" refers to one of the local regression or classification models at a leaf of the tree.

3 POSTERIOR CREDIT AND THE EM ALGORITHM

Within a probabilistic framework, learning can be viewed quite generally as the problem of assigning posterior credit to structural components of the learning system. Algorithms can be developed for assigning posterior credit to parameters, to collections of parameters, and to other significant structural components of a learning system, including the system itself. One of the major advantages of a probabilistic approach is this ability to assign credit to whole structural components. Moreover, the probabilistic approach makes it possible to combine posterior credit in a consistent manner.

There is a broad class of iterative estimation algorithms in statistics, known as *Expectation-Maximization (EM)* algorithms (Baum, et al., 1970; Dempster, Laird, & Rubin, 1977), in which the idea of posterior credit assignment assumes a particularly prominent role. A typical application of EM begins by identifying a set of "hidden variables," which, if they were known, would substantially simplify the estimation problem. In a clustering problem, for example, the hidden variables might correspond to the unknown cluster "labels" associated with each data point. Each iteration of the EM algorithm begins by taking the expectation of the hidden variables (the so-called "E-step" of EM), conditional on the data and conditional on the current values of the parameters. Because the hidden variables are typically indicator random variables, this step essentially assigns posterior credit to the corresponding structural components of the model. In Hidden Markov Models (HMM's), for example, the relevant structural components are the states of the underlying Markov chain, and the hidden variables are therefore the identities of the states at each moment in time. The E-step of EM for HMM's is a recursive computation (the "forward-backward algorithm") that yields a posterior probability distribution across the states at each moment in time.

Once posterior credit has been assigned to components of the system, the parameters associated with each component can be updated. This update (the "M-step" of EM) is generally achieved by solving a maximum likelihood estimation problem independently for each component, conditional on the expectations obtained during the E-step. In the clustering problem, for example, each cluster is often modeled as a Gaussian distribution, and the M-step updates the means of each distribution by weighting the data points by their associated posterior probabilities.

3.1 Applying EM to probabilistic decision trees

EM can be applied to the problem of estimating the parameters of a probabilistic decision tree in a straightforward manner. The nature of a decision tree is that each subtree is itself a decision tree, thus these subtrees are natural structural components to which posterior probability can be meaningfully assigned. The decisions taken at each of the nonterminals provide a corresponding set of hidden variables. Taking the expected value of these decision variables effectively assigns posterior credit to the corresponding subtree. Moreover, as is shown below, there is a recursive relationship between poste-

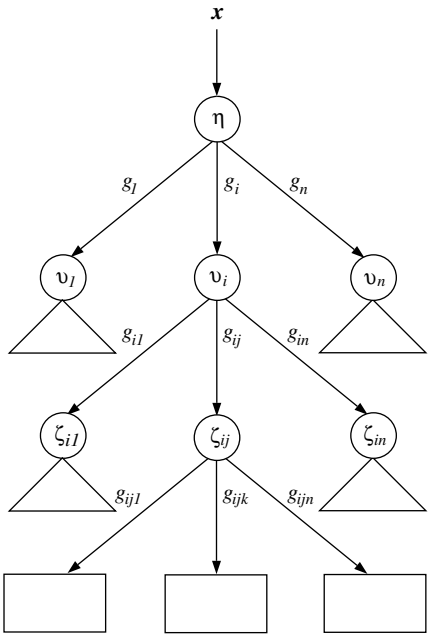

Figure 4: Propagating the prior probabilities in the decision tree. The symbol $g_{ij\cdots k}$ denotes the probability $P(\omega_i, \omega_{ij}, \ldots, \omega_{ij\cdots k}|\boldsymbol{x})$.

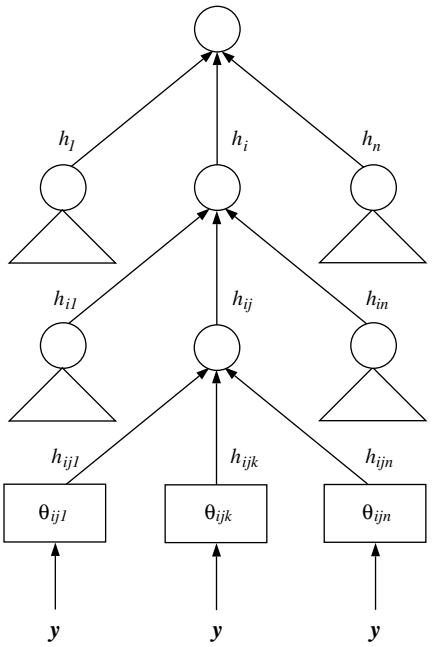

Figure 5: Propagating the posterior probabilities in the decision tree.

rior credit such that the posterior credit of a nonterminal can be computed by combining the posterior credit of its daughter nodes.

The derivation of an EM algorithm for the probabilistic decision tree is presented in Jordan and Jacobs (1994); here we simply outline the major steps of the resulting algorithm. The major computational task is to compute posterior probabilities at each node of the tree for each data point. Consider the posterior probability $P(\omega_i|\boldsymbol{x}, \boldsymbol{y})$.[2] Using Bayes' rule, this probability can be written as follows:

$$P(\omega_i|\boldsymbol{x}, \boldsymbol{y}) = \frac{P(\omega_i|\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x}, \omega_i)}{\sum_i P(\omega_i|\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x}, \omega_i)}$$

Note that each of the probabilities on the right hand side appears in the probability model. In general, at depth k in the tree, an application of Bayes' rule yields

$$P(\omega_i, \omega_{ij}, \ldots, \omega_{ij\cdots k}|\boldsymbol{x}, \boldsymbol{y}) = \frac{P(\omega_i|\boldsymbol{x})P(\omega_{ij}|\boldsymbol{x}, \omega_i)\cdots}{\sum_i P(\omega_i|\boldsymbol{x})\sum_j P(\omega_{ij}|\boldsymbol{x}, \omega_i)\cdots} \frac{\cdots P(\omega_{ij\cdots k}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots)P(\boldsymbol{y}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots)}{\cdots\sum_k P(\omega_{ij\cdots k}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots)P(\boldsymbol{y}|\boldsymbol{x}, \omega_i, \omega_{ij}, \ldots)}$$

The numerator of this expression can be computed via a downward recursion in the tree, taking products of the $g_{k|ij\cdots k'}$ factors (cf. Figure 4).

Denoting the posterior $P(\omega_i, \omega_{ij}, \ldots, \omega_{ij\cdots k}|\boldsymbol{x}, \boldsymbol{y})$ by $h_{ij\cdots k}$, it is easily verified that

$$h_{ij\cdots k'} = \sum_k h_{ij\cdots k}$$

showing that posterior probabilities can be propagated upward in the tree and summed to compute posterior probabilities for parent nodes (cf. Figure 5).

An alternative way of organizing the posterior credit computation is to recurse upward to compute the conditional posterior probabilities

$$h_{k|ij\cdots k'} \equiv P(\omega_{ij\cdots k}|\boldsymbol{x}, \boldsymbol{y}, \omega_i, \omega_{ij}, \ldots, \omega_{ij\cdots k'})$$

followed by a downward recursion to convert these conditional posterior probabilities into the desired joint probabilities ($h_{ij\cdots k}$).

Once the posterior probabilities have been computed, the parameter update is straightforward. The EM derivation shows that the parameters $\boldsymbol{\eta}$ at the top level of the tree can be updated by maximizing the expression

$$\sum_p \sum_i h_i^{(p)} \log g_i^{(p)} \tag{2}$$

which is the cross entropy between the posterior probabilities $h_i^{(p)}$ (where the superscript p indexes the training set) and the decision probabilities $g_i^{(p)} \equiv P(\omega_i|\boldsymbol{x}^{(p)}, \boldsymbol{\eta})$.[3] A similar cross entropy term is obtained at level k of the tree, where the posterior distribution is $h_{ij\cdots k}^{(p)}$ and the decision probabilities are $g_{ij\cdots k}^{(p)}$. At the leaves of the tree the parameters $\boldsymbol{\theta}_{ij\cdots k}$ are updated by maximizing

$$\sum_p h_{ij\cdots k}^{(p)} \log P(\boldsymbol{y}^{(p)}|\boldsymbol{x}^{(p)}, \omega_i, \omega_{ij}, \ldots, \boldsymbol{\theta}_{ij\cdots k}) \tag{3}$$

which is a weighted maximum likelihood problem.

[2] In this formula and in the remainder of the section, we drop reference to the parameters to simplify the notation.

[3] As described in the following section, the cross entropy is the log likelihood under a multinomial logit model.

4 COMPONENT DENSITIES

A wide variety of decision tree models can be obtained by choosing particular parametric forms for the component densities in the decision tree. The choice of component densities is determined by a variety of factors, including the type of problem being solved (regression, classification, etc.), possible prior knowledge about the form of the regression surface, the need for diagnostics of tree performance, and the desire for an interpretable result. The computational complexity of the overall estimation procedure is also heavily dependent on the choice of component densities as is the ability to analyze the model theoretically.

Let us consider first the component densities that generate $\boldsymbol{y}$ from $\boldsymbol{x}$ at the leaf nodes of the tree. For regression problems, a simple and natural option is a linear regression model with additive Gaussian noise. A mixture of such models (cf. Equation 1) yields a smoothed, piecewise linear regression surface. The parameter update (Equation 3) is simply a weighted least squares problem, where the weights are the posterior probabilities $h_{ij\cdots k}$ and $\boldsymbol{\theta}_{ij\cdots k}$ is a matrix:

$$\boldsymbol{\theta}_{ij\cdots k} = \text{argmin}_{\boldsymbol{\theta}_{ij\cdots k}} \sum_p h_{ij\cdots k}^{(p)} \|\boldsymbol{y}^{(p)} - \boldsymbol{\theta}_{ij\cdots k}\boldsymbol{x}^{(p)}\|^2$$

For binary classification problems, linear regression is an option, but a preferable alternative is logistic regression. In this case the parameter update involves solving the following system of equations for $\boldsymbol{\theta}_{ij\cdots k}$:

$$\sum_p h_{ij\cdots k}^{(p)} (y^{(p)} - f(\boldsymbol{\theta}_{ij\cdots k}^T \boldsymbol{x}^{(p)}))\boldsymbol{x}^{(p)} = 0$$

where f is the logistic function (Jordan & Jacobs, 1994). This problem can be solved efficiently by a quadratically-convergent algorithm known as (weighted) iteratively reweighted least squares (IRLS) (McCullagh & Nelder, 1983). In the case of multiway classification, a generalization of logistic regression known as the multinomial logit model can be used. A variety of other classification models can also be considered.

Our emphasis is on using simple parametrizations at the leaves of the tree and in the decision models, capturing the nonlinearities of the data via the structure of the decision tree. It is also possible, however, to make use of more complex nonparametric estimators such as neural networks as either leaf models or decision models. Also, a bridge between decision trees and multivariate statistical analysis can be created by utilizing models such as principal components analysis and canonical correlation as local models at the leaves of the tree.[4]

There is somewhat more constraint on the choice of models for the decisions at the nonterminals of the tree. Consider the density $P(\omega_i|\boldsymbol{x},\boldsymbol{\eta})$. Given that ω_i is a categorical random variable, this density is essentially a posterior probability for a multiway classification problem. Any of the standard parametric forms for multiway classification can be utilized to model this density. One approach is to utilize the multinomial logit model, a member of the generalized linear model (GLIM) family (McCullagh & Nelder, 1983). It can be shown that the likelihood for this model is a cross entropy of the form in Equation 2 (see Jordan & Jacobs, 1994), making this model a natural choice for modeling decisions in decision trees. Another approach is to invert the posterior $P(\omega_i|\boldsymbol{x})$ using Bayes' rule, and to model the class-conditional densities $P(\boldsymbol{x}|\omega_i)$ instead (Duda & Hart, 1973; Xu, Jordan, & Hinton, 1994).

With the choice of a multinomial logit model for the decisions the probabilistic decision tree is closely related to the CART (Breiman, et al., 1984) and C4.5 (Quinlan, 1993) models. In the multinomial logit model, the decision probabilities are functions of linear discriminants. In particular, the decision probabilities for the top level of the tree are as follows (Jordan & Jacobs, 1994):

$$P(\omega_i|\boldsymbol{x},\boldsymbol{\eta}) = \frac{e^{\boldsymbol{\eta}_i^T\boldsymbol{x}}}{\sum_j e^{\boldsymbol{\eta}_j^T\boldsymbol{x}}} \tag{4}$$

where $\boldsymbol{\eta} = \{\boldsymbol{\eta}_i\}_{i=1}^n$. Note that the inner product can be written as

$$\boldsymbol{\eta}_i^T\boldsymbol{x} = \|\boldsymbol{\eta}_i\|(\frac{\boldsymbol{\eta}_i}{\|\boldsymbol{\eta}_i\|})^T\boldsymbol{x}$$

showing that the probability depends both on the distance of the input vector from the plane orthogonal to the unit vector $\boldsymbol{\eta}_i/\|\boldsymbol{\eta}_i\|$ and on the magnitude of $\boldsymbol{\eta}_i$, which acts like an inverse temperature, scaling the decision probability. In the limit of parameter vectors of large magnitude, the split becomes a sharp linear decision boundary, and the decision tree reduces to a nested sequence of sharp splits as in the CART and C4.5 models. For parameter vectors of smaller magnitude, these splits are softened, allowing data points to contribute to parameter estimates across the splits. As discussed in the introduction, this local averaging provides control over the variance of the tree estimator.

5 EFFICIENCY

To assess the efficiency of the EM approach to parameter estimation in decision trees, Jordan and Jacobs studied a large scale regression problem involving 12 input variables and 15,000 data points. The tree was a binary tree with linear decision boundaries and linear leaf models. The tree had four levels and a total of 1,012 parameters. Figure 6 shows the convergence of the relative error on the test set as a function of the iterations through the training set. As can be seen, the algorithm converges rapidly, approximately two orders of magnitude faster than backpropagation in a multilayer perceptron with a comparable number of parameters.[5]

Jordan and Xu (1993) have studied the convergence of the EM algorithm for probabilistic decision trees and have shown that the algorithm converges according to a geometric series, with a rate that is a function of the condition number of the matrix $(P^{\frac{1}{2}})^T H(\boldsymbol{\phi})(P^{\frac{1}{2}})$, where P is a covariance matrix and H is the Hessian of the log likelihood.

[4] This involves using the decision tree as an unconditional density estimator rather than a conditional density estimator. See Jordan & Jacobs (in press).

[5] The final relative errors were .09 for the multilayer perceptron, .10 for HME, and .13 for CART.

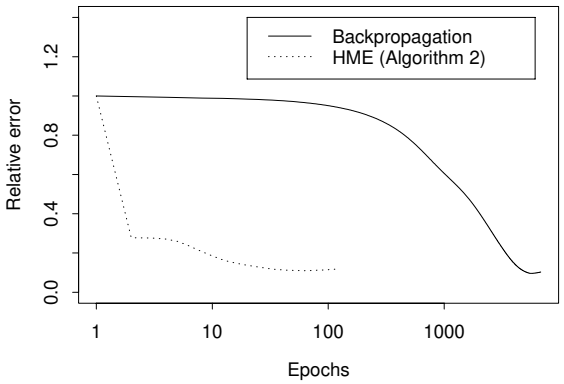

Figure 6: Relative error on the test set for a backpropagation network and a four-level HME architecture trained with batch algorithms. (Copied with permission from Jordan and Jacobs, 1994).

6 MODEL SELECTION

The problem of parameter estimation is the problem of choosing "best" parameter values given a particular fixed choice of a parametric model. General principles such as the maximum likelihood principle and the maximum a posteriori principle can be invoked to solve parameter estimation problems, and the problem reduces to the computational problem of finding an algorithm that searches efficiently for the appropriate optima. The problem of model selection is that of choosing between parameterized models. It is substantially more difficult than parameter estimation, and in most cases requires heuristic and sub-optimal methods.

The model selection problem for probabilistic decision trees involves the choice of the number and arrangement of the decision nodes in the tree. One approach to dealing with the problem is to make use of the pruning methods that have been developed for decision trees algorithms such as CART and C4.5. These methods involve heuristics for penalizing complex subtrees and for correcting for overfitting bias. An even more direct approach is to make use of the final tree from CART or C4.5 as an initialization step for the EM algorithm. The initial parameter vectors $\{\boldsymbol{\eta}, \boldsymbol{\nu}_i, \ldots\}$ are obtained in a straightforward manner from the hyperplanes used to define decisions in the CART or C4.5 trees, where the initial length of the parameter vectors is a free parameter. The initial parameter vectors $\boldsymbol{\theta}_{ij\ldots k}$ can either be copied directly from the tree (e.g., in the case of regression trees) or the training data at each leaf node of the CART or C4.5 tree can be used in an initial step to train a soft classifier at the corresponding leaf node of the probabilistic tree. This approach takes advantage of the discrete nature of the classical decision tree algorithms to provide an efficient search among alternative trees, and takes advantage of the continuous nature of the probabilistic approach to yield an improved "soft" estimator.[6] The approach is reminiscent of a widely used technique in the clustering domain: the K-means algorithm is used to initialize subsequent iterations of an EM clustering algorithm.

A difference between classical decision tree algorithms and the statistical approach is that in the latter the splits are not all-or-none, but instead are modulated continuously from a sharp split to no split at all. As the magnitudes of the $\{\boldsymbol{\eta}, \boldsymbol{\nu}_i, \ldots\}$ variables go to zero, the splits vanish.[7] This implies that the number of degrees of freedom in a fixed probabilistic decision tree is not the same as the "effective" number of degrees of freedom of the tree. When all splits vanish, the entire decision tree is effectively equivalent to a single leaf node. Because of the empirical fact that the fitting of these trees tends to proceed from coarse to fine (i.e., parameter magnitudes tend to grow fastest higher up in the tree), a fixed architecture effectively grows in depth during the iterations of the EM algorithm.

This effective growth in complexity motivates a number of other approaches to the model selection problem. One approach involves using ridge regression for the decision models and the leaf models. Ridge regression "shrinks" the parameters of a regression toward zero, providing control over the complexity of the model (Draper & Smith, 1981). This shrinkage can be viewed as a form of pruning; indeed, the ridge regression approach can be used in conjunction with a conservatively pruned CART or C4.5 tree. Another approach is to utilize a cross-validation procedure to stop the iterative process. This early stopping controls the effective depth of the tree.

To date our experience with model selection procedures has been limited to the CART-based initialization approach, the ridge regression approach and cross-validation stopping. We have not found large differences between these approaches to date. Other model selection methods worth investigating include Bayesian methods as well as the minimum description length approach, making use of Quinlan and Rivest's (1989) work on the coding of decision trees.

7 HIDDEN MARKOV DECISION TREES

An advantage of formulating a probability model for decision trees lies in the possibility of combining the decision tree methodology with other probabilistic estimation methods. One interesting hybrid involves the combination of decision trees with Hidden Markov Models.

Let us assume that each decision in the decision tree is dependent not only on the current input vector $\boldsymbol{x}$, but also on the previous decision at that node of the tree. For example, the probabilities at the top level of the tree are given as follows

$$P(\omega_i(t)|\omega_i(t-1), \boldsymbol{x}(t), \boldsymbol{\eta})$$

Each decision in the tree is obtained in this manner, yielding an architecture (see Figure 7) in which the decisions probabil-

[6] In a "hard" decision tree, all decision hyperplanes that split the data between the coordinates of a neighboring pair of data points are equivalent, because all such hyperplanes yield the same splits of the training set. Thus only a finite number of splits need to be considered (Breiman, et al., 1984). In a "soft" decision tree these splits are no longer equivalent because they lead to different posterior weights.

[7] That is, Equation 4 yields equal probabilities on either side of the fictitious split.

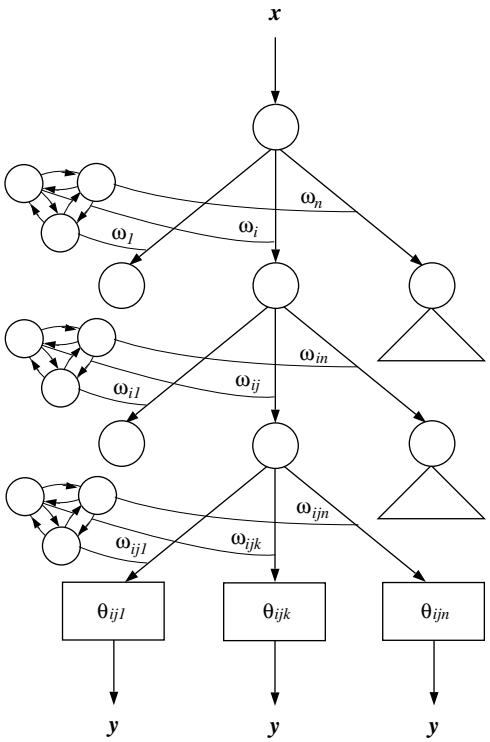

Figure 7: A hidden Markov decision tree. Each decision in the tree is conditional on the state of a local Markov chain.

ities are the modeled as the transition probabilities in each of a set of local Markov chains. Given the current states and the transition probabilities of each of these Markov chains, the output y at the next moment in time is determined according to the usual hierarchical mixture model (cf. Equation 1).[8]

An EM algorithm can be developed for maximum likelihood parameter estimation in the hidden Markov decision tree. In this case the significant structural components of the model are the Markov states, and the E step of the algorithm involves assigning posterior probability to each state at each moment in time. This assignment process involves an algorithm analogous to Baum's forward-backward algorithm to assign credit in time, combined with an upward-downward algorithm to assign posterior credit within the tree at each moment in time.

8 DISCUSSION

The probabilistic approach provides a general framework within which a number of additional issues of interest to decision tree researchers can be addressed. One example is the missing data problem (Quinlan, 1993). This problem can be addressed readily via the EM algorithm, which was originally developed as much to handle missing data problems as to handle mixture problems (Dempster, et al., 1977). EM allows missing data problems to be handled in a natural way—the missing data are treated as *actual* missing values in a manner analogous to the treatment of the *virtual* hidden values described above. The E step of EM fills in these missing values according to the values of the known variables and the assumed probabilistic structure of the model. Ghahramani and Jordan (1994) discuss this problem further in the context of a flat mixture model.

Another issue that can be addressed within the probabilistic framework is the problem of data that do not fit naturally into either a regression framework or a classification framework. Examples include data that are counts, or data that are time intervals (e.g., the time until failure). Such data can be treated within the decision tree framework by allowing more general models at the leaves of the tree. In particular, the class of generalized linear models (GLIM's) includes models that are appropriate for counts and time intervals. These models utilize densities from the exponential family other than the Gaussian or the multinomial. They fit cleanly within the EM framework discussed in this paper; in particular, the IRLS algorithm is available for the M-step of EM.

A topic of recent interest in the decision tree literature (Murthy, Kasif & Salzberg, 1993; Utgoff & Brodley, 1990) involves the use of decision hyperplanes at oblique angles to the axes. As discussed earlier, oblique decision hyperplanes arise in the probabilistic approach as the parametric component of multinomial logit models. Our empirical results, as well as our theoretical convergence results, suggest that the EM fitting of these hyperplanes does not greatly slow the convergence time. One reason for this is the quadratic convergence of the IRLS algorithm for multinomial logit models.

Finally, the EM approach can be extended to the case of on-line algorithms. Jordan and Jacobs (1994) present an approximate on-line EM algorithm for decision trees based on Kalman filtering ideas, and Neal and Hinton (1993) present a general discussion of on-line EM.

[8] Similar models have been considered for flat mixtures by Bengio and Frasconi (in press), Cacciatore and Nowlan (in press) and Meila and Jordan (1994).

Acknowledgments

This project was supported in part by a grant from the McDonnell-Pew Foundation, by a grant from ATR Human Information Processing Research Laboratories, by a grant from Siemens Corporation, and by grant N00014-90-J-1942 from the Office of Naval Research. The project was also supported by NSF grant ASC-9217041 in support of the Center for Biological and Computational Learning at MIT, including funds provided by DARPA under the HPCC program. Michael I. Jordan is a NSF Presidential Young Investigator.

References

Baum, L.E., Petrie, T., Soules, G., & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics, 41*, 164-171.

Bengio, Y., & Frasconi, P. (in press). Credit assignment through time: Alternatives to backpropagation. *Neural Information Processing Systems 6*. San Mateo, CA: Morgan Kaufmann.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group.

Cacciatore, T. & Nowlan, S. (in press). Mixtures of controllers for jump linear and non-linear plants. *Neural Information Processing Systems 6*. San Mateo, CA: Morgan Kaufmann.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, *39*, 1-38.

Draper, N. R., & Smith, H. (1981). *Applied Regression Analysis*. New York: John Wiley.

Duda, R. O., & Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. New York: John Wiley.

Ghahramani, Z., & Jordan, M. I. (in press). Supervised learning from incomplete data via the EM approach. *Neural Information Processing Systems 6*. San Mateo, CA: Morgan Kaufmann.

Jordan, M. I. & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, *6*, 181-214.

Jordan, M. I., & Xu, L. (1993). *Convergence properties of the EM approach to learning in mixture-of-experts architectures*. MIT Artificial Intelligence Laboratory Tech. Rep. 1458, Cambridge, MA.

McCullagh, P. & Nelder, J.A. (1983). *Generalized Linear Models*. London: Chapman and Hall.

Meila, M. P., & Jordan, M. I. (1994). *Learning the parameters of HMMs with auxiliary input*. MIT Computational Cognitive Science Tech. Rep. 9401, Cambridge, MA.

Murthy, S. K., Kasif, S., & Salzberg, S. (1993). *OC1: A randomized algorithm for building oblique decision trees*. Technical Report, Department of Computer Science, Johns Hopkins University.

Neal, R., & Hinton, G. E. (1993). A new view of the EM algorithm that justifies incremental and other variants. Submitted to *Biometrika*.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Quinlan, J. R., & Rivest, R. L. (1989). Inferring decision trees using the Minimum Description Length Principle. *Information and Computation, 80*, 227-248.

Utgoff, P. E., & Brodley, C. E. (1990). An incremental method for finding multivariate splits for decision trees. In *Proceedings of the Seventh International Conference on Machine Learning*, Los Altos, CA.

Xu, L., Jordan, M. I., & Hinton, G. E. (1994). *An alternative mixture of experts model*. MIT Computational Cognitive Science Tech. Rep. 9402, Cambridge, MA.