Hierarchical mixtures of experts and the

Michael Jordan
University of California, Berkeley
1,002 PUBLICATIONS **241,334** CITATIONS

SEE PROFILE

Robert Jacobs
University of Rochester
137 PUBLICATIONS **18,616** CITATIONS

SEE PROFILE

ARTICLE Communicated by Steven Nowlan

Hierarchical Mixtures of Experts and the EM Algorithm

Michael I. Jordan
Department of Brain and Cognitive Sciences,
Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Robert A. Jacobs
Department of Psychology, University of Rochester,
Rochester, NY 14627 USA

We present a tree-structured architecture for supervised learning. The statistical model underlying the architecture is a hierarchical mixture model in which both the mixture coefficients and the mixture components are generalized linear models (GLIM's). Learning is treated as a maximum likelihood problem; in particular, we present an Expectation–Maximization (EM) algorithm for adjusting the parameters of the architecture. We also develop an on-line learning algorithm in which the parameters are updated incrementally. Comparative simulation results are presented in the robot dynamics domain.

1 Introduction

The principle of divide-and-conquer is a principle with wide applicability throughout applied mathematics. Divide-and-conquer algorithms attack a complex problem by dividing it into simpler problems whose solutions can be combined to yield a solution to the complex problem. This approach can often lead to simple, elegant, and efficient algorithms. In this paper we explore a particular application of the divide-and-conquer principle to the problem of learning from examples. We describe a network architecture and a learning algorithm for the architecture, both of which are inspired by the philosophy of divide-and-conquer.

In the statistical literature and in the machine learning literature, divide-and-conquer approaches have become increasingly popular. The CART algorithm of Breiman *et al.* (1984), the MARS algorithm of Friedman (1991), and the ID3 algorithm of Quinlan (1986) are well-known examples. These algorithms fit surfaces to data by explicitly dividing the input space into a nested sequence of regions, and by fitting simple surfaces (e.g., constant functions) within these regions. They have convergence times that are often orders of magnitude faster than gradient-based neural network algorithms.

Neural Computation **6, 181–214** (1994) © 1994 Massachusetts Institute of Technology

Although divide-and-conquer algorithms have much to recommend them, one should be concerned about the statistical consequences of dividing the input space. Dividing the data can have favorable consequences for the bias of an estimator, but it generally increases the variance. Consider linear regression, for example, in which the variance of the estimates of the slope and intercept depends quadratically on the spread of data on the x-axis. The points that are the most peripheral in the input space are those that have the maximal effect in decreasing the variance of the parameter estimates.

The foregoing considerations suggest that divide-and-conquer algorithms generally tend to be variance-increasing algorithms. This is indeed the case and is particularly problematic in high-dimensional spaces where data become exceedingly sparse (Scott 1992). One response to this dilemma—that adopted by CART, MARS, and ID3, and also adopted here—is to utilize piecewise constant or piecewise linear functions. These functions minimize variance at a cost of increased bias. We also make use of a second variance-decreasing device; a device familiar in the neural network literature. We make use of "soft" splits of data (Bridle 1989; Nowlan 1991; Wahba *et al.* 1993), allowing data to lie simultaneously in multiple regions. This approach allows the parameters in one region to be influenced by data in neighboring regions. CART, MARS, and ID3 rely on "hard" splits, which, as we remarked above, have particularly severe effects on variance. By allowing soft splits the severe effects of lopping off distant data can be ameliorated. We also attempt to minimize the bias that is incurred by using piecewise linear functions, by allowing the splits to be formed along hyperplanes at arbitrary orientations in the input space. This lessens the bias due to high-order interactions among the inputs and allows the algorithm to be insensitive to the particular choice of coordinates used to encode the data (an improvement over methods such as MARS and ID3, which are coordinate-dependent).

The work that we describe here makes contact with a number of branches of statistical theory. First, as in our earlier work (Jacobs *et al.* 1991), we formulate the learning problem as a mixture estimation problem (cf. Cheeseman *et al.* 1988; Duda and Hart 1973; Nowlan 1991; Redner and Walker 1984; Titterington *et al.* 1985). We show that the algorithm that is generally employed for the *unsupervised* learning of mixture parameters—the Expectation–Maximization (EM) algorithm of Dempster *et al.* (1977)—can also be exploited for *supervised* learning. Second, we utilize generalized linear model (GLIM) theory (McCullagh and Nelder 1983) to provide the basic statistical structure for the components of the architecture. In particular, the "soft splits" referred to above are modeled as *multinomial logit* models—a specific form of GLIM. We also show that the algorithm developed for fitting GLIMs—the iteratively reweighted least squares (IRLS) algorithm—can be usefully employed in our model, in particular as the M step of the EM algorithm. Finally, we show that these ideas can be developed in a recursive manner, yielding a tree-structured approach to estimation that is reminiscent of CART, MARS, and ID3.

The remainder of the paper proceeds as follows. We first introduce the hierarchical mixture-of-experts architecture and present the likelihood function for the architecture. After describing a gradient descent algorithm, we develop a more powerful learning algorithm for the architecture that is a special case of the general Expectation–Maximization (EM) framework of Dempster *et al.* (1977). We also describe a least-squares version of this algorithm that leads to a particularly efficient implementation. Both of the latter algorithms are batch learning algorithms. In the final section, we present an on-line version of the least-squares algorithm that in practice appears to be the most efficient of the algorithms that we have studied.

2 Hierarchical Mixtures of Experts

The algorithms that we discuss in this paper are supervised learning algorithms. We explicitly address the case of regression, in which the input vectors are elements of $\Re^m$ and the output vectors are elements of $\Re^n$. We also consider classification models and counting models in which the outputs are integer-valued. The data are assumed to form a countable set of paired observations $\mathcal{X} = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}$. In the case of the *batch* algorithms discussed below, this set is assumed to be finite; in the case of the *on-line* algorithms, the set may be infinite.

We propose to solve nonlinear supervised learning problems by dividing the input space into a nested set of regions and fitting simple surfaces to the data that fall in these regions. The regions have "soft" boundaries, meaning that data points may lie simultaneously in multiple regions. The boundaries between regions are themselves simple parameterized surfaces that are adjusted by the learning algorithm.

The hierarchical mixture-of-experts (HME) architecture is shown in Figure 1.[1] The architecture is a tree in which the *gating networks* sit at the nonterminals of the tree. These networks receive the vector $\mathbf{x}$ as input and produce scalar outputs that are a partition of unity at each point in the input space. The *expert networks* sit at the leaves of the tree. Each expert produces an output vector $\boldsymbol{\mu}_{ij}$ for each input vector. These output vectors proceed up the tree, being blended by the gating network outputs.

All of the expert networks in the tree are linear with a single output nonlinearity. We will refer to such a network as "generalized linear," borrowing the terminology from statistics (McCullagh and Nelder 1983).

[1] To simplify the presentation, we restrict ourselves to a two-level hierarchy throughout the paper. All of the algorithms that we describe, however, generalize readily to hierarchies of arbitrary depth. See Jordan and Xu (1993) for a recursive formalism that handles arbitrary hierarchies.

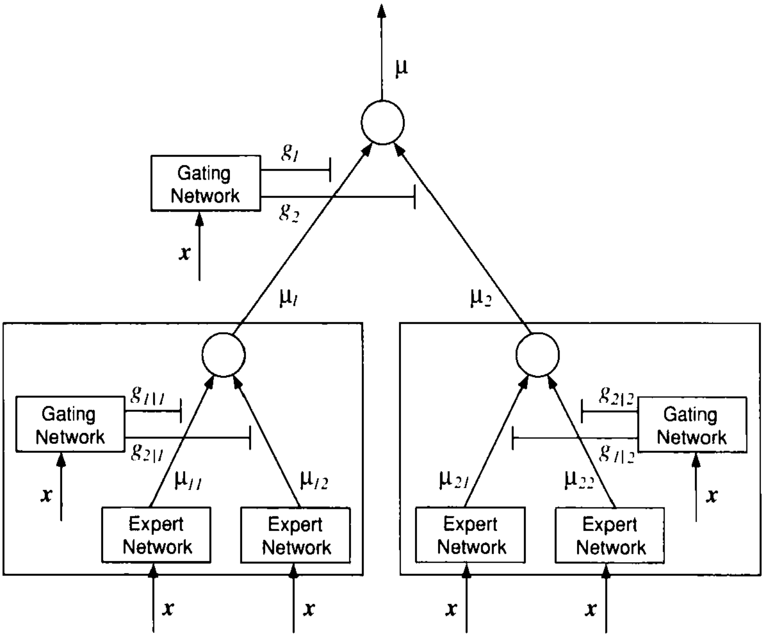

Figure 1: A two-level hierarchical mixture of experts. To form a deeper tree, each expert is expanded recursively into a gating network and a set of subexperts.

Expert network (i, j) produces its output μ_{ij} as a generalized linear function of the input **x**:

$$\mu_{ij} = f(U_{ij}\mathbf{x}) \tag{2.1}$$

where U_{ij} is a weight matrix and f is a fixed continuous nonlinearity. The vector **x** is assumed to include a fixed component of one to allow for an intercept term.

For regression problems, $f(\cdot)$ is generally chosen to be the identity function (i.e., the experts are linear). For binary classification problems, $f(\cdot)$ is generally taken to be the logistic function, in which case the expert outputs are interpreted as the log odds of "success" under a Bernoulli probability model (see below). Other models (e.g., multiway classification, counting, rate estimation, and survival estimation) are handled by making other choices for $f(\cdot)$. These models are smoothed piecewise analogs of the corresponding GLIM models (cf. McCullagh and Nelder 1983).

The gating networks are also generalized linear. Define intermediate variables ξ_i as follows:

$$\xi_i = \mathbf{v}_i^T \mathbf{x} \tag{2.2}$$

where $\mathbf{v}_i$ is a weight vector. Then the ith output of the top-level gating network is the "softmax" function of the ξ_i (Bridle 1989; McCullagh and Nelder 1983):

$$g_i = \frac{e^{\xi_i}}{\sum_k e^{\xi_k}} \tag{2.3}$$

Note that the g_i are positive and sum to one for each $\mathbf{x}$. They can be interpreted as providing a "soft" partitioning of the input space.

Similarly, the gating networks at lower levels are also generalized linear systems. Define ξ_{ij} as follows:

$$\xi_{ij} = \mathbf{v}_{ij}^T \mathbf{x} \tag{2.4}$$

Then

$$g_{j|i} = \frac{e^{\xi_{ij}}}{\sum_k e^{\xi_{ik}}} \tag{2.5}$$

is the output of the jth unit in the ith gating network at the second level of the architecture. Once again, the $g_{j|i}$ are positive and sum to one for each $\mathbf{x}$. They can be interpreted as providing a "soft" sub-partition of the input space nested within the partitioning providing by the higher-level gating network.

The output vector at each nonterminal of the tree is the weighted output of the experts below that nonterminal. That is, the output at the ith nonterminal in the second layer of the two-level tree is

$$\boldsymbol{\mu}_i = \sum_j g_{j|i} \boldsymbol{\mu}_{ij}$$

and the output at the top level of the tree is

$$\boldsymbol{\mu} = \sum_i g_i \boldsymbol{\mu}_i$$

Note that both the g's and the μ's depend on the input $\mathbf{x}$, thus the total output is a nonlinear function of the input.

2.1 Regression Surface. Given the definitions of the expert networks and the gating networks, the regression surface defined by the hierarchy is a piecewise blend of the regression surfaces defined by the experts. The gating networks provide a nested, "soft" partitioning of the input space and the expert networks provide local regression surfaces within the partition. There is overlap between neighboring regions. To understand the nature of the overlap, consider a one-level hierarchy with two expert networks. In this case, the gating network has two outputs, g_1 and g_2. The gating output g_1 is given by

$$g_1 = \frac{e^{\xi_1}}{e^{\xi_1} + e^{\xi_2}} \tag{2.6}$$

$$= \frac{1}{1 + e^{-(\mathbf{v}_1 - \mathbf{v}_2)^T \mathbf{x}}} \tag{2.7}$$

which is a logistic ridge function whose orientation is determined by the direction of the vector $\mathbf{v}_1 - \mathbf{v}_2$. The gating output g_2 is equal to $1 - g_1$. For a given $\mathbf{x}$, the total output μ is the convex combination $g_1\mu_1 + g_2\mu_2$. This is a weighted average of the experts, where the weights are determined by the values of the ridge function. Along the ridge, $g_1 = g_2 = 1/2$, and both experts contribute equally. Away from the ridge, one expert or the other dominates. The amount of smoothing across the ridge is determined by the magnitude of the vector $\mathbf{v}_2 - \mathbf{v}_1$. If $\mathbf{v}_2 - \mathbf{v}_1$ is large, then the ridge function becomes a sharp split and the weighted output of the experts becomes piecewise (generalized) linear. If $\mathbf{v}_2 - \mathbf{v}_1$ is small, then each expert contributes to a significant degree on each side of the ridge, thereby smoothing the piecewise map. In the limit of a zero difference vector, $g_1 = g_2 = 1/2$ for all $\mathbf{x}$, and the total output is the same fixed average of the experts on both sides of the fictitious "split."

In general, a given gating network induces a smoothed planar partitioning of the input space. Lower-level gating networks induce a partition within the partition induced by higher-level gating networks. The weights in a given gating network determine the amount of smoothing across the partition at that particular level of resolution: large weight vectors imply sharp changes in the regression surface across a ridge and small weights imply a smoother surface. In the limit of zero weights in all gating networks, the entire hierarchy reduces to a fixed average (a linear system in the case of regression).

2.2 A Probability Model. The hierarchy can be given a probabilistic interpretation. We suppose that the mechanism by which data are generated by the environment involves a nested sequence of decisions that terminates in a regressive process that maps $\mathbf{x}$ to $\mathbf{y}$. The decisions are modeled as multinomial random variables. That is, for each $\mathbf{x}$, we interpret the values $g_i(\mathbf{x}, \mathbf{v}_i^0)$ as the multinomial probabilities associated with the first decision and the $g_{j|i}(\mathbf{x}, \mathbf{v}_{ij}^0)$ as the (conditional) multinomial probabilities associated with the second decision, where the superscript "0" refers to the "true" values of the parameters. The decisions form a decision tree. We use a statistical model to model this decision tree; in particular, our choice of parameterization (cf. Equations 2.2, 2.3, 2.4, and 2.5) corresponds to a *multinomial logit* probability model at each nonterminal of the tree (see Appendix B). A multinomial logit model is a special case of a GLIM that is commonly used for "soft" multiway classification (McCullagh and Nelder 1983). Under the multinomial logit model, we interpret the gating networks as modeling the input-dependent, multinomial probabilities associated with decisions at particular levels of resolution in a tree-structured model of the data.

Once a particular sequence of decisions has been made, resulting in a choice of regressive process (i, j), output $\mathbf{y}$ is assumed to be generated according to the following statistical model. First, a linear predictor η_{ij} is formed:

$$\eta_{ij}^0 = U_{ij}^0 \mathbf{x}$$

The expected value of $\mathbf{y}$ is obtained by passing the linear predictor through the *link function* f:[2]

$$\mu_{ij}^0 = f(\eta_{ij}^0)$$

The output $\mathbf{y}$ is then chosen from a probability density P, with mean μ_{ij}^0 and "dispersion" parameter ϕ_{ij}^0. We denote the density of $\mathbf{y}$ as

$$P(\mathbf{y}|\mathbf{x}, \theta_{ij}^0)$$

where the parameter vector θ_{ij}^0 includes the weights U_{ij}^0 and the dispersion parameter ϕ_{ij}^0:

$$\theta_{ij}^0 = \begin{bmatrix} U_{ij}^0 \\ \phi_{ij}^0 \end{bmatrix}$$

We assume the density P to be a member of the exponential family of densities (McCullagh and Nelder 1983). The interpretation of the dispersion parameter depends on the particular choice of density. For example, in the case of the n-dimensional gaussian, the dispersion parameter is the covariance matrix Σ_{ij}^0.[3]

Given these assumptions, the total probability of generating $\mathbf{y}$ from $\mathbf{x}$ is the mixture of the probabilities of generating $\mathbf{y}$ from each of the component densities, where the mixing proportions are multinomial probabilities:

$$P(\mathbf{y}|\mathbf{x}, \theta^0) = \sum_i g_i(\mathbf{x}, \mathbf{v}_i^0) \sum_j g_{j|i}(\mathbf{x}, \mathbf{v}_{ij}^0) P(\mathbf{y}|\mathbf{x}, \theta_{ij}^0) \tag{2.8}$$

Note that θ^0 includes the expert network parameters θ_{ij}^0 as well as the gating network parameters $\mathbf{v}_i^0$ and $\mathbf{v}_{ij}^0$. Note also that we have explicitly

[2] We utilize the neural network convention in defining links. In GLIM theory, the convention is that the link function relates η to μ; thus, $\eta = h(\mu)$, where h is equivalent to our f^{-1}.

[3] Not all exponential family densities have a dispersion parameter; in particular, the Bernoulli density discussed below has no dispersion parameter.

indicated the dependence of the probabilities g_i and $g_{j|i}$ on the input **x** and on the parameters. In the remainder of the paper we drop the explicit reference to the input and the parameters to simplify the notation:

$$P(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}^0) = \sum_i g_i^0 \sum_j g_{j|i}^0 P_{ij}^0(\mathbf{y}) \tag{2.9}$$

We also utilize equation 2.9 without the superscripts to refer to the probability model defined by a particular HME architecture, irrespective of any reference to a "true" model.

2.2.1 *Example (Regression).* In the case of regression the probabilistic component of the model is generally assumed to be gaussian. Assuming identical covariance matrices of the form $\sigma^2 I$ for each of the experts yields the following hierarchical probability model:

$$P(\mathbf{y}|\mathbf{x},\boldsymbol{\theta}) = \frac{1}{(2\pi)^{n/2}\sigma^n} \sum_i g_i \sum_j g_{j|i} e^{-(1/2\sigma^2)(\mathbf{y}-\boldsymbol{\mu}_{ij})^T(\mathbf{y}-\boldsymbol{\mu}_{ij})}$$

2.2.2 *Example (Binary Classification).* In binary classification problems the output y is a discrete random variable having possible outcomes of "failure" and "success." The probabilistic component of the model is generally assumed to be the Bernoulli distribution (Cox 1970). In this case, the mean μ_{ij} is the conditional probability of classifying the input as "success." The resulting hierarchical probability model is a mixture of Bernoulli densities:

$$P(y|\mathbf{x},\boldsymbol{\theta}) = \sum_i g_i \sum_j g_{j|i} \mu_{ij}^y (1-\mu_{ij})^{1-y}$$

2.3 Posterior Probabilities. In developing the learning algorithms to be presented in the remainder of the paper, it will prove useful to define posterior probabilities associated with the nodes of the tree. The terms "posterior" and "prior" have meaning in this context during the training of the system. We refer to the probabilities g_i and $g_{j|i}$ as *prior* probabilities, because they are computed based only on the input **x**, without knowledge of the corresponding target output **y**. A *posterior* probability is defined once both the input and the target output are known. Using Bayes' rule, we define the posterior probabilities at the nodes of the tree as follows:

$$h_i = \frac{g_i \sum_j g_{j|i} P_{ij}(\mathbf{y})}{\sum_i g_i \sum_j g_{j|i} P_{ij}(\mathbf{y})} \tag{2.10}$$

and

$$h_{j|i} = \frac{g_{j|i} P_{ij}(\mathbf{y})}{\sum_j g_{j|i} P_{ij}(\mathbf{y})} \tag{2.11}$$

We will also find it useful to define the joint posterior probability h_{ij}, the product of h_i and $h_{j|i}$:

$$h_{ij} = \frac{g_i g_{j|i} P_{ij}(\mathbf{y})}{\sum_i g_i \sum_j g_{j|i} P_{ij}(\mathbf{y})} \tag{2.12}$$

This quantity is the probability that expert network (i, j) can be considered to have generated the data, based on knowledge of both the input and the output. Once again, we emphasize that all of these quantities are conditional on the input **x**.

In deeper trees, the posterior probability associated with an expert network is simply the product of the conditional posterior probabilities along the path from the root of the tree to that expert.

2.4 The Likelihood and a Gradient Ascent Learning Algorithm. Jordan and Jacobs (1992) presented a gradient ascent learning algorithm for the hierarchical architecture. The algorithm was based on earlier work by Jacobs *et al.* (1991), who treated the problem of learning in mixture-of-experts architectures as a maximum likelihood estimation problem. The log likelihood of a data set $\mathcal{X} = \{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}_1^N$ is obtained by taking the log of the product of N densities of the form of equation 2.9, which yields the following log likelihood:

$$l(\boldsymbol{\theta}; \mathcal{X}) = \sum_t \ln \sum_i g_i^{(t)} \sum_j g_{j|i}^{(t)} P_{ij}(\mathbf{y}^{(t)}) \tag{2.13}$$

Let us assume that the probability density P is gaussian with an identity covariance matrix and that the link function is the identity. In this case, by differentiating $l(\boldsymbol{\theta}; \mathcal{X})$ with respect to the parameters, we obtain the following gradient ascent learning rule for the weight matrix U_{ij}:

$$\Delta U_{ij} = \rho \sum_t h_i^{(t)} h_{j|i}^{(t)} (\mathbf{y}^{(t)} - \boldsymbol{\mu}^{(t)}) \mathbf{x}^{(t)T} \tag{2.14}$$

where ρ is a learning rate. The gradient ascent learning rule for the ith weight vector in the top-level gating network is given by

$$\Delta \mathbf{v}_i = \rho \sum_t (h_i^{(t)} - g_i^{(t)}) \mathbf{x}^{(t)} \tag{2.15}$$

and the gradient ascent rule for the jth weight vector in the ith lower-level gating network is given by

$$\Delta \mathbf{v}_{ij} = \rho \sum_t h_i^{(t)} (h_{j|i}^{(t)} - g_{j|i}^{(t)}) \mathbf{x}^{(t)} \tag{2.16}$$

Updates can also be obtained for covariance matrices (Jordan and Jacobs 1992).

The algorithm given by equations 2.14, 2.15, and 2.16 is a batch learning algorithm. The corresponding on-line algorithm is obtained by simply dropping the summation sign and updating the parameters after each stimulus presentation. Thus, for example,

$$U_{ij}^{(t+1)} = U_{ij}^{(t)} + \rho h_i^{(t)} h_{j|i}^{(t)} (\mathbf{y}^{(t)} - \boldsymbol{\mu}^{(t)}) \mathbf{x}^{(t)T} \tag{2.17}$$

is the stochastic update rule for the weights in the (i, j)th expert network based on the tth stimulus pattern.

2.5 The EM Algorithm. In the following sections we develop a learning algorithm for the HME architecture based on the Expectation–Maximization (EM) framework of Dempster *et al.* (1977). We derive an EM algorithm for the architecture that consists of the iterative solution of a coupled set of iteratively-reweighted least-squares problems.

The EM algorithm is a general technique for maximum likelihood estimation. In practice EM has been applied almost exclusively to unsupervised learning problems. This is true of the neural network literature and machine learning literature, in which EM has appeared in the context of clustering (Cheeseman *et al.* 1988; Nowlan 1991) and density estimation (Specht 1991), as well as the statistics literature, in which applications include missing data problems (Little and Rubin 1987), mixture density estimation (Redner and Walker 1984), and factor analysis (Dempster *et al.* 1977). Another unsupervised learning application is the learning problem for Hidden Markov Models, for which the Baum–Welch reestimation formulas are a special case of EM. There is nothing in the EM framework that precludes its application to regression or classification problems; however, such applications have been few.[4]

EM is an iterative approach to maximum likelihood estimation. Each iteration of an EM algorithm is composed of two steps: an Estimation (E) step and a Maximization (M) step. The M step involves the maximization of a likelihood function that is redefined in each iteration by the E step. If the algorithm simply increases the function during the M step, rather than maximizing the function, then the algorithm is referred to as a Generalized EM (GEM) algorithm. The Boltzmann learning algorithm (Hinton and Sejnowski 1986) is a neural network example of a GEM algorithm. GEM algorithms are often significantly slower to converge than EM algorithms.

An application of EM generally begins with the observation that the optimization of the likelihood function $l(\boldsymbol{\theta}; \mathcal{X})$ would be simplified if only a set of additional variables, called "missing" or "hidden" variables, were known. In this context, we refer to the observable data $\mathcal{X}$ as the "incomplete data" and posit a "complete data" set $\mathcal{Y}$ that includes the missing variables $\mathcal{Z}$. We specify a probability model that links the fictive missing variables to the actual data: $P(\mathbf{y}, \mathbf{z}|\mathbf{x}, \boldsymbol{\theta})$. The logarithm of the density P defines the "complete-data likelihood," $l_c(\boldsymbol{\theta}; \mathcal{Y})$. The original likelihood,

[4] An exception is the "switching regression" model of Quandt and Ramsey (1972). For further discussion of switching regression, see Jordan and Xu (1993).

$l(\boldsymbol{\theta}; \mathcal{X})$, is referred to in this context as the "incomplete-data likelihood." It is the relationship between these two likelihood functions that motivates the EM algorithm. Note that the complete-data likelihood is a random variable, because the missing variables $\mathcal{Z}$ are in fact unknown. An EM algorithm first finds the expected value of the complete-data likelihood, given the observed data and the current model. This is the E step:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(p)}) = E[l_c(\boldsymbol{\theta}; \mathcal{Y}) | \mathcal{X}]$$

where $\boldsymbol{\theta}^{(p)}$ is the value of the parameters at the pth iteration and the expectation is taken with respect to $\boldsymbol{\theta}^{(p)}$. This step yields a deterministic function Q. The M step maximizes this function with respect to $\boldsymbol{\theta}$ to find the new parameter estimates $\boldsymbol{\theta}^{(p+1)}$:

$$\boldsymbol{\theta}^{(p+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(p)})$$

The E step is then repeated to yield an improved estimate of the complete likelihood and the process iterates.

An iterative step of EM chooses a parameter value that increases the value of Q, the expectation of the complete likelihood. What is the effect of such a step on the incomplete likelihood? Dempster *et al.* proved that an increase in Q implies an increase in the incomplete likelihood:

$$l(\boldsymbol{\theta}^{(p+1)}; \mathcal{X}) \geq l(\boldsymbol{\theta}^{(p)}; \mathcal{X})$$

Equality obtains only at the stationary points of l (Wu 1983). Thus the likelihood l increases monotonically along the sequence of parameter estimates generated by an EM algorithm. In practice this implies convergence to a local maximum.

2.6 Applying EM to the HME Architecture. To develop an EM algorithm for the HME architecture, we must define appropriate "missing data" so as to simplify the likelihood function. We define indicator variables z_i and $z_{j|i}$, such that one and only one of the z_i is equal to one, and one and only one of the $z_{j|i}$ is equal to one. These indicator variables have an interpretation as the labels that correspond to the decisions in the probability model. We also define the indicator variable z_{ij}, which is the product of z_i and $z_{j|i}$. This variable has an interpretation as the label that specifies the expert (the regressive process) in the probability model. If the labels z_i, $z_{j|i}$, and z_{ij} were known, then the maximum likelihood problem would decouple into a separate set of regression problems for each expert network and a separate set of multiway classification problems for the gating networks. These problems would be solved independently of each other, yielding a rapid one-pass learning algorithm. Of course, the missing variables are not known, but we can specify a probability model that links them to the observable data. This probability model can be written in terms of the z_{ij} as follows:

$$
\begin{aligned}
P(\mathbf{y}^{(t)}, z_{ij}^{(t)} | \mathbf{x}^{(t)}, \boldsymbol{\theta}) &= g_i^{(t)} g_{j|i}^{(t)} P_{ij}(\mathbf{y}^{(t)}) && (2.18) \\
&= \prod_i \prod_j \{ g_i^{(t)} g_{j|i}^{(t)} P_{ij}(\mathbf{y}^{(t)}) \}^{z_{ij}^{(t)}} && (2.19)
\end{aligned}
$$

using the fact that $z_{ij}^{(t)}$ is an indicator variable. Taking the logarithm of this probability model yields the following complete-data likelihood:

$$
\begin{aligned}
l_c(\boldsymbol{\theta}; \mathcal{Y}) &= \sum_t \sum_i \sum_j z_{ij}^{(t)} \ln\{ g_i^{(t)} g_{j|i}^{(t)} P_{ij}(\mathbf{y}^{(t)}) \} && (2.20) \\
&= \sum_t \sum_i \sum_j z_{ij}^{(t)} \{ \ln g_i^{(t)} + \ln g_{j|i}^{(t)} + \ln P_{ij}(\mathbf{y}^{(t)}) \} && (2.21)
\end{aligned}
$$

Note the relationship of the complete-data likelihood in equation 2.21 to the incomplete-data likelihood in equation 2.13. The use of the indicator variables z_{ij} has allowed the logarithm to be brought inside the summation signs, substantially simplifying the maximization problem. We now define the E step of the EM algorithm by taking the expectation of the complete-data likelihood:

$$
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(p)}) = \sum_t \sum_i \sum_j h_{ij}^{(t)} \{ \ln g_i^{(t)} + \ln g_{j|i}^{(t)} + \ln P_{ij}(\mathbf{y}^{(t)}) \} \tag{2.22}
$$

where we have used the fact that

$$
\begin{aligned}
E[z_{ij}^{(t)} | \mathcal{X}] &= P(z_{ij}^{(t)} = 1 | \mathbf{y}^{(t)}, \mathbf{x}^{(t)}, \boldsymbol{\theta}^{(p)}) && (2.23) \\
&= \frac{P(\mathbf{y}^{(t)} | z_{ij}^{(t)} = 1, \mathbf{x}^{(t)}, \boldsymbol{\theta}^{(p)}) P(z_{ij}^{(t)} = 1 | \mathbf{x}^{(t)}, \boldsymbol{\theta}^{(p)})}{P(\mathbf{y}^{(t)} | \mathbf{x}^{(t)}, \boldsymbol{\theta}^{(p)})} && (2.24) \\
&= \frac{P(\mathbf{y}^{(t)} | \mathbf{x}^{(t)}, \boldsymbol{\theta}_{ij}{}^{(p)}) g_i^{(t)} g_{j|i}^{(t)}}{\sum_i g_i^{(t)} \sum_j g_{j|i}^{(t)} P(\mathbf{y}^{(t)} | \mathbf{x}^{(t)}, \boldsymbol{\theta}_{ij}{}^{(p)})} && (2.25) \\
&= h_{ij}^{(t)} && (2.26)
\end{aligned}
$$

(Note also that $E[z_i^{(t)} | \mathcal{X}] = h_i^{(t)}$ and $E[z_{j|i}^{(t)} | \mathcal{X}] = h_{j|i}^{(t)}$.)

The M step requires maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(p)})$ with respect to the expert network parameters and the gating network parameters. Examining equation 2.22, we see that the expert network parameters influence the Q function only through the terms $h_{ij}^{(t)} \ln P_{ij}(\mathbf{y}^{(t)})$, and the gating network parameters influence the Q function only through the terms $h_{ij}^{(t)} \ln g_i^{(t)}$ and $h_{ij}^{(t)} \ln g_{j|i}^{(t)}$. Thus the M step reduces to the following separate maximization problems:

$$
\begin{aligned}
\boldsymbol{\theta}_{ij}^{(p+1)} &= \arg\max_{\boldsymbol{\theta}_{ij}} \sum_t h_{ij}^{(t)} \ln P_{ij}(\mathbf{y}^{(t)}) && (2.27) \\
\mathbf{v}_i^{(p+1)} &= \arg\max_{\mathbf{v}_i} \sum_t \sum_k h_k^{(t)} \ln g_k^{(t)} && (2.28)
\end{aligned}
$$

and

$$\mathbf{v}_{ij}^{(p+1)} = \arg\max_{\mathbf{v}_{ij}} \sum_t \sum_k h_k^{(t)} \sum_l h_{l|k}^{(t)} \ln g_{l|k}^{(t)} \tag{2.29}$$

Each of these maximization problems is itself a maximum likelihood problem. This is clearly true in the case of equation 2.27, which is simply a weighted maximum likelihood problem in the probability density P_{ij}. Given our parameterization of P_{ij}, the log likelihood in equation 2.27 is a weighted log likelihood for a GLIM. An efficient algorithm known as iteratively reweighted least-squares (IRLS) is available to solve the maximum likelihood problem for such models (McCullagh and Nelder 1983). We discuss IRLS in Appendix A.

Equation 2.28 involves maximizing the cross-entropy between the posterior probabilities $h_k^{(t)}$ and the prior probabilities $g_k^{(t)}$. This cross-entropy is the log likelihood associated with a multinomial logit probability model in which the $h_k^{(t)}$ act as the output observations (see Appendix B). Thus the maximization in equation 2.28 is also a maximum likelihood problem for a GLIM and can be solved using IRLS. The same is true of equation 2.29, which is a weighted maximum likelihood problem with output observations $h_{l|k}^{(t)}$ and observation weights $h_k^{(t)}$.

In summary, the EM algorithm that we have obtained involves a calculation of posterior probabilities in the outer loop (the E step), and the solution of a set of IRLS problems in the inner loop (the M step). We summarize the algorithm as follows:

Algorithm 1

1. For each data pair $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$, compute the posterior probabilities $h_i^{(t)}$ and $h_{j|i}^{(t)}$ using the current values of the parameters.
2. For each expert (i, j), solve an IRLS problem with observations $\{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}_1^N$ and observation weights $\{h_{ij}^{(t)}\}_1^N$.
3. For each top-level gating network, solve an IRLS problem with observations $\{(\mathbf{x}^{(t)}, h_k^{(t)})\}_1^N$.
4. For each lower-level gating network, solve a weighted IRLS problem with observations $\{(\mathbf{x}^{(t)}, h_{l|k}^{(t)})\}_1^N$ and observation weights $\{h_k^{(t)}\}_1^N$.
5. Iterate using the updated parameter values.

2.7 A Least-Squares Algorithm. In the case of regression, in which a gaussian probability model and an identity link function are used, the IRLS loop for the expert networks reduces to weighted least squares, which can be solved (in one pass) by any of the standard least-squares algorithms (Golub and van Loan 1989). The gating networks still require iterative processing. Suppose, however, that we fit the parameters of the gating networks using least squares rather than maximum likelihood. In this case, we might hope to obtain an algorithm in which the gating network parameters are fit by a one-pass algorithm. To motivate this approach, note that we can express the IRLS problem for the gating networks as follows. Differentiating the cross-entropy (equation 2.28) with respect to the parameters $\mathbf{v}_i$ (using the fact that $\partial g_i / \partial \xi_j = g_i(\delta_{ij} - g_j)$, where δ_{ij} is the Kronecker delta) and setting the derivatives to zero yields the following equations:

$$\sum_t (h_i^{(t)} - g_i(\mathbf{x}^{(t)}, \mathbf{v}_i))\mathbf{x}^{(t)} = 0 \tag{2.30}$$

which are a coupled set of equations that must be solved for each i. Similarly, for each gating network at the second level of the tree, we obtain the following equations:

$$\sum_t h_i^{(t)} (h_{j|i}^{(t)} - g_{j|i}(\mathbf{x}^{(t)}, \mathbf{v}_{ij}))\mathbf{x}^{(t)} = 0 \tag{2.31}$$

which must be solved for each i and j. There is one aspect of these equations that renders them unusual. Recall that if the labels $z_i^{(t)}$ and $z_{j|i}^{(t)}$ were known, then the gating networks would be essentially solving a set of multiway classification problems. The supervised errors $(z_i^{(t)} - g_i^{(t)})$ and $(z_{j|i}^{(t)} - g_{j|i}^{(t)})$ would appear in the algorithm for solving these problems. Note that these errors are differences between indicator variables and probabilities. In equations 2.30 and 2.31, on the other hand, the errors that drive the algorithm are the differences $(h_i^{(t)} - g_i^{(t)})$ and $(h_{j|i}^{(t)} - g_{j|i}^{(t)})$, which are differences between probabilities. The EM algorithm effectively "fills in" the missing labels with estimated probabilities h_i and $h_{j|i}$. These estimated probabilities can be thought of as targets for the g_i and the $g_{j|i}$. This suggests that we can compute "virtual targets" for the underlying linear predictors ξ_i and $\xi_{j|i}$, by inverting the softmax function. (Note that this option would not be available for the z_i and $z_{j|i}$, even if they were known, because zero and one are not in the range of the softmax function.) Thus the targets for the ξ_i are the values:

$$\ln h_i^{(t)} - \ln C$$

where $C = \sum_k e^{\xi_k}$ is the normalization constant in the softmax function. Note, however, that constants that are common to all of the ξ_i can be omitted, because such constants disappear when ξ_i are converted to g_i (cf. equation 2.3). Thus the values $\ln h^{(t)}{}_i$ can be used as targets for the ξ_i. A similar argument shows that the values $\ln h^{(t)}{}_{l|k}$ can be used as targets for the ξ_{ij}, with observation weights $h^{(t)}{}_k$.

The utility of this approach is that once targets are available for the linear predictors ξ_i and ξ_{ij}, the problem of finding the parameters $\mathbf{v}_i$ and $\mathbf{v}_{ij}$ reduces to a coupled set of weighted least-squares problems. Thus we obtain an algorithm in which all of the parameters in the hierarchy, both in the expert networks and the gating networks, can be obtained by solving least-squares problems. This yields the following learning algorithm:

Algorithm 2

1. For each data pair $(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$, compute the posterior probabilities $h_i^{(t)}$ and $h_{j|i}^{(t)}$ using the current values of the parameters.

2. For each expert (i, j), solve a weighted least-squares problem with observations $\{(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})\}_1^N$ and observation weights $\{h_{ij}^{(t)}\}_1^N$.

3. For each top-level gating network, solve a least-squares problem with observations $\{(\mathbf{x}^{(t)}, \ln h_k^{(t)})\}_1^N$.

4. For each lower-level gating network, solve a weighted least-squares problem with observations $\{(\mathbf{x}^{(t)}, \ln h_{l|k}^{(t)})\}_1^N$ and observation weights $\{h_k^{(t)}\}_1^N$.

5. Iterate using the updated parameter values.

It is important to note that this algorithm does not yield the same parameter estimates as Algorithm 1; the gating network residuals $(h_i^{(t)} - g_i^{(t)})$ are being fit by least squares rather than maximum likelihood. The algorithm can be thought of as an approximation to Algorithm 1, an approximation based on the assumption that the differences between $h_i^{(t)}$ and $g_i^{(t)}$ are small. This assumption is equivalent to the assumption that the architecture can fit the underlying regression surface (a consistency condition) and the assumption that the noise is small. In practice we have found that the least-squares algorithm works reasonably well, even in the early stages of fitting when the residuals can be large. The ability to use least squares is certainly appealing from a computational point of view. One possible hybrid algorithm involves using the least-squares algorithm to converge quickly to the neighborhood of a solution and then using IRLS to refine the solution.

2.8 Simulation Results. We tested Algorithm 1 and Algorithm 2 on a nonlinear system identification problem. The data were obtained from a simulation of a four-joint robot arm moving in three-dimensional space (Fun and Jordan 1993). The network must learn the *forward dynamics* of the arm; a state-dependent mapping from joint torques to joint accelerations. The state of the arm is encoded by eight real-valued variables: four positions (rad) and four angular velocities (rad/sec). The torque was encoded as four real-valued variables (N · m). Thus there were 12 inputs to the learning system. Given these 12 input variables, the network must predict the four accelerations at the joints (rad/sec^2). This mapping is highly nonlinear due to the rotating coordinate systems and the interaction torques between the links of the arm.

We generated 15,000 data points for training and 5,000 points for testing. For each epoch (i.e., each pass through the training set), we computed the relative error on the test set. Relative error is computed as a ratio between the mean squared error and the mean squared error that would be obtained if the learner were to output the mean value of the accelerations for all data points.

We compared the performance of a binary hierarchy to that of a backpropagation network. The hierarchy was a four-level hierarchy with 16 expert networks and 15 gating networks. Each expert network had 4 output units and each gating network had 1 output unit. The backpropagation network had 60 hidden units, which yields approximately the same number of parameters in the network as in the hierarchy.

The HME architecture was trained by Algorithms 1 and 2, utilizing Cholesky decomposition to solve the weighted least-squares problems (Golub and van Loan 1989). Note that the HME algorithms have no free parameters. The free parameters for the backpropagation network (the learning rate and the momentum term) were chosen based on a coarse search of the parameter space. (Values of 0.00001 and 0.15 were chosen for these parameters.) There were difficulties with local minima (or plateaus) using the backpropagation algorithm: Five of 10 runs failed to converge to "reasonable" error values. (As we report in the next section, no such difficulties were encountered in the case of *on-line* backpropagation.) We report average convergence times and average relative errors only for those runs that converged to "reasonable" error values. All 10 runs for both of the HME algorithms converged to "reasonable" error values.

Figure 2 shows the performance of the hierarchy and the backpropagation network. The horizontal axis of the graph gives the training time in epochs. The vertical axis gives generalization performance as measured by the average relative error on the test set.

Table 1 reports the average relative errors for both architectures measured at the minima of the relative error curves. (Minima were defined by a sequence of three successive increases in the relative error.) We also report values of relative error for the best linear approximation, the CART algorithm, and the MARS algorithm. Both CART and MARS were run four times, once for each of the output variables. We combined the results from these four computations to compute the total relative error. Two versions of CART were run; one in which the splits were restricted to be parallel to the axes and one in which linear combinations of the input variables were allowed.

The MARS algorithm requires choices to be made for the values of two structural parameters: the maximum number of basis functions and the maximum number of interaction terms. Each basis function in MARS yields a linear surface defined over a rectangular region of the input

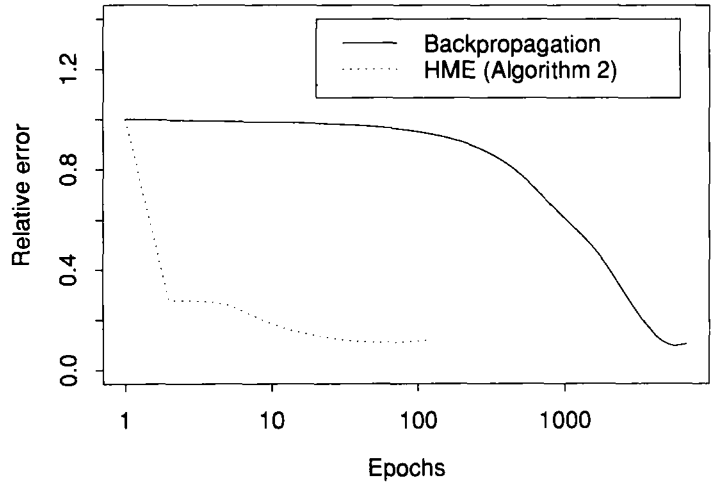

Figure 2: Relative error on the test set for a backpropagation network and a four-level HME architecture trained with batch algorithms. The standard errors at the minima of the curves are 0.013 for backpropagation and 0.002 for HME.

Table 1: Average Values of Relative Error and Number of Epochs Required for Convergence for the Batch Algorithms.

Architecture	Relative Error	# Epochs
Linear	0.31	1
Backpropagation	0.09	5,500
HME (Algorithm 1)	0.10	35
HME (Algorithm 2)	0.12	39
CART	0.17	NA
CART (linear)	0.13	NA
MARS	0.16	NA

space, corresponding roughly to the function implemented by a single expert in the HME architecture. Therefore we chose a maximum of 16 basis functions to correspond to the 16 experts in the four-level hierarchy. To choose the maximum number of interactions (*mi*), we compared the performance of MARS for $mi = 1, 2, 3, 6$, and 12, and chose the value that yielded the best performance ($mi = 3$).

For the iterative algorithms, we also report the number of epochs required for convergence. Because the learning curves for these algorithms generally have lengthy tails, we defined convergence as the first epoch at which the relative error drops within 5% of the minimum.

All of the architectures that we studied performed significantly better than the best linear approximation. As expected, the CART architecture with linear combinations performed better than CART with axis-parallel splits.[5] The HME architecture yielded a modest improvement over MARS and CART. Backpropagation produced the lowest relative error of the algorithms tested (ignoring the difficulties with convergence).

These differences in relative error should be treated with some caution. The need to set free parameters for some of the architectures (e.g., backpropagation) and the need to make structural choices (e.g., number of hidden units, number of basis functions, number of experts) make it difficult to match architectures. The HME architecture, for example, involves parameter dependencies that are not present in a backpropagation network. A gating network at a high level in the tree can "pinch off" a branch of the tree, rendering useless the parameters in that branch of the tree. Raw parameter count is therefore only a very rough guide to architecture capacity; more precise measures are needed (e.g., VC dimension) before definitive quantitative comparisons can be made.

The differences between backpropagation and HME in terms of convergence time are more definitive. Both HME algorithms reliably converge more than two orders of magnitude faster than backpropagation.

As shown in Figure 3, the HME architecture lends itself well to graphic investigation. This figure displays the time sequence of the distributions of posterior probabilities across the training set at each node of the tree. At Epoch 0, before any learning has taken place, most of the posterior probabilities at each node are approximately 0.5 across the training set. As the training proceeds, the histograms flatten out, eventually approaching bimodal distributions in which the posterior probabilities are either one or zero for most of the training patterns. This evolution is indicative of increasingly sharp splits being fit by the gating networks. Note that there is a tendency for the splits to be formed more rapidly at higher levels in the tree than at lower levels.

Figure 4 shows another graphic device that can be useful for understanding the way in which an HME architecture fits a data set. This figure, which we refer to as a "deviance tree," shows the deviance (mean squared error) that would be obtained at each level of the tree if the tree were clipped at that level. We construct a clipped tree at a given level by replacing each nonterminal at that level with a matrix that is a weighted average of the experts below that nonterminal. The weights are the total prior probabilities associated with each expert across the training set. The error for each output unit is then calculated by passing the test set through the clipped tree. As can be seen in the figure, the deviance is

[5] It should be noted that CART is at an advantage relative to the other algorithms in this comparison, because no structural parameters were fixed for CART. That is, CART is allowed to find the best tree of any size to fit the data.

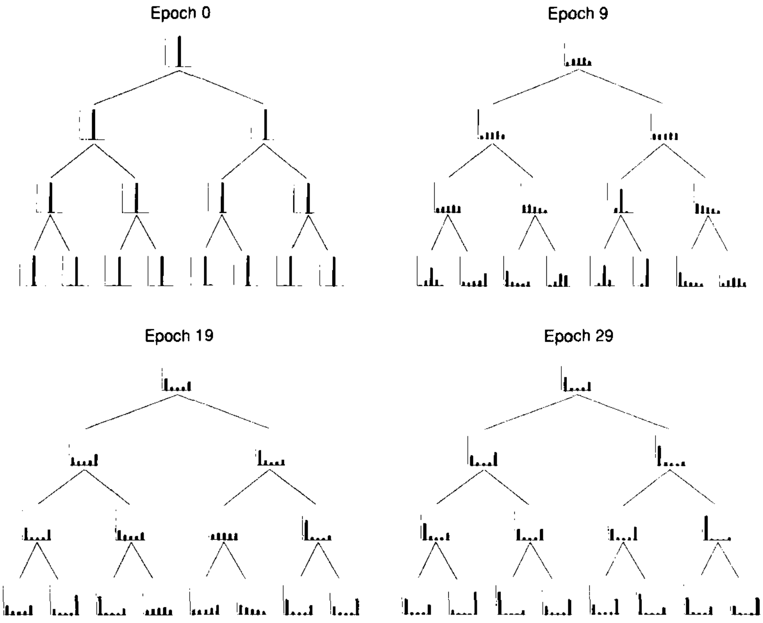

Figure 3: A sequence of histogram trees for the HME architecture. Each histogram displays the distribution of posterior probabilities across the training set at each node in the tree.

substantially smaller for deeper trees (note that the ordinate of the plots is on a log scale). The deviance in the right branch of the tree is larger than in the left branch of the tree. Information such as this can be useful for purposes of exploratory data analysis and for model selection.

2.9 An On-Line Algorithm. The batch least-squares algorithm that we have described (Algorithm 2) can be converted into an on-line algorithm by noting that linear least squares and weighted linear least squares problems can be solved by recursive procedures that update the parameter estimates with each successive data point (Ljung and Söderström 1986). Our application of these recursive algorithms is straightforward; however, care must be taken to handle the observation weights (the posterior probabilities) correctly. These weights change as a function of the changing parameter values. This implies that the recursive least squares algorithm must include a decay parameter that allows the system to "forget" older values of the posterior probabilities.

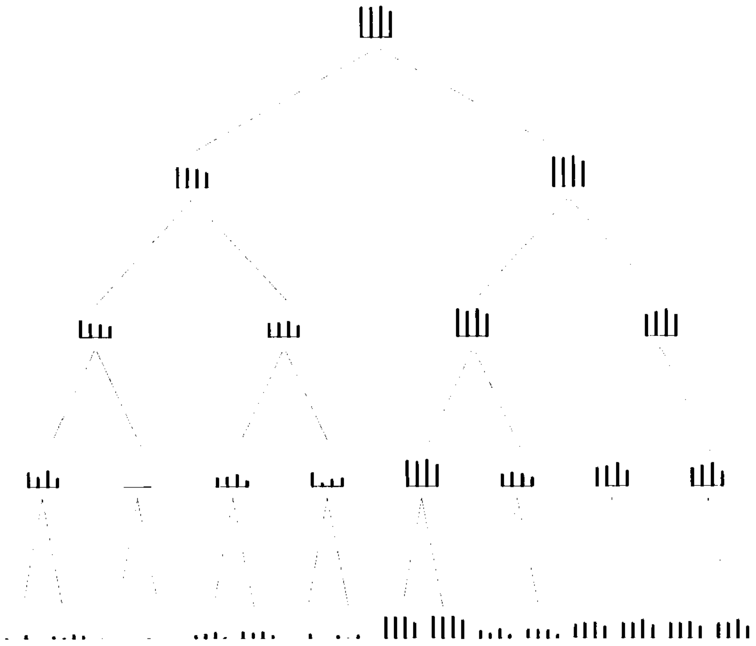

Figure 4: A deviance tree for the HME architecture. Each plot displays the mean squared error (MSE) for the four output units of the clipped tree. The plots are on a log scale covering approximately three orders of magnitude.

In this section we present the equations for the on-line algorithm. These equations involve an update not only of the parameters in each of the networks,[6] but also the storage and updating of an inverse covariance matrix for each network. Each matrix has dimensionality $m \times m$, where m is the dimensionality of the input vector. (Note that the size of these matrices depends on the square of the number of *input* variables, not the square of the number of *parameters*. Note also that the update equation for the inverse covariance matrix updates the inverse matrix directly; there is never a need to invert matrices.)

The on-line update rule for the parameters of the expert networks is given by the following recursive equation:

$$U_{ij}^{(t+1)} = U_{ij}^{(t)} + h_i^{(t)} h_{j|i}^{(t)} (\mathbf{y}^{(t)} - \boldsymbol{\mu}_{ij}^{(t)}) \mathbf{x}^{(t)T} R_{ij}^{(t)} \tag{2.32}$$

[6] Note that in this section we use the term "parameters" for the variables that are traditionally called "weights" in the neural network literature. We reserve the term "weights" for the observation weights.

where R_{ij} is the inverse covariance matrix for expert network (i, j). This matrix is updated via the equation:

$$R_{ij}^{(t)} = \lambda^{-1} R_{ij}^{(t-1)} - \lambda^{-1} \frac{R_{ij}^{(t-1)} \mathbf{x}^{(t)} \mathbf{x}^{(t)T} R_{ij}^{(t-1)}}{\lambda [h_{ij}^{(t)}]^{-1} + \mathbf{x}^{(t)T} R_{ij}^{(t-1)} \mathbf{x}^{(t)}} \tag{2.33}$$

where λ is the decay parameter.

It is interesting to note the similarity between the parameter update rule in equation 2.32 and the gradient rule presented earlier (cf. equation 2.14). These updates are essentially the same, except that the scalar ρ is replaced by the matrix $R_{ij}^{(t)}$. It can be shown, however, that $R_{ij}^{(t)}$ is an estimate of the inverse Hessian of the least-squares cost function (Ljung and Söderström 1986), thus equation 2.32 is in fact a stochastic approximation to a Newton–Raphson method rather than a gradient method.[7]

Similar equations apply for the updates of the gating networks. The update rule for the parameters of the top-level gating network is given by the following equation (for the ith output of the gating network):

$$\mathbf{v}_i^{(t+1)} = \mathbf{v}_i^{(t)} + S_i^{(t)} (\ln h_i^{(t)} - \xi_i^{(t)}) \mathbf{x}^{(t)} \tag{2.34}$$

where the inverse covariance matrix S_i is updated by

$$S_i^{(t)} = \lambda^{-1} S_i^{(t-1)} - \lambda^{-1} \frac{S_i^{(t-1)} \mathbf{x}^{(t)} \mathbf{x}^{(t)T} S_i^{(t-1)}}{\lambda + \mathbf{x}^{(t)T} S_i^{(t-1)} \mathbf{x}^{(t)}} \tag{2.35}$$

Finally, the update rule for the parameters of the lower-level gating network is as follows:

$$\mathbf{v}_{ij}^{(t+1)} = \mathbf{v}_{ij}^{(t)} + S_{ij}^{(t)} h_i^{(t)} (\ln h_{j|i}^{(t)} - \xi_{ij}^{(t)}) \mathbf{x}^{(t)} \tag{2.36}$$

where the inverse covariance matrix S_i is updated by

$$S_{ij}^{(t)} = \lambda^{-1} S_{ij}^{(t-1)} - \lambda^{-1} \frac{S_{ij}^{(t-1)} \mathbf{x}^{(t)} \mathbf{x}^{(t)T} S_{ij}^{(t-1)}}{\lambda [h_i^{(t)}]^{-1} + \mathbf{x}^{(t)T} S_{ij}^{(t-1)} \mathbf{x}^{(t)}} \tag{2.37}$$

2.10 Simulation Results. The on-line algorithm was tested on the robot dynamics problem described in the previous section. Preliminary simulations convinced us of the necessity of the decay parameter (λ). We also found that this parameter should be slowly increased as training proceeds—on the early trials the posterior probabilities are changing rapidly so that the covariances should be decayed rapidly, whereas on later trials the posterior probabilities have stabilized and the covariances should be decayed less rapidly. We used a simple fixed schedule: λ was

[7] This is true for fixed values of the posterior probabilities. These posterior probabilities are also changing over time, however, as required by the EM algorithm. The overall convergence rate of the algorithm is determined by the convergence rate of EM, not the convergence rate of Newton–Raphson.

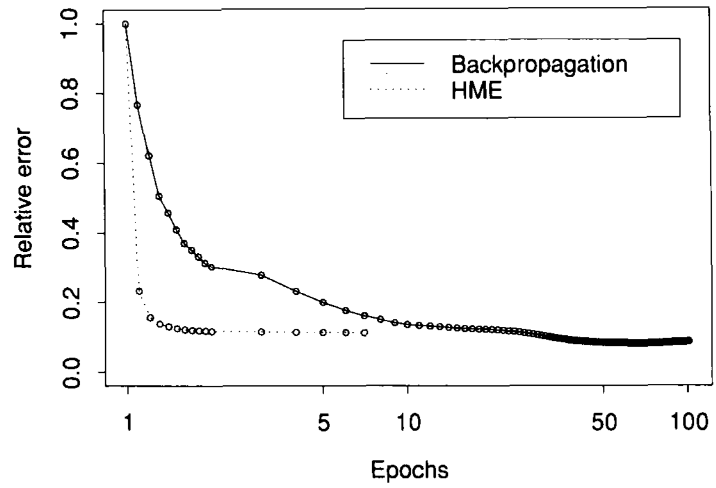

Figure 5: Relative error on the test set for a backpropagation network and a four-level hierarchy trained with on-line algorithms. The standard errors at the minima of the curves are 0.008 for backpropagation and 0.009 for HME.

initialized to 0.99 and increased a fixed fraction (0.6) of the remaining distance to 1.0 every 1000 time steps.

The performance of the on-line algorithm was compared to an on-line backpropagation network. Parameter settings for the backpropagation network were obtained by a coarse search through the parameter space, yielding a value of 0.15 for the learning rate and 0.20 for the momentum. The results for both architectures are shown in Figure 5. As can be seen, the on-line algorithm for backpropagation is significantly faster than the corresponding batch algorithm (cf. Fig. 2). This is also true of the on-line HME algorithm, which has nearly converged within the first epoch.

The minimum values of relative error and the convergence times for both architectures are provided in Table 2. We also provide the corresponding values for a simulation of the on-line gradient algorithm for the HME architecture (equation 2.17).

We also performed a set of simulations which tested a variety of different HME architectures. We compared a one-level hierarchy with 32 experts to hierarchies with five levels (32 experts), and six levels (64 experts). We also simulated two three-level hierarchies, one with branching factors of 4, 4, and 2 (proceeding from the top of the tree to the bottom), and one with branching factors of 2, 4, and 4. (Each three-level hierarchy contained 32 experts.) The results are shown in Figure 6. As can be

Table 2: Average Values of Relative Error and Number of Epochs Required for Convergence for the On-Line Algorithms.

Architecture	Relative Error	Number of Epochs
Linear	0.32	1
Backpropagation (on-line)	0.08	63
HME (on-line)	0.12	2
HME (gradient)	0.15	104

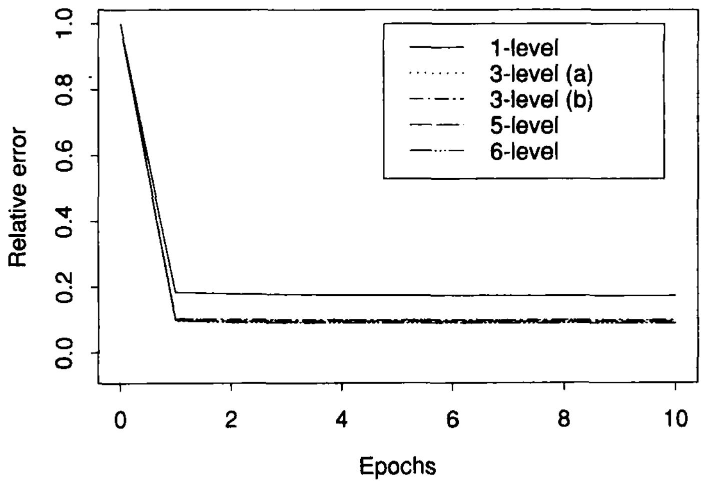

Figure 6: Relative error on the test set for HME hierarchies with different structures. "3-level (a)" refers to a 3-level hierarchy with branching factors of 4, 4, and 2, and "3-level (b)" refers to a 3-level hierarchy with branching factors of 2, 4, and 4. The standard errors for all curves at their respective minima were approximately 0.009.

seen, there was a significant difference between the one-level hierarchy and the other architectures. There were smaller differences among the multilevel hierarchies. No significant difference was observed between the two different 3-level architectures.

3 Model Selection

Utilizing the HME approach requires that choices be made regarding the structural parameters of the model, in particular the number of levels and the branching factor of the tree. As with other flexible estimation techniques, it is desirable to allow these structural parameters to be chosen based at least partly on the data. This model selection problem can be addressed in a variety of ways. In this paper we have utilized a test set approach to model selection, stopping the training when the error on the test set reaches a minimum. As is the case with other neural network algorithms, this procedure can be justified as a complexity control measure. As we have noted, when the parameters in the gating networks of an HME architecture are small, the entire system reduces to a single "averaged" GLIM at the root of the tree. As the training proceeds, the parameters in the gating networks begin to grow in magnitude and splits are formed. When a split is formed the parameters in the branches of the tree on either side of the split are decoupled and the effective number of degrees of freedom in the system increases. This increase in complexity takes place gradually as the values of the parameters increase and the splits sharpen. By stopping the training of the system based on the performance on a test set, we obtain control over the effective number of degrees of freedom in the architecture.

Other approaches to model selection can also be considered. One natural approach is to use ridge regression in each of the expert networks and the gating networks. This approach extends naturally to the on-line setting in the form of a "weight decay." It is also worth considering Bayesian techniques of the kind considered in the decision tree literature by Buntine (1991), as well as the MDL methods of Quinlan and Rivest (1989).

4 Related work

There are a variety of ties that can be made between the HME architecture and related work in statistics, machine learning, and neural networks. In this section we briefly mention some of these ties and make some comparative remarks.

Our architecture is not the only nonlinear approximator to make substantial use of GLIMs and the IRLS algorithm. IRLS also figures prominently in a branch of nonparametric statistics known as generalized additive models (GAMs; Hastie and Tibshirani 1990). It is interesting to note the complementary roles of IRLS in these two architectures. In the GAM model, the IRLS algorithm appears in the outer loop, providing an adjusted dependent variable that is fit by a backfitting procedure in the inner loop. In the HME approach, on the other hand, the outer loop is the E step of EM and IRLS is in the inner loop. This complementarity suggests that it might be of interest to consider hybrid models in which a HME is nested inside a GAM or vice versa.

We have already mentioned the close ties between the HME approach and other tree-structured estimators such as CART and MARS. Our approach differs from MARS and related architectures—such as the basis-function trees of Sanger (1991)—by allowing splits that are oblique with respect to the axes. We also differ from these architectures by using a statistical model—the multinomial logit model—for the splits. We believe that both of these features can play a role in increasing predictive ability—the use of oblique splits should tend to decrease bias, and the use of smooth multinomial logit splits should generally decrease variance. Oblique splits also render the HME architecture insensitive to the particular choice of coordinates used to encode the data. Finally, it is worth emphasizing the difference in philosophy behind these architectures. Whereas CART and MARS are entirely nonparametric, the HME approach has a strong flavor of parametric statistics, via its use of generalized linear models, mixture models, and maximum likelihood.

Similar comments can be made with respect to the decision tree methodology in the machine learning literature. Algorithms such as ID3 build trees that have axis-parallel splits and use heuristic splitting algorithms (Quinlan 1986). More recent research has studied decision trees with oblique splits (Murthy *et al.* 1993; Utgoff and Brodley 1990). None of these papers, however, has treated the problem of splitting data as a statistical problem, nor have they provided a global goodness-of-fit measure for their trees.

There are a variety of neural network architectures that are related to the HME architecture. The multiresolution aspect of HME is reminiscent of Moody's (1989) multiresolution CMAC hierarchy, differing in that Moody's levels of resolution are handled explicitly by separate networks. The "neural tree" algorithm (Strömberg *et al.* 1991) is a decision tree with multilayer perceptions (MLPs) at the nonterminals. This architecture can form oblique (or curvilinear) splits, however, the MLPs are trained by a heuristic that has no clear relationship to overall classification performance. Finally, Hinton and Nowlan (see Nowlan 1991) have independently proposed extending the Jacobs *et al.* (1991) modular architecture to a tree-structured system. They did not develop a likelihood approach to the problem, however, proposing instead a heuristic splitting scheme.

5 Conclusions

We have presented a tree-structured architecture for supervised learning. We have developed the learning algorithm for this architecture within the framework of maximum likelihood estimation, utilizing ideas from mixture model estimation and generalized linear model theory. The maximum likelihood framework allows standard tools from statistical theory to be brought to bear in developing inference procedures and measures of uncertainty for the architecture (Cox and Hinkley 1974). It also opens the door to the Bayesian approaches that have been found to be useful in the context of unsupervised mixture model estimation (Cheeseman *et al.* 1988).

Although we have not emphasized theoretical issues in this paper, there are a number of points that are worth mentioning. First, the set of exponentially smoothed piecewise linear functions that we have utilized is clearly dense in the set of piecewise linear functions on compact sets in $\Re^m$, thus it is straightforward to show that the hierarchical architecture is dense in the set of continuous functions on compact sets in $\Re^m$. That is, the architecture is "universal" in the sense of Hornik *et al.* (1989). From this result it would seem straightforward to develop consistency results for the architecture (cf. Geman *et al.* 1992; Stone 1977). We are currently developing this line of argument and are studying the asymptotic distributional properties of fixed hierarchies. Second, convergence results are available for the architecture. We have shown that the convergence rate of the algorithm is linear in the condition number of a matrix that is the product of an inverse covariance matrix and the Hessian of the log likelihood for the architecture (Jordan and Xu 1993).

Finally, it is worth noting a number of possible extensions of the work reported here. Our earlier work on hierarchical mixtures of experts utilized the multilayer perceptron as the primitive function for the expert networks and gating networks (Jordan and Jacobs 1992). That option is still available, although we lose the EM proof of convergence (cf. Jordan and Xu 1993) and we lose the ability to fit the subnetworks efficiently with IRLS. One interesting example of such an application is the case where the experts are autoassociators (Bourlard and Kamp 1988), in which case the architecture fits hierarchically nested local principal component decompositions. Another area in unsupervised learning worth exploring is the nonassociative version of the hierarchical architecture. Such a model would be a recursive version of classical mixture-likelihood clustering and may have interesting ties to hierarchical clustering models. Finally, it is also of interest to note that the recursive least squares algorithm that we utilized in obtaining an on-line variant of Algorithm 2 is not the only possible on-line approach. Any of the fast filter algorithms (Haykin 1991) could also be utilized, giving rise to a family of on-line algorithms. Also, it is worth studying the application of the recursive algorithms to PRESS-like cross-validation calculations to efficiently compute the changes in likelihood that arise from adding or deleting parameters or data points.

Appendix A: Iteratively Reweighted Least Squares

The iteratively reweighted least squares (IRLS) algorithm is the inner loop of the algorithm that we have proposed for the HME architecture. In this section, we describe the IRLS algorithm, deriving it as a special case of the Fisher scoring method for generalized linear models. Our presentation derives from McCullagh and Nelder (1983).

IRLS is an iterative algorithm for computing the maximum likelihood estimates of the parameters of a generalized linear model. It is a special case of a general algorithm for maximum likelihood estimation known as the Fisher scoring method (Finney 1973). Let $l(\boldsymbol{\beta}; \mathcal{X})$ be a log likelihood function—a function of the parameter vector $\boldsymbol{\beta}$—and let $(\partial l / \partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T)$ denote the Hessian of the log likelihood. The Fisher scoring method updates the parameter estimates $\boldsymbol{\beta}$ as follows:

$$\boldsymbol{\beta}_{r+1} = \boldsymbol{\beta}_r - \left\{ E\left[\frac{\partial l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} \right] \right\}^{-1} \frac{\partial l}{\partial \boldsymbol{\beta}} \tag{5.1}$$

where $\boldsymbol{\beta}_r$ denotes the parameter estimate at the rth iteration and $\partial l / \partial \boldsymbol{\beta}$ is the gradient vector. Note that the Fisher scoring method is essentially the same as the Newton–Raphson algorithm, except that the expected value of the Hessian replaces the Hessian. There are statistical reasons for preferring the expected value of the Hessian—and the expected value of the Hessian is often easier to compute—but Newton–Raphson can also be used in many cases.

The likelihood in generalized linear model theory is a product of densities from the exponential family of distributions. This family is an important class in statistics and includes many useful densities, such as the normal, the Poisson, the binomial, and the gamma. The general form of a density in the exponential family is the following:

$$P(y, \eta, \phi) = \exp\{(\eta y - b(\eta))/\phi + c(y, \phi)\} \tag{5.2}$$

where η is known as the "natural parameter" and ϕ is the dispersion parameter.[8]

Example (Bernoulli Density). The Bernoulli density with mean π has the following form:

$$\begin{aligned} P(y, \pi) &= \pi^y (1 - \pi)^{1-y} \\ &= \exp\{\ln(\frac{\pi}{1 - \pi}) y + \ln(1 - \pi)\} \\ &= \exp\{\eta y - \ln(1 + e^\eta)\} \end{aligned} \tag{5.3}$$

where $\eta = \ln(\pi / 1 - \pi)$ is the natural parameter of the Bernoulli density. This parameter has the interpretation as the log odds of "success" in a random Bernoulli experiment.

In a generalized linear model, the parameter η is modeled as a linear function of the input $\mathbf{x}$:

$$\eta = \boldsymbol{\beta}^T \mathbf{x}$$

[8] We restrict ourselves to scalar-valued random variables to simplify the presentation, and describe the (straightforward) extension to vector-valued random variables at the end of the section.

where β is a parameter vector. Substituting this expression into equation 5.2 and taking the product of N such densities yields the following log likelihood for a data set $\mathcal{X} = \{(\mathbf{x}^{(t)}, y^{(t)})\}_1^N$:

$$l(\beta, \mathcal{X}) = \sum_t \{(\beta^T \mathbf{x}^{(t)} y^{(t)} - b(\beta^T \mathbf{x}^{(t)}))/\phi + c(y^{(t)}, \phi)\}$$

The observations $y^{(t)}$ are assumed to be sampled independently from densities $P(y, \eta^{(t)}, \phi)$, where $\eta^{(t)} = \beta^T \mathbf{x}^{(t)}$.

We now compute the gradient of the log likelihood:

$$\frac{\partial l}{\partial \beta} = \sum_t (y^{(t)} - b'(\beta^T \mathbf{x}^{(t)}))\mathbf{x}^{(t)}/\phi \tag{5.4}$$

and the Hessian of the log likelihood:

$$\frac{\partial l}{\partial \beta \partial \beta^T} = -\sum_t b''(\beta^T \mathbf{x}^{(t)})\mathbf{x}^{(t)}\mathbf{x}^{(t)T}/\phi \tag{5.5}$$

These quantities could be substituted directly into equation 5.1, however, there is additional mathematical structure that can be exploited. First note the following identity, which is true of any log likelihood:

$$E\left[\frac{\partial l}{\partial \beta}\right] = 0$$

(This fact can be proved by differentiating both sides of the identity $\int P(y, \beta, \phi) dy = 1$ with respect to β.) Because this identity is true for any set of observed data, including all subsets of $\mathcal{X}$, we have the following:

$$E[y^{(t)}] = b'(\beta^T \mathbf{x}^{(t)})$$

for all t. This equation implies that the mean of $y^{(t)}$, which we denote as $\mu^{(t)}$, is a function of $\eta^{(t)}$. We therefore include in the generalized linear model the *link function*, which models μ as a function of η:

$$\mu^{(t)} = f(\eta^{(t)})$$

Example (Bernoulli Density). Equation 5.3 shows that $b(\eta) = \ln(1 + e^\eta)$ for the Bernoulli density. Thus

$$\mu = b'(\eta) = \frac{e^\eta}{1 + e^\eta}$$

which is the logistic function. Inverting the logistic function yields $\eta = \ln(\mu/1 - \mu)$; thus, μ equals π, as it must.

The link function $f(\eta) = b'(\eta)$ is known in generalized linear model theory as the *canonical link*. By parameterizing the exponential family density in terms of η (cf. equation 5.2), we have forced the choice of the canonical link. It is also possible to use other links, in which case η

no longer has the interpretation as the natural parameter of the density. There are statistical reasons, however, to prefer the canonical link (McCullagh and Nelder 1983). Moreover, by choosing the canonical link, the Hessian of the likelihood turns out to be constant (cf. equation 5.5), and the Fisher scoring method therefore reduces to Newton–Raphson.[9]

To continue the development, we need an additional fact about log likelihoods. By differentiating the identity $\int P(y,\beta)dy = 1$ twice with respect to β, the following identity can be established:

$$E\left[\frac{\partial l}{\partial\beta\partial\beta^T}\right] = -E\left[\frac{\partial l}{\partial\beta}\right]\left[\frac{\partial l}{\partial\beta}\right]^T$$

This identity can be used to obtain a relationship between the variance of η and the function $b(\eta)$ in the exponential family density. Beginning with equation 5.5, we have

$$\begin{aligned}
-E&\left[\sum_t b''(\beta^T\mathbf{x}^{(t)})\mathbf{x}^{(t)}\mathbf{x}^{(t)T}/\phi\right]\\
&= E\left[\frac{\partial l}{\partial\beta\partial\beta^T}\right]\\
&= -E\left[\frac{\partial l}{\partial\beta}\right]\left[\frac{\partial l}{\partial\beta}\right]^T\\
&= -\frac{1}{\phi^2}E\left[\sum_t(y^{(t)} - b'(\beta^T\mathbf{x}^{(t)}))\mathbf{x}^{(t)}\sum_s(y^{(s)} - b'(\beta^T\mathbf{x}^{(s)}))\mathbf{x}^{(s)T}\right]\\
&= -\frac{1}{\phi^2}E\left[\sum_t(y^{(t)} - b'(\beta^T\mathbf{x}^{(t)}))^2\mathbf{x}^{(t)}\mathbf{x}^{(t)T}\right]\\
&= -\frac{1}{\phi^2}\sum_t \mathrm{Var}[y^{(t)}]\mathbf{x}^{(t)}\mathbf{x}^{(t)T}
\end{aligned}$$

where we have used the independence assumption in the fourth step. Comparing equation 5.5 with the last equation, we obtain the following relationship:

$$\mathrm{Var}[y^{(t)}] = \phi b''(\beta^T\mathbf{x}^{(t)})$$

Moreover, because $f(\eta) = b'(\eta)$, we have

$$\mathrm{Var}[y^{(t)}] = \phi f'(\beta^T\mathbf{x}^{(t)}) \tag{5.6}$$

We now assemble the various pieces. First note that equation 5.6 can be utilized to express the Hessian (equation 5.5) in the following form:

$$\frac{\partial l}{\partial\beta\partial\beta^T} = -\sum_t \mathbf{x}^{(t)}\mathbf{x}^{(t)T}w^{(t)}$$

[9] Whether or not the canonical link is used, the results presented in the remainder of this section are correct for the Fisher scoring method. If noncanonical links are used, then Newton–Raphson will include additional terms (terms that vanish under the expectation operator).

Finally, note that equation 6.2 implies that $b(\eta)$ must be defined as follows (cf. equation 5.2):

$$b(\eta) = n \ln \left(\sum_{i=1}^{n} e^{\eta_i} \right)$$

which implies

$$\mu_i = \frac{\partial b(\eta)}{\partial \eta_i} = \frac{n e^{\eta_i}}{\sum_{j=1}^{n} e^{\eta_j}} = n p_i \tag{6.5}$$

The fitting of a multinomial logit model proceeds by IRLS as described in Appendix A, using equations 6.4 and 6.5 for the link function and the mean, respectively.

Acknowledgments

We want to thank Geoffrey Hinton, Tony Robinson, Mitsuo Kawato, Carlotta Domeniconi, and Daniel Wolpert for helpful comments on the manuscript. 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, by Grant IRI-9013991 from the National Science Foundation, 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, and NSF Grant ECS-9216531 to support an Initiative in Intelligent Control at MIT. Michael I. Jordan is an NSF Presidential Young Investigator.

References

Bourlard, H., and Kamp, Y. 1988. Auto-association by multilayer perceptrons and singular value decomposition. *Biol. Cybern.* **59**, 291–294.

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

Bridle, J. 1989. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In *Neurocomputing: Algorithms, Architectures, and Applications,* F. Fogelman-Soulie and J. Hérault, eds. Springer-Verlag, New York.

Buntine, W. 1991. *Learning classification trees.* NASA Ames Tech. Rep. FIA-90-12-19-01, Moffett Field, CA.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., and Freeman, D. 1988. Autoclass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning,* Ann Arbor, MI.

Cox, D. R. 1970. *The Analysis of Binary Data.* Chapman-Hall, London.

Cox, D. R., and Hinkley, D. V. 1974. *Theoretical Statistics*. Chapman-Hall, London.

Dempster, A. P., Laird, N. M., and Rubin, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B* **39**, 1–38.

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

Finney, D. J. 1973. *Statistical Methods in Biological Assay*. Hafner, New York.

Friedman, J. H. 1991. Multivariate adaptive regression splines. *Ann. Statist.* **19**, 1–141.

Fun, W., and Jordan, M. I. 1993. *The Moving Basin: Effective Action Search in Forward Models*. MIT Computational Cognitive Science Tech. Report 9205, Cambridge, MA.

Geman, S., Bienenstock, E., and Doursat, R. 1992. Neural networks and the bias/variance dilemma. *Neural Comp.* **4**, 1–52.

Golub, G. H., and Van Loan, G. F. 1989. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD.

Hastie, T. J., and Tibshirani, R. J. 1990. *Generalized Additive Models*. Chapman and Hall, London.

Haykin, S. 1991. *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, NJ.

Hinton, G. E., and Sejnowski, T. J. 1986. Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, eds., Vol. 1, pp. 282–317. MIT Press, Cambridge, MA.

Hornik, K., Stinchcombe, M., and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural Comp.* **3**, 79–87.

Jordan, M. I., and Jacobs, R. A. 1992. Hierarchies of adaptive experts. In *Advances in Neural Information Processing Systems 4*, J. Moody, S. Hanson, and R. Lippmann, eds., pp. 985-993. Morgan Kaufmann, San Mateo, CA.

Jordan, M. I., and Xu, L. 1993. *Convergence Properties of the EM Approach to Learning in Mixture-of-Experts Architectures*. Computational Cognitive Science Tech. Rep. 9301, MIT, Cambridge, MA.

Little, R. J. A., and Rubin, D. B. 1987. *Statistical Analysis with Missing Data*. John Wiley, New York.

Ljung, L., and Söderström, T. 1986. *Theory and Practice of Recursive Identification*. MIT Press, Cambridge.

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

Moody, J. 1989. Fast learning in multi-resolution hierarchies. In *Advances in Neural Information Processing Systems*, D. S. Touretzky, ed. Morgan Kaufmann, San Mateo, CA.

Murthy, S. K., Kasif, S., and Salzberg, S. 1993. *OC1: A Randomized Algorithm for Building Oblique Decision Trees*. Tech. Rep., Department of Computer Science, The Johns Hopkins University.

Nowlan, S. J. 1990. Maximum likelihood competitive learning. In *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, ed. Morgan Kaufmann, San Mateo, CA.

Nowlan, S. J. 1991. *Soft Competitive Adaptation: Neural Network Learning Algorithms Based on Fitting Statistical Mixtures.* Tech. Rep. CMU-CS-91-126, CMU, Pittsburgh, PA.

Quandt, R. E., and Ramsey, J. B. 1972. A new approach to estimating switching regressions. *J. Am. Statist. Soc.* **67**, 306–310.

Quinlan, J. R. 1986. Induction of decision trees. *Machine Learn.* **1**, 81–106.

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

Redner, R. A., and Walker, H. F. 1984. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **26**, 195–239.

Sanger, T. D. 1991. A tree-structured adaptive network for function approximation in high dimensional spaces. *IEEE Transact. Neural Networks* **2**, 285–293.

Scott, D. W. 1992. *Multivariate Density Estimation.* John Wiley, New York.

Specht, D. F. 1991. A general regression neural network. *IEEE Transact. Neural Networks* **2**, 568–576.

Stone, C. J. 1977. Consistent nonparametric regression. *Ann. Statist.* **5**, 595–645.

Strömberg, J. E., Zrida, J., and Isaksson, A. 1991. Neural trees—using neural nets in a tree classifier structure. *IEEE International Conference on Acoustics, Speech and Signal Processing*, 137–140.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. 1985. *Statistical Analysis of Finite Mixture Distributions.* John Wiley, New York.

Utgoff, P. E., and 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.

Wahba, G., Gu, C., Wang, Y., and Chappell, R. 1993. *Soft Classification, a.k.a. Risk Estimation, via Penalized Log Likelihood and Smoothing Spline Analysis of Variance.* Tech. Rep. 899, Department of Statistics, University of Wisconsin, Madison.

Wu, C. F. J. 1983. On the convergence properties of the EM algorithm. *Ann. Statist.* **11**, 95–103.

Received February 22, 1993; accepted July 9, 1993.

This article has been cited by:

1. Nikolay Nikolaev, Peter Tino, Evgueni Smirnov. 2013. Time-dependent series variance learning with recurrent mixture density networks. *Neurocomputing* **122**, 501-512. [CrossRef]
2. Jason Sherwin, Paul Sajda. 2013. Musical experts recruit action-related neural structures in harmonic anomaly detection: Evidence for embodied cognition in expertise. *Brain and Cognition* **83**:2, 190-202. [CrossRef]
3. Bruno Damas, José Santos-Victor. 2013. Online Learning of Single- and Multivalued Functions with an Infinite Mixture of Linear Experts. *Neural Computation* **25**:11, 3044-3091. [Abstract] [Full Text] [PDF] [PDF Plus]
4. Martin Fergie, Aphrodite Galata. 2013. Mixtures of Gaussian Process Models for Human Pose Estimation. *Image and Vision Computing* . [CrossRef]
5. Weixin Yao, Yan Wei, Chun Yu. 2013. Robust mixture regression using the -distribution. *Computational Statistics & Data Analysis* . [CrossRef]
6. Choo Jun Tan, Chee Peng Lim, Yu–N Cheah. 2013. A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models. *Neurocomputing* . [CrossRef]
7. Chyon Hae Kim, Hiroshi Tsujino, Hiroyuki Nakahara. 2013. Reinforcement learning system based on heuristics free state focusing. *Advanced Robotics* **27**:10, 749-758. [CrossRef]
8. Ashfaqur Rahman, Brijesh Verma. 2013. Ensemble classifier generation using non-uniform layered clustering and Genetic Algorithm. *Knowledge-Based Systems* **43**, 30-42. [CrossRef]
9. Mehrdad Javadi, Seyed Ali Asghar Abbaszadeh Arani, Atena Sajedin, Reza Ebrahimpour. 2013. Classification of ECG arrhythmia by a modular neural network based on Mixture of Experts and Negatively Correlated Learning. *Biomedical Signal Processing and Control* **8**:3, 289-296. [CrossRef]
10. Erwin Jeremiah, Lucy Marshall, Scott A Sisson, Ashish Sharma. 2013. Specifying a hierarchical mixture of experts for hydrologic modeling: Gating function variable selection. *Water Resources Research* **49**:5, 2926-2939. [CrossRef]
11. David J. Nott, Lucy Marshall, Mark Fielding, Shie-Yui Liong. 2013. Mixtures of experts for understanding model discrepancy in dynamic computer models. *Computational Statistics & Data Analysis* . [CrossRef]
12. Jeff A. Tracey, Jun Zhu, Erin Boydston, Lisa Lyren, Robert N. Fisher, Kevin R. Crooks. 2013. Mapping behavioral landscapes for animal movement: a finite mixture modeling approach. *Ecological Applications* **23**:3, 654-669. [CrossRef]
13. Sheng Jin, Dian-hai Wang, Cheng Xu, Dong-fang Ma. 2013. Short-term traffic safety forecasting using Gaussian mixture model and Kalman filter. *Journal of Zhejiang University SCIENCE A* **14**:4, 231-243. [CrossRef]

14. Monica Billio, Roberto Casarin, Francesco Ravazzolo, Herman K. van Dijk. 2013. Time-varying combinations of predictive densities using nonlinear filtering. *Journal of Econometrics* . [CrossRef]

15. P. Hall, Y. Xia, J.-H. Xue. 2013. Simple tiered classifiers. *Biometrika* . [CrossRef]

16. Tarek Helmy, S. M. Rahman, Muhammad Imtiaz Hossain, Abdulaziz Abdelraheem. 2013. Non-linear Heterogeneous Ensemble Model for Permeability Prediction of Oil Reservoirs. *Arabian Journal for Science and Engineering* . [CrossRef]

17. Martin Schels, Stefan Scherer, Michael Glodek, Hans A. Kestler, Günther Palm, Friedhelm Schwenker. 2013. On the discovery of events in EEG data utilizing information fusion. *Computational Statistics* **28**:1, 5-18. [CrossRef]

18. P. Arun Raj Kumar, S. Selvakumar. 2013. Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Computer Communications* **36**:3, 303-319. [CrossRef]

19. Yanan Fan, David J. Nott, Scott A. Sisson. 2013. Approximate Bayesian computation via regression density estimation. *Stat* **2**:1, 34-48. [CrossRef]

20. Montserrat Fuentes, Kristen FoleyEnsemble Models . [CrossRef]

21. Julien Cornebise, Eric Moulines, Jimmy Olsson. 2013. Adaptive sequential Monte Carlo by means of mixture of experts. *Statistics and Computing* . [CrossRef]

22. Sungmin Myoung. 2013. Modified Mixture of Experts for the Diagnosis of Perfusion Magnetic Resonance Imaging Measures in Locally Rectal Cancer Patients. *Healthcare Informatics Research* **19**:2, 130. [CrossRef]

23. Mahmoud Tarokh. 2013. Solving inverse problems by decomposition, classification and simple modeling. *Information Sciences* **218**, 51-60. [CrossRef]

24. Antonio CiampiPrediction Trees 1054-1060. [CrossRef]

25. Mattias Villani, Robert Kohn, David J. Nott. 2012. Generalized smooth finite mixtures. *Journal of Econometrics* **171**:2, 121-133. [CrossRef]

26. Asim Ansari, Ricardo Montoya, Oded Netzer. 2012. Dynamic learning in behavioral games: A hidden Markov mixture of experts approach. *Quantitative Marketing and Economics* **10**:4, 475-503. [CrossRef]

27. K. Pichara, P. Protopapas, D.-W. Kim, J.-B. Marquette, P. Tisserand. 2012. An improved quasar detection method in EROS-2 and MACHO LMC data sets. *Monthly Notices of the Royal Astronomical Society* **427**:2, 1284-1297. [CrossRef]

28. Eduardo F. Mendes, Wenxin Jiang. 2012. On Convergence Rates of Mixtures of Polynomial Experts. *Neural Computation* **24**:11, 3025-3051. [Abstract] [Full Text] [PDF] [PDF Plus]

29. Syed Masiur Rahman, A.N. Khondaker, Radwan Abdel-Aal. 2012. Self organizing ozone model for Empty Quarter of Saudi Arabia: Group method data handling based modeling approach. *Atmospheric Environment* **59**, 398-407. [CrossRef]

30. Salvatore Ingrassia, Simona C. Minotti, Giorgio Vittadini. 2012. Local Statistical Modeling via a Cluster-Weighted Approach with Elliptical Distributions. *Journal of Classification* **29**:3, 363-401. [CrossRef]

31. Marcus Musselman, Dragan Djurdjanovic. 2012. Time–frequency distributions in the classification of epilepsy from EEG signals. *Expert Systems with Applications* **39**:13, 11413-11422. [CrossRef]

32. J.J. Triano, M. Descarreaux, C. Dugas. 2012. Biomechanics – Review of approaches for performance training in spinal manipulation. *Journal of Electromyography and Kinesiology* **22**:5, 732-739. [CrossRef]

33. Marek Kurzynski, Michal Wozniak. 2012. Combining classifiers under probabilistic models: experimental comparative analysis of methods. *Expert Systems* **29**:4, 374-393. [CrossRef]

34. Igor T. Podolak, Adam Roman. 2012. THEORETICAL FOUNDATIONS AND EXPERIMENTAL RESULTS FOR A HIERARCHICAL CLASSIFIER WITH OVERLAPPING CLUSTERS. *Computational Intelligence* no-no. [CrossRef]

35. Wenting Lu, Lei Li, Jingxuan Li, Tao Li, Honggang Zhang, Jun Guo. 2012. A multimedia information fusion framework for web image categorization. *Multimedia Tools and Applications* . [CrossRef]

36. Reza Ebrahimpour, Naser Sadeghnejad, Atena Sajedin, Nima Mohammadi. 2012. Electrocardiogram beat classification via coupled boosting by filtering and preloaded mixture of experts. *Neural Computing and Applications* . [CrossRef]

37. David J. Nott, Siew Li Tan, Mattias Villani, Robert Kohn. 2012. Regression Density Estimation With Variational Methods and Stochastic Approximation. *Journal of Computational and Graphical Statistics* **21**:3, 797-820. [CrossRef]

38. Sung-Min Myoung, Dong-Geon Kim, Jin-Nam Jo. 2012. A Study of HME Model in Time-Course Microarray Data. *Korean Journal of Applied Statistics* **25**:3, 415-422. [CrossRef]

39. CHUANYU SUN, XIAO-LIN WU, KENT A. WEIGEL, GUILHERME J. M. ROSA, STEWART BAUCK, BRENT W. WOODWARD, ROBERT D. SCHNABEL, JEREMY F. TAYLOR, DANIEL GIANOLA. 2012. An ensemble-based approach to imputation of moderate-density genotypes for genomic selection with application to Angus cattle. *Genetics Research* **94**:03, 133-150. [CrossRef]

40. Mian Huang, Weixin Yao. 2012. Mixture of Regression Models With Varying Mixing Proportions: A Semiparametric Approach. *Journal of the American Statistical Association* **107**:498, 711-724. [CrossRef]

41. Jaihak Chung, Vithala R Rao. 2012. A General Consumer Preference Model for Experience Products: Application to Internet Recommendation Services. *Journal of Marketing Research* **49**:3, 289-305. [CrossRef]

42. Saeed Masoudnia, Reza Ebrahimpour. 2012. Mixture of experts: a literature survey. *Artificial Intelligence Review* . [CrossRef]

43. MATTEO RE, GIORGIO VALENTINIEnsemble Methods **20124949**, . [CrossRef]

44. Shankaracharya, Devang Odedra, Medhavi Mallick, Prateek Shukla, Subir Samanta, Ambarish S. Vidyarthi. 2012. Java-Based Diabetes Type 2 Prediction Tool for Better Diagnosis. *Diabetes Technology & Therapeutics* **14**:3, 251-256. [CrossRef]

45. Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, Antti Sorjamaa. 2012. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications* . [CrossRef]

46. Sungmin Myoung, Ji Hong Chang, Kijun Song. 2012. A Mixture of Experts Model for the Diagnosis of Liver Cirrhosis by Measuring the Liver Stiffness. *Healthcare Informatics Research* **18**:1, 29. [CrossRef]

47. Ana R. Ricardo, Rui Oliveira, Svetlozar Velizarov, Maria A.M. Reis, João G. Crespo. 2012. Hybrid modeling of counterion mass transfer in a membrane-supported biofilm reactor. *Biochemical Engineering Journal* . [CrossRef]

48. Parthasarathy Subhasini, Bernadetta Kwintiana Ane, Dieter Roller, Marimuthu KrishnaveniIntelligent Classifiers Fusion for Enhancing Recognition of Genes and Protein Pattern of Hereditary Diseases 220-248. [CrossRef]

49. Yunxiao He, Chuanhai Liu. 2011. The dynamic 'expectation-conditional maximization either' algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* no-no. [CrossRef]

50. Sheng Jin, Xiaobo Qu, Dianhai Wang. 2011. Assessment of Expressway Traffic Safety Using Gaussian Mixture Model based on Time to Collision. *International Journal of Computational Intelligence Systems* **4**:6, 1122-1130. [CrossRef]

51. Nisar Ahmed, Mark Campbell. 2011. On estimating simple probabilistic discriminative models with subclasses. *Expert Systems with Applications* . [CrossRef]

52. O. Laurino, R. D'Abrusco, G. Longo, G. Riccio. 2011. Astroinformatics of galaxies and quasars: a new general method for photometric redshifts estimation. *Monthly Notices of the Royal Astronomical Society* no-no. [CrossRef]

53. Andrew R. Webb, Keith D. CopseyReferences 591-636. [CrossRef]

54. Tomasz Woloszynski, Marek Kurzynski. 2011. A probabilistic model of classifier competence for dynamic ensemble selection. *Pattern Recognition* **44**:10-11, 2656-2668. [CrossRef]

55. José Santos, Rui Oliveira, João CrespoHybrid Modeling of Membrane Processes 133-156. [CrossRef]

56. Yunwu Xiong, Rony Wallach, Alex Furman. 2011. Modeling multidimensional flow in wettable and water-repellent soils using artificial neural networks. *Journal of Hydrology* . [CrossRef]

57. Yan Yang, Jinwen Ma. 2011. Asymptotic Convergence Properties of the EM Algorithm for Mixture of Experts. *Neural Computation* **23**:8, 2140-2168. [Abstract] [Full Text] [PDF] [PDF Plus] [Supplementary Content]

58. John W Krakauer, Pietro Mazzoni. 2011. Human sensorimotor learning: adaptation, skill, and beyond. *Current Opinion in Neurobiology* **21**:4, 636-644. [CrossRef]

59. Lei XuLearning Algorithms for RBF Functions and Subspace Based Functions 1034-1065. [CrossRef]

60. Lei Xu, Shun-ichi AmariCombining Classifiers and Learning Mixture-of-Experts 243-252. [CrossRef]

61. Pramod P. Nair. 2011. A multigradient algorithm using a mixture of experts architecture for land cover classification of multisensor images. *International Journal of Remote Sensing* 1-9. [CrossRef]

62. Ana R. Ricardo, Rui Oliveira, Svetlozar Velizarov, Maria A.M. Reis, João G. Crespo. 2011. Multivariate statistical modelling of mass transfer in a membrane-supported biofilm reactor. *Process Biochemistry* . [CrossRef]

63. Dong-Hun Seo, Won-Don Lee. 2011. A New Ensemble System using Dynamic Weighting Method. *The Journal of the Korean Institute of Information and Communication Engineering* **15**:6, 1213-1220. [CrossRef]

64. Ezequiel López-Rubio. 2011. Stochastic approximation learning for mixtures of multivariate elliptical distributions. *Neurocomputing* . [CrossRef]

65. Isobel Claire Gormley, Thomas Brendan MurphyMixture of Experts Modelling with Social Science Applications 101-121. [CrossRef]

66. D. Michael TitteringtonThe EM Algorithm, Variational Approximations and Expectation Propagation for Mixtures 1-29. [CrossRef]

67. Feng Li, Mattias Villani, Robert KohnModelling Conditional Densities Using Finite Smooth Mixtures 123-144. [CrossRef]

68. Shawndra Hill, Noah Ready-Campbell. 2011. Expert Stock Picker: The Wisdom of (Experts in) Crowds. *International Journal of Electronic Commerce* **15**:3, 73-102. [CrossRef]

69. Chen Wu. 2011. Deriving collective intelligence from reviews on the social Web using a supervised learning approach. *Expert Systems with Applications* . [CrossRef]

70. Lei Yang, Nanning Zheng, Jie Yang. 2011. A unified context assessing model for object categorization. *Computer Vision and Image Understanding* **115**:3, 310-322. [CrossRef]

71. Lei Xu, Yanda Li. 2011. Machine learning and intelligence science: Sino-foreign interchange workshop IScIDE2010 (A). *Frontiers of Electrical and Electronic Engineering in China* **6**:1, 1-5. [CrossRef]

72. Peter Müller. 2011. A Product Partition Model With Regression on Covariates. *Journal of Computational and Graphical Statistics* **20**:1, 260-278. [CrossRef]

73. M. Olteanu, J. Rynkiewicz. 2011. Asymptotic properties of mixture-of-experts models. *Neurocomputing* . [CrossRef]

74. Maryam S. Mirian, Majid Nili Ahmadabadi, Babak N. Araabi, Roland R. Siegwart. 2011. Learning Active Fusion of Multiple Experts' Decisions: An Attention-Based Approach. *Neural Computation* **23**:2, 558-591. [Abstract] [Full Text] [PDF] [PDF Plus]

75. Paul D. Yoo, Bing Bing Zhou, Albert Y. ZomayaProtein Domain Boundary Prediction 501-519. [CrossRef]

76. Phaedon-Stelios Koutsourelakis, Elias Bilionis. 2011. Scalable Bayesian Reduced-Order Models for Simulating High-Dimensional Multiscale Dynamical Systems. *Multiscale Modeling & Simulation* **9**:1, 449-485. [CrossRef]

77. R. A. Mat Noor, Z. Ahmad, M. Mat Don, M. H. Uzir. 2010. Modelling and control of different types of polymerization processes using neural networks technique: A review. *The Canadian Journal of Chemical Engineering* **88**:6, 1065-1084. [CrossRef]

78. Zhouyu Fu, A Robles-Kelly, Jun Zhou. 2010. Mixing Linear SVMs for Nonlinear Classification. *IEEE Transactions on Neural Networks* **21**:12, 1963-1975. [CrossRef]

79. Abbas Khalili. 2010. New estimation and feature selection methods in mixture-of-experts models. *Canadian Journal of Statistics* **38**:4, 519-539. [CrossRef]

80. Robi Polikar, Joseph DePasquale, Hussein Syed Mohammed, Gavin Brown, Ludmilla I. Kuncheva. 2010. Learn++.MF: A random subspace approach for the missing feature problem. *Pattern Recognition* **43**:11, 3817-3832. [CrossRef]

81. Marios G. Philiastides, Guido Biele, Niki Vavatzanidis, Philipp Kazzer, Hauke R. Heekeren. 2010. Temporal dynamics of prediction error processing during reward-based decision making. *NeuroImage* **53**:1, 221-232. [CrossRef]

82. D.S. Young, D.R. Hunter. 2010. Mixtures of regressions with predictor-dependent mixing proportions. *Computational Statistics & Data Analysis* **54**:10, 2253-2266. [CrossRef]

83. Brian J. Reich, Howard D. Bondell, Lexin Li. 2010. Sufficient Dimension Reduction via Bayesian Mixture Modeling. *Biometrics* no-no. [CrossRef]

84. Ling Xu, Timothy Hanson, Edward J. Bedrick, Carla Restrepo. 2010. Hypothesis Tests on Mixture Model Components with Applications in Ecology and Agriculture. *Journal of Agricultural, Biological, and Environmental Statistics* **15**:3, 308-326. [CrossRef]

85. T. Hancock, I. Takigawa, H. Mamitsuka. 2010. Mining metabolic pathways through gene expression. *Bioinformatics* **26**:17, 2128-2135. [CrossRef]

86. Dimitri Bettebghor, Nathalie Bartoli, Stéphane Grihon, Joseph Morlier, Manuel Samuelides. 2010. Surrogate modeling approximation using a mixture of experts

based on EM joint estimation. *Structural and Multidisciplinary Optimization* . [CrossRef]

87. R. Pavón, F. Díaz, R. Laza, M.V. Luzón. 2010. Experimental evaluation of an automatic parameter setting system. *Expert Systems with Applications* **37**:7, 5224-5238. [CrossRef]
88. Hichem Frigui, Lijun Zhang, Paul D. Gader. 2010. Context-Dependent Multisensor Fusion and Its Application to Land Mine Detection. *IEEE Transactions on Geoscience and Remote Sensing* **48**:6, 2528-2543. [CrossRef]
89. Jun Namikawa, Jun Tani. 2010. Learning to imitate stochastic time series in a compositional way by chaos. *Neural Networks* **23**:5, 625-638. [CrossRef]
90. Elif Derya Übeyli, Konuralp Ilbay, Gul Ilbay, Deniz Sahin, Gur Akansel. 2010. Differentiation of Two Subtypes of Adult Hydrocephalus by Mixture of Experts. *Journal of Medical Systems* **34**:3, 281-290. [CrossRef]
91. K. A. Le Cao, E. Meugnier, G. J. McLachlan. 2010. Integrative mixture of experts to combine clinical factors and gene markers. *Bioinformatics* **26**:9, 1192-1198. [CrossRef]
92. Antonio CiampiPrediction Trees 1054-1060. [CrossRef]
93. Ona Wu, Rick M. Dijkhuizen, Alma Gregory Sorensen. 2010. Multiparametric Magnetic Resonance Imaging of Brain Disorders. *Topics in Magnetic Resonance Imaging* **21**:2, 129-138. [CrossRef]
94. I. Zaier, C. Shu, T.B.M.J. Ouarda, O. Seidou, F. Chebana. 2010. Estimation of ice thickness on lakes using artificial neural network ensembles. *Journal of Hydrology* **383**:3-4, 330-340. [CrossRef]
95. Lior Rokach. 2010. Ensemble-based classifiers. *Artificial Intelligence Review* **33**:1-2, 1-39. [CrossRef]
96. Paul Sajda, Robin I. Goldman, Mads Dyrholm, Truman R. BrownSignal Processing and Machine Learning for Single-trial Analysis of Simultaneously Acquired EEG and fMRI 311-334. [CrossRef]
97. Rajib Nayak, James Gomes. 2010. Generalized hybrid control synthesis for affine systems using sequential adaptive networks. *Journal of Chemical Technology & Biotechnology* **85**:1, 59-76. [CrossRef]
98. Mattias Villani, Robert Kohn, Paolo Giordani. 2009. Regression density estimation using smooth adaptive Gaussian mixtures. *Journal of Econometrics* **153**:2, 155-173. [CrossRef]
99. Mehran Emadi Andani, Fariba Bahrami, Parviz Jabehdar Maralani, Auke Jan Ijspeert. 2009. MODEM: a multi-agent hierarchical structure to model the human motor control system. *Biological Cybernetics* **101**:5-6, 361-377. [CrossRef]
100. I. Heintz, E. Fosler-Lussier, C. Brew. 2009. Discriminative Input Stream Combination for Conditional Random Field Phone Recognition. *IEEE Transactions on Audio, Speech, and Language Processing* **17**:8, 1533-1546. [CrossRef]

101. Clodoaldo A. M. Lima, André L. V. Coelho, Fernando J. Zuben. 2009. Pattern classification with mixtures of weighted least-squares support vector machine experts. *Neural Computing and Applications* **18**:7, 843-860. [CrossRef]

102. Minyoung Kim, V. Pavlovic. 2009. Discriminative Learning for Dynamic State Prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**:10, 1847-1861. [CrossRef]

103. Steven C.H. Hoi, Rong Jin, Michael R. Lyu. 2009. Batch Mode Active Learning with Applications to Text Categorization and Image Retrieval. *IEEE Transactions on Knowledge and Data Engineering* **21**:9, 1233-1248. [CrossRef]

104. Elif Derya Übeyli. 2009. Modified mixture of experts employing eigenvector methods and Lyapunov exponents for analysis of electroencephalogram signals. *Expert Systems* **26**:4, 339-354. [CrossRef]

105. Elif Derya Übeyli. 2009. Modified Mixture of Experts for Diabetes Diagnosis. *Journal of Medical Systems* **33**:4, 299-305. [CrossRef]

106. Adam Tashman, Robert Frey. 2009. Modeling risk in arbitrage strategies using finite mixtures. *Quantitative Finance* **9**:5, 495-503. [CrossRef]

107. Habtom W. Ressom, Getachew K. Befekadu, Mahlet G. Tadesse. 2009. Analysis of LC-MS Data Using probabilitic-based mixture regression models Analyse von LC-MS-Daten mit wahrscheinlichkeitsbasierter Mischung von Regressionsmodellen. *at - Automatisierungstechnik* **57**:9, 453-465. [CrossRef]

108. Elif Derya Ubeyli. 2009. Implementation of automated diagnostic systems: ophthalmic arterial disorders detection case. *International Journal of Systems Science* **40**:7, 669-683. [CrossRef]

109. Rozita A. Dara, Mohamed S. Kamel, Nayer Wanas. 2009. Data dependency in multiple classifier systems. *Pattern Recognition* **42**:7, 1260-1273. [CrossRef]

110. Zainal Ahmad, Rabiatul 'Adawiah Mat Noor, Jie Zhang. 2009. Multiple neural networks modeling techniques in process control: a review. *Asia-Pacific Journal of Chemical Engineering* **4**:4, 403-419. [CrossRef]

111. Ryunosuke Nishimoto, Jun Tani. 2009. Development of hierarchical structures for actions and motor imagery: a constructivist view from synthetic neuro-robotics study. *Psychological Research Psychologische Forschung* **73**:4, 545-558. [CrossRef]

112. Faicel Chamroukhi, Allou Samé, Gérard Govaert, Patrice Aknin. 2009. Time series modeling by a regression approach based on a latent process. *Neural Networks* **22**:5-6, 593-602. [CrossRef]

113. E.D. Ubeyli. 2009. Eigenvector Methods for Automated Detection of Electrocardiographic Changes in Partial Epileptic Patients. *IEEE Transactions on Information Technology in Biomedicine* **13**:4, 478-485. [CrossRef]

114. Denisse Hidalgo, Oscar Castillo, Patricia Melin. 2009. Type-1 and type-2 fuzzy inference systems as integration methods in modular neural networks for multimodal biometry and its optimization with genetic algorithms. *Information Sciences* **179**:13, 2123-2145. [CrossRef]

115. Olivier Cappé, Eric Moulines. 2009. On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71**:3, 593-613. [CrossRef]

116. J. Mora-Flórez, J. Cormane-Angarita, G. Ordóñez-Plata. 2009. k-means algorithm and mixture distributions for locating faults in power systems. *Electric Power Systems Research* **79**:5, 714-721. [CrossRef]

117. R. Ratcliff, M. G. Philiastides, P. Sajda. 2009. Quality of evidence for perceptual decision making is indexed by trial-to-trial variability of the EEG. *Proceedings of the National Academy of Sciences* **106**:16, 6539-6544. [CrossRef]

118. Elif Derya Übeyli. 2009. Features for analysis of electrocardiographic changes in partial epileptic patients. *Expert Systems with Applications* **36**:3, 6780-6789. [CrossRef]

119. E UBEYLI. 2009. Decision support systems for time-varying biomedical signals: EEG signals classification. *Expert Systems with Applications* **36**:2, 2275-2284. [CrossRef]

120. R PAVON, F DIAZ, R LAZA, V LUZON. 2009. Automatic parameter tuning with a Bayesian case-based reasoning system. A case of study. *Expert Systems with Applications* **36**:2, 3407-3420. [CrossRef]

121. P. Rojanavasu, Hai Huong Dam, H.A. Abbass, C. Lokan, O. Pinngern. 2009. A Self-Organized, Distributed, and Adaptive Rule-Based Induction System. *IEEE Transactions on Neural Networks* **20**:3, 446-459. [CrossRef]

122. Monica Adya, Edward J. Lusk, Moncef Balhadjali. 2009. Decomposition as a Complex-Skill Acquisition Strategy in Management Education: A Case Study in Business Forecasting. *Decision Sciences Journal of Innovative Education* **7**:1, 9-36. [CrossRef]

123. LAURENCE T. MALONEY, PASCAL MAMASSIAN. 2009. Bayesian decision theory as a model of human visual perception: Testing Bayesian transfer. *Visual Neuroscience* **26**:01, 147. [CrossRef]

124. COLIN FYFE, WESAM BARBAKH, WEI CHUAN OOI, HANSEOK KO. 2008. TOPOLOGICAL MAPPINGS OF VIDEO AND AUDIO DATA. *International Journal of Neural Systems* **18**:06, 481-489. [CrossRef]

125. J NAMIKAWA, J TANI. 2008. A model for learning to segment temporal sequences, utilizing a mixture of RNN experts together with adaptive variance. *Neural Networks* **21**:10, 1466-1475. [CrossRef]

126. Fernanda L. Minku, Teresa B. Ludermir. 2008. Clustering and co-evolution to construct neural network ensembles: An experimental study. *Neural Networks* **21**:9, 1363-1379. [CrossRef]

127. V. I. Gorodetskiy, S. V. Serebryakov. 2008. Methods and algorithms of collective recognition. *Automation and Remote Control* **69**:11, 1821-1851. [CrossRef]

128. Minwoo Jeong, Gary Geunbae Lee. 2008. Triangular-Chain Conditional Random Fields. *IEEE Transactions on Audio, Speech, and Language Processing* **16**:7, 1287-1302. [CrossRef]

129. G. Polzlbauer, T. Lidy, A. Rauber. 2008. Decision Manifolds—A Supervised Learning Algorithm Based on Self-Organization. *IEEE Transactions on Neural Networks* **19**:9, 1518-1530. [CrossRef]

130. KOSTAS FRAGOS, SPIROS PANETSOS. 2008. DISAMBIGUATION OF GREEK POLYSEMOUS WORDS USING HIERARCHICAL PROBABILISTIC NETWORKS AND A CHI-SQUARE FEATURE SELECTION STRATEGY. *International Journal on Artificial Intelligence Tools* **17**:04, 687-701. [CrossRef]

131. Dongbing Gu. 2008. Distributed EM Algorithm for Gaussian Mixtures in Sensor Networks. *IEEE Transactions on Neural Networks* **19**:7, 1154-1166. [CrossRef]

132. M.M. Islam, Xin Yao, S.M. Shahriar Nirjon, M.A. Islam, K. Murase. 2008. Bagging and Boosting Negatively Correlated Neural Networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **38**:3, 771-784. [CrossRef]

133. Sally A Wood, Robert Kohn, Remy Cottet, Wenxin Jiang, Martin Tanner. 2008. Locally Adaptive Nonparametric Binary Regression. *Journal of Computational and Graphical Statistics* **17**:2, 352-372. [CrossRef]

134. Elif Derya Übeyli. 2008. Implementing wavelet transform/mixture of experts network for analysis of electrocardiogram beats. *Expert Systems* **25**:2, 150-162. [CrossRef]

135. Taeryon Choi. 2008. Convergence of posterior distribution in the mixture of regressions. *Journal of Nonparametric Statistics* **20**:4, 337-351. [CrossRef]

136. J CLARKE, M WEST. 2008. Bayesian Weibull tree models for survival analysis of clinico-genomic data. *Statistical Methodology* **5**:3, 238-262. [CrossRef]

137. J. Peres, R. Oliveira, S. Feyo de Azevedo. 2008. Bioprocess hybrid parametric/nonparametric modelling based on the concept of mixture of experts. *Biochemical Engineering Journal* **39**:1, 190-206. [CrossRef]

138. E UBEYLI. 2008. Wavelet/mixture of experts network structure for EEG signals classification. *Expert Systems with Applications* **34**:3, 1954-1962. [CrossRef]

139. Yong Liu,, Xin Yao,. 2008. Nature Inspired Neural Network Ensemble Learning. *Journal of Intelligent Systems* **17**:supplement, 5. [CrossRef]

140. Norbert Tóth, Béla Pataki. 2008. Classification confidence weighted majority voting using decision tree classifiers. *International Journal of Intelligent Computing and Cybernetics* **1**:2, 169-192. [CrossRef]

141. Xia Hong, Sheng Chen, Chris J. Harris. 2008. A Forward-Constrained Regression Algorithm for Sparse Kernel Density Estimation. *IEEE Transactions on Neural Networks* **19**:1, 193-198. [CrossRef]

142. R POLIKAR, A TOPALIS, D PARIKH, D GREEN, J FRYMIARE, J KOUNIOS, C CLARK. 2008. An ensemble based data fusion approach for early diagnosis of Alzheimer's disease. *Information Fusion* **9**:1, 83-95. [CrossRef]

143. Minh Ha Nguyen, H.A. Abbass, R.I. McKay. 2008. Analysis of CCME: Coevolutionary Dynamics, Automatic Problem Decomposition, and Regularization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **38**:1, 100-109. [CrossRef]

144. Vanessa Gomez-Verdejo, JerÓnimo Arenas-Garcia, AnÍbal R. Figueiras-Vidal. 2008. A Dynamically Adjusted Mixed Emphasis Method for Building Boosting Ensembles. *IEEE Transactions on Neural Networks* **19**:1, 3-17. [CrossRef]

145. A MOJSILOVIC, B RAY, R LAWRENCE, S TAKRITI. 2007. A logistic regression framework for information technology outsourcing lifecycle management. *Computers & Operations Research* **34**:12, 3609-3627. [CrossRef]

146. Estevam R. Hruschka, Eduardo R. Hruschka, Nelson F. F. Ebecken. 2007. Bayesian networks for imputation in classification problems. *Journal of Intelligent Information Systems* **29**:3, 231-252. [CrossRef]

147. Cristian Sminchisescu, Atul Kanaujia, Dimitris N. Metaxas. 2007. BM^{3}E : Discriminative Density Propagation for Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**:11, 2030-2044. [CrossRef]

148. Shun-ichi Amari. 2007. Integration of Stochastic Models by Minimizing α-Divergence. *Neural Computation* **19**:10, 2780-2796. [Abstract] [PDF] [PDF Plus]

149. Kin-Chung Wong, Wei-Yang Lin, Yu Hen Hu, Nigel Boston, Xueqin Zhang. 2007. Optimal Linear Combination of Facial Regions for Improving Identification Performance. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **37**:5, 1138-1148. [CrossRef]

150. S COHEN, L ROKACH, O MAIMON. 2007. Decision-tree instance-space decomposition with grouped gain-ratio. *Information Sciences* **177**:17, 3592-3612. [CrossRef]

151. Gregor Gregorcic, Gordon Lightbody. 2007. Local Model Network Identification With Gaussian Processes. *IEEE Transactions on Neural Networks* **18**:5, 1404-1423. [CrossRef]

152. Reza Ebrahimpour, Ehsanollah Kabir, Mohammad Reza Yousefi. 2007. Face Detection Using Mixture of MLP Experts. *Neural Processing Letters* **26**:1, 69-82. [CrossRef]

153. M. Ortega-Moral, D. Gutiérrez-González, M. L. De-Pablo, J. Cid-Sueiro. 2007. Training Classifiers for Tree-structured Categories with Partially Labeled Data. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology* **48**:1-2, 53-65. [CrossRef]

154. THOMAS R. SHULTZ, FRANÇOIS RIVEST, LÁSZLÓ EGRI, JEAN-PHILIPPE THIVIERGE, FRÉDÉRIC DANDURAND. 2007. COULD KNOWLEDGE-BASED NEURAL LEARNING BE USEFUL IN

DEVELOPMENTAL ROBOTICS? THE CASE OF KBCC. *International Journal of Humanoid Robotics* **04**:02, 245-279. [CrossRef]

155. C LIMA, A COELHO, F VONZUBEN. 2007. Hybridizing mixtures of experts with support vector machines: Investigation into nonlinear dynamic systems identification. *Information Sciences* **177**:10, 2049-2074. [CrossRef]
156. Giovanna Jona Lasinio, Fabio Divino, Annibale Biggeri. 2007. Environmental risk assessment in the Tuscany region: a proposal. *Environmetrics* **18**:3, 315-332. [CrossRef]
157. A MELLIT, M BENGHANEM. 2007. Sizing of stand-alone photovoltaic systems using neural network adaptive model. *Desalination* **209**:1-3, 64-72. [CrossRef]
158. David B. Dunson, Natesh Pillai, Ju-Hyun Park. 2007. Bayesian density regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69**:2, 163-183. [CrossRef]
159. Devi Parikh, Robi Polikar. 2007. An Ensemble-Based Incremental Learning Approach to Data Fusion. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **37**:2, 437-450. [CrossRef]
160. Lucy Marshall, David Nott, Ashish Sharma. 2007. Towards dynamic catchment modelling: a Bayesian hierarchical mixtures of experts framework. *Hydrological Processes* **21**:7, 847-861. [CrossRef]
161. Colin Fyfe. 2007. Two topographic maps for data visualisation. *Data Mining and Knowledge Discovery* **14**:2, 207-224. [CrossRef]
162. Elif Derya Übeyli. 2007. Comparison of different classification algorithms in clinical decision-making. *Expert Systems* **24**:1, 17-31. [CrossRef]
163. Tamara Hayes, Misha Pavel, Nicole Larimer, Ishan Tsay, John Nutt, Andre Adami. 2007. Distributed Healthcare: Simultaneous Assessment of Multiple Individuals. *IEEE Pervasive Computing* **6**:1, 36-43. [CrossRef]
164. Michael Defoin-Platel, Malik Chami. 2007. How ambiguous is the inverse problem of ocean color in coastal waters?. *Journal of Geophysical Research* **112**:C3. . [CrossRef]
165. P HARTONO, S HASHIMOTO. 2007. Learning from imperfect data. *Applied Soft Computing* **7**:1, 353-363. [CrossRef]
166. Eiji Uchibe, Kenji Doya. 2007. *The Brain & Neural Networks* **14**:4, 293-304. [CrossRef]
167. Lucy Marshall, Ashish Sharma, David Nott. 2007. A single model ensemble versus a dynamic modeling platform: Semi-distributed rainfall runoff modeling in a Hierarchical Mixtures of Experts framework. *Geophysical Research Letters* **34**:1. . [CrossRef]
168. Justin C. Sanchez, José C. Principe. 2007. Brain–Machine Interface Engineering. *Synthesis Lectures on Biomedical Engineering* **2**:1, 1-234. [CrossRef]

169. Lucy Marshall, Ashish Sharma, David Nott. 2006. Modeling the catchment via mixtures: Issues of model specification and validation. *Water Resources Research* **42**:11, n/a-n/a. [CrossRef]

170. I. Guler, E.D. Ubeyli. 2006. Automated Diagnostic Systems With Diverse and Composite Features for Doppler Ultrasound Signals. *IEEE Transactions on Biomedical Engineering* **53**:10, 1934-1942. [CrossRef]

171. Lior Rokach. 2006. Decomposition methodology for classification tasks: a meta decomposer framework. *Pattern Analysis and Applications* **9**:2-3, 257-271. [CrossRef]

172. D. M. TitteringtonNeural Networks . [CrossRef]

173. Woo-Kyung Choi, Sang-Hyung Ha, Seong-Joo Kim, Yong-Taek Kim, Hong-Tae Jeon. 2006. The Intelligent Control System for Biped Robot Using Hierarchical Mixture of Experts. *Journal of Fuzzy Logic and Intelligent Systems* **16**:4, 389-395. [CrossRef]

174. G.-B. Huang, L. Chen, C.-K. Siew. 2006. Universal Approximation Using Incremental Constructive Feedforward Networks With Random Hidden Nodes. *IEEE Transactions on Neural Networks* **17**:4, 879-892. [CrossRef]

175. I AUTIO. 2006. Using natural class hierarchies in multi-class visual classification. *Pattern Recognition* **39**:7, 1290-1299. [CrossRef]

176. D. Wedge, D. Ingram, D. Mclean, C. Mingham, Z. Bandar. 2006. On Global–Local Artificial Neural Networks for Function Approximation. *IEEE Transactions on Neural Networks* **17**:4, 942-952. [CrossRef]

177. W WANG, P GELDER, J VRIJLING, J MA. 2006. Forecasting daily streamflow using hybrid ANN models. *Journal of Hydrology* **324**:1-4, 383-399. [CrossRef]

178. C. Alippi, F. Scotti. 2006. Exploiting Application Locality to Design Low-Complexity, Highly Performing, and Power-Aware Embedded Classifiers. *IEEE Transactions on Neural Networks* **17**:3, 745-754. [CrossRef]

179. RÓMer Rosales, Stan Sclaroff. 2006. Combining Generative and Discriminative Models in a Framework for Articulated Pose Estimation. *International Journal of Computer Vision* **67**:3, 251-276. [CrossRef]

180. Robi PolikarPattern Recognition . [CrossRef]

181. B COBB, P SHENOY. 2006. Inference in hybrid Bayesian networks with mixtures of truncated exponentials. *International Journal of Approximate Reasoning* **41**:3, 257-286. [CrossRef]

182. Mingyang Xu, Michael W. Golay. 2006. Data-guided model combination by decomposition and aggregation. *Machine Learning* **63**:1, 43-67. [CrossRef]

183. J. Zhang, Q. Jin, Y. Xu. 2006. Inferential Estimation of Polymer Melt Index Using Sequentially Trained Bootstrap Aggregated Neural Networks. *Chemical Engineering & Technology* **29**:4, 442-448. [CrossRef]

184. G FENG. 2006. Eye movements as time-series random variables: A stochastic model of eye movement control in reading. *Cognitive Systems Research* **7**:1, 70-95. [CrossRef]

185. D LOYOLAR. 2006. Applications of neural network methods to the processing of earth observation satellite data. *Neural Networks* **19**:2, 168-177. [CrossRef]

186. Marcus FreanConnectionist Architectures: Optimization . [CrossRef]

187. J ZHANG. 2006. Improved on-line process fault diagnosis through information fusion in multiple neural networks. *Computers & Chemical Engineering* **30**:3, 558-571. [CrossRef]

188. Yang Ge, Wenxin Jiang. 2006. On Consistency of Bayesian Inference with Mixtures of Logistic Regression. *Neural Computation* **18**:1, 224-243. [Abstract] [PDF] [PDF Plus]

189. J. Peres, F. Freitas, MAM Reis, S. Feyo de Azevedo, R. OliveiraHybrid modular mechanistic/ANN modelling of a wastewater phosphorus removal process **21**, 1717-1722. [CrossRef]

190. Alejandro Villagran, Gabriel HuertaBayesian Inference on Mixture-of-Experts for Estimation of Stochastic Volatility **20**, 277-296. [CrossRef]

191. Alexandre X. Carvalho, Martin A. Tanner. 2006. Modeling nonlinearities with mixtures-of-experts of time series models. *International Journal of Mathematics and Mathematical Sciences* **2006**, 1-23. [CrossRef]

192. Biswanath BhattacharyaReferences . [CrossRef]

193. Z AHMAD, J ZHANG. 2005. Combination of multiple neural networks using data fusion techniques for enhanced nonlinear process modelling. *Computers & Chemical Engineering* **30**:2, 295-308. [CrossRef]

194. C. C Holmes, D. G. T Denison, S Ray, B. K Mallick. 2005. Bayesian Prediction via Partitioning. *Journal of Computational and Graphical Statistics* **14**:4, 811-830. [CrossRef]

195. G.-B. Huang, K.Z. Mao, C.-K. Siew, D.-S. Huang. 2005. Fast Modular Network Implementation for Support Vector Machines. *IEEE Transactions on Neural Networks* **16**:6, 1651-1663. [CrossRef]

196. Elif Derya Übeyli. 2005. A Mixture of Experts Network Structure for Breast Cancer Diagnosis. *Journal of Medical Systems* **29**:5, 569-579. [CrossRef]

197. Andreas Lindemann, Christian L. Dunis, Paulo Lisboa. 2005. Probability distributions and leveraged trading strategies: an application of Gaussian mixture models to the Morgan Stanley Technology Index Tracking Fund. *Quantitative Finance* **5**:5, 459-474. [CrossRef]

198. Feng Zhang, Bani Mallick, Zhujun Weng. 2005. A Bayesian method for identifying independent sources of non-random spatial patterns. *Statistics and Computing* **15**:4, 329-339. [CrossRef]

199. G ADAMI, P AVESANI, D SONA. 2005. Clustering documents into a web directory for bootstrapping a supervised classification. *Data & Knowledge Engineering* **54**:3, 301-325. [CrossRef]

200. Andreas Lindemann, Christian L. Dunis, Paulo Lisboa. 2005. Level estimation, classification and probability distribution architectures for trading the EUR/USD exchange rate. *Neural Computing and Applications* **14**:3, 256-271. [CrossRef]

201. A MELLIT, M BENGHANEM, A ARAB, A GUESSOUM. 2005. An adaptive artificial neural network model for sizing stand-alone photovoltaic systems: application for isolated sites in Algeria. *Renewable Energy* **30**:10, 1501-1524. [CrossRef]

202. B. D. RipleyComputer-Intensive Methods . [CrossRef]

203. V. Cherkassky, Y. Ma. 2005. Multiple Model Regression Estimation. *IEEE Transactions on Neural Networks* **16**:4, 785-798. [CrossRef]

204. J.I. Arribas, J. Cid-Sueiro. 2005. A Model Selection Algorithm for a Posteriori Probability Estimation With Neural Networks. *IEEE Transactions on Neural Networks* **16**:4, 799-809. [CrossRef]

205. Abedalrazq Khalil, Mohammad N. Almasri, Mac McKee, Jagath J. Kaluarachchi. 2005. Applicability of statistical learning algorithms in groundwater quality modeling. *Water Resources Research* **41**:5, n/a-n/a. [CrossRef]

206. 2005. Identification of piecewise affine systems based on statistical clustering technique. *Automatica* **41**:5, 905-913. [CrossRef]

207. Kamal Morad, Brent R. Young, William Y. Svrcek. 2005. Rectification of plant measurements using a statistical framework. *Computers & Chemical Engineering* **29**:5, 919-940. [CrossRef]

208. Zainal Ahmad, Jie Zhang. 2005. Bayesian selective combination of multiple neural networks for improving long-range predictions in nonlinear process modelling. *Neural Computing and Applications* **14**:1, 78-87. [CrossRef]

209. Alexandre X. Carvalho, Martin A. Tanner. 2005. Modeling nonlinear time series with local mixtures of generalized linear models. *Canadian Journal of Statistics* **33**:1, 97-113. [CrossRef]

210. Patricia Melin, Cristina Felix, Oscar Castillo. 2005. Face recognition using modular neural networks and the fuzzy Sugeno integral for response integration. *International Journal of Intelligent Systems* **20**:2, 275-291. [CrossRef]

211. Carlos Ordonez, Edward Omiecinski. 2005. Accelerating EM clustering to find high-quality solutions. *Knowledge and Information Systems* **7**:2, 135-157. [CrossRef]

212. A.X. Carvalho, M.A. Tanner. 2005. Mixtures-of-Experts of Autoregressive Time Series: Asymptotic Normality and Model Specification. *IEEE Transactions on Neural Networks* **16**:1, 39-56. [CrossRef]

213. A HAGEN. 2005. Recent advances in the multi-stream HMM/ANN hybrid approach to noise robust ASR. *Computer Speech & Language* **19**:1, 3-30. [CrossRef]

214. Gabriel MateescuA New Model for Probabilistic Information Retrieval on the Web **177**, 327-347. [CrossRef]

215. Andreas Lindemann, Christian L. Dunis, Paulo Lisboa. 2004. Probability distributions, trading strategies and leverage: an application of Gaussian mixture models. *Journal of Forecasting* **23**:8, 559-585. [CrossRef]

216. R PAINE, J TANI. 2004. Motor primitive and sequence self-organization in a hierarchical recurrent neural network. *Neural Networks* **17**:8-9, 1291-1309. [CrossRef]

217. A. Sun, E.-P. Lim, W.-K. Ng, J. Srivastava. 2004. Blocking reduction strategies in hierarchical text classification. *IEEE Transactions on Knowledge and Data Engineering* **16**:10, 1305-1308. [CrossRef]

218. M VERSACE. 2004. Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks. *Expert Systems with Applications* **27**:3, 417-425. [CrossRef]

219. R HASKELL. 2004. Geno-fuzzy classification trees. *Pattern Recognition* **37**:8, 1653-1659. [CrossRef]

220. C. Ordonez, E. Omiecinski. 2004. Efficient disk-based K-means clustering for relational databases. *IEEE Transactions on Knowledge and Data Engineering* **16**:8, 909-921. [CrossRef]

221. CHRISTOPH KÖNIG, GIUSEPPINA GINI, MARIAN CRACIUN, EMILIO BENFENATI. 2004. MULTICLASS CLASSIFIER FROM A COMBINATION OF LOCAL EXPERTS: TOWARD DISTRIBUTED COMPUTATION FOR REAL-PROBLEM CLASSIFIERS. *International Journal of Pattern Recognition and Artificial Intelligence* **18**:05, 801-817. [CrossRef]

222. T. Rohlfing, D.B. Russakoff, C.R. Maurer. 2004. Performance-Based Classifier Combination in Atlas-Based Image Segmentation Using Expectation-Maximization Parameter Estimation. *IEEE Transactions on Medical Imaging* **23**:8, 983-994. [CrossRef]

223. Jayanta Basak, Ravi Kothari. 2004. A Classification Paradigm for Distributed Vertically Partitioned Data. *Neural Computation* **16**:7, 1525-1544. [Abstract] [PDF] [PDF Plus]

224. L. Xu. 2004. Advances on BYY Harmony Learning: Information Theoretic Perspective, Generalized Projection Geometry, and Independent Factor Autodetermination. *IEEE Transactions on Neural Networks* **15**:4, 885-902. [CrossRef]

225. G. Fang, W. Gao, D. Zhao. 2004. Large Vocabulary Sign Language Recognition Based on Fuzzy Decision Trees. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **34**:3, 305-314. [CrossRef]

226. D.R. Martin, C.C. Fowlkes, J. Malik. 2004. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**:5, 530-549. [CrossRef]

227. M.A. Moussa. 2004. Combining Expert Neural Networks Using Reinforcement Feedback for Learning Primitive Grasping Behavior. *IEEE Transactions on Neural Networks* **15**:3, 629-638. [CrossRef]

228. D. Malerba, F. Esposito, M. Ceci, A. Appice. 2004. Top-down induction of model trees with regression and splitting nodes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**:5, 612-625. [CrossRef]

229. S.-K. Ng, G.J. McLachlan. 2004. Using the EM Algorithm to Train Neural Networks: Misconceptions and a New Algorithm for Multiclass Classification. *IEEE Transactions on Neural Networks* **15**:3, 738-749. [CrossRef]

230. G. Biau, L. Devroye, G. Biau, L. Devroye. 2004. A Note on Density Model Size Testing. *IEEE Transactions on Information Theory* **50**:3, 576-581. [CrossRef]

231. Hyun-Chul Kim, Daijin Kim, Sung Yang Bang, Sang-Youn Lee. 2004. Face recognition using the second-order mixture-of-eigenfaces method. *Pattern Recognition* **37**:2, 337-349. [CrossRef]

232. 2004. Learning for Environment and Behavior Pattern Using Recurrent Modular Neural Network Based on Estimated Emotion. *Journal of Fuzzy Logic and Intelligent Systems* **14**:1, 9-14. [CrossRef]

233. J. Peres, R. Oliveira, L.S. Serafim, P. Lemos, M.A. Reis, S. Feyo de AzevedoHybrid Modelling of a PHA Production Process Using Modular Neural Networks **18**, 733-738. [CrossRef]

234. References 367-382. [CrossRef]

235. Ori Rosen, Ayala Cohen. 2003. Analysis of growth curves via mixtures. *Statistics in Medicine* **22**:23, 3641-3654. [CrossRef]

236. H Kim. 2003. Constructing support vector machine ensemble. *Pattern Recognition* **36**:12, 2757-2767. [CrossRef]

237. A AlShaher. 2003. Learning mixtures of point distribution models with the EM algorithm. *Pattern Recognition* **36**:12, 2805-2818. [CrossRef]

238. B Luo. 2003. A unified framework for alignment and correspondence. *Computer Vision and Image Understanding* **92**:1, 26-55. [CrossRef]

239. C. K. Tham, C. K. Heng, W. C. Chin. 2003. Predicting Risk of Coronary Artery Disease from DNA Microarray-Based Genotyping Using Neural Networks and Other Statistical Analysis Tool. *Journal of Bioinformatics and Computational Biology* **01**:03, 521-539. [CrossRef]

240. A. Garg, V. Pavlovic, J.M. Rehg. 2003. Boosted learning in dynamic bayesian networks for multimodal speaker detection. *Proceedings of the IEEE* **91**:9, 1355-1369. [CrossRef]

241. S. Raudys. 2003. Experts' boasting in trainable fusion rules. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**:9, 1178-1182. [CrossRef]

242. Ingo Ruczinski, Charles Kooperberg, Michael LeBlanc. 2003. Logic Regression. *Journal of Computational and Graphical Statistics* **12**:3, 475-511. [CrossRef]

243. Z Zhang. 2003. EM algorithms for Gaussian mixtures with split-and-merge operation. *Pattern Recognition* **36**:9, 1973-1983. [CrossRef]

244. Md.M. Islam, Xin Yao, K. Murase. 2003. A constructive algorithm for training cooperative neural network ensembles. *IEEE Transactions on Neural Networks* **14**:4, 820-834. [CrossRef]

245. M.K. Titsias, A. Likas. 2003. Class conditional density estimation using mixtures with constrained component sharing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**:7, 924-928. [CrossRef]

246. Nan Xie, H. Leung, Hing Chan. 2003. A multiple-model prediction approach for sea clutter modeling. *IEEE Transactions on Geoscience and Remote Sensing* **41**:6, 1491-1502. [CrossRef]

247. H KIM. 2003. An efficient model order selection for PCA mixture model. *Pattern Recognition Letters* **24**:9-10, 1385-1393. [CrossRef]

248. Chee Peng Lim, R.F. Harrison. 2003. Online pattern classification with multiple neural network systems: an experimental study. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* **33**:2, 235-247. [CrossRef]

249. H Kim. 2003. Extensions of LDA by PCA mixture model and class-wise features. *Pattern Recognition* **36**:5, 1095-1105. [CrossRef]

250. Robert A. Jacobs, Melissa Dominguez. 2003. Visual Development and the Acquisition of Motion Velocity Sensitivities. *Neural Computation* **15**:4, 761-781. [Abstract] [PDF] [PDF Plus]

251. 2003. Mobile robot control by MNN using optimal EN. *Journal of Fuzzy Logic and Intelligent Systems* **13**:2, 186-191. [CrossRef]

252. Liu Yong, Zou Xiu-fen. 2003. Analysis of negative correlation learning. *Wuhan University Journal of Natural Sciences* **8**:1, 165-175. [CrossRef]

253. Liu Yong, Zou Xiu-fen. 2003. From designing a single neural network to designing neural network ensembles. *Wuhan University Journal of Natural Sciences* **8**:1, 155-164. [CrossRef]

254. M LAZARO. 2003. A new EM-based training algorithm for RBF networks. *Neural Networks* **16**:1, 69-77. [CrossRef]

255. J. Peres, R. Oliveira, S. Feyo de AzevedoModelling cells reaction kinetics with artificial neural networks: A comparison of three network architectures **14**, 839-844. [CrossRef]

256. Antonio Torralba. 2003. Modeling global scene factors in attention. *Journal of the Optical Society of America A* **20**:7, 1407. [CrossRef]

257. Junghui Chen, Yuezhi Yea. 2003. Design Pole Placement Controller Using Linearized Neural Networks for MISO Systems. *JOURNAL OF CHEMICAL ENGINEERING OF JAPAN* **36**:8, 1005-1011. [CrossRef]

258. M CARCASSONI, E HANCOCK. 2003. Spectral correspondence for point pattern matching. *Pattern Recognition* **36**:1, 193-204. [CrossRef]

259. LIANG-WEI HO, GARY G. YEN. 2002. RECONFIGURABLE CONTROL SYSTEM DESIGN FOR FAULT DIAGNOSIS AND ACCOMMODATION. *International Journal of Neural Systems* **12**:06, 497-520. [CrossRef]

260. A. Kehagias, V. Petridis. 2002. Predictive modular neural networks for unsupervised segmentation of switching time series: the data allocation problem. *IEEE Transactions on Neural Networks* **13**:6, 1432-1449. [CrossRef]

261. L XU. 2002. BYY harmony learning, structural RPCL, and topological self-organizing on mixture models1. *Neural Networks* **15**:8-9, 1125-1151. [CrossRef]

262. Michalis K. Titsias, Aristidis Likas. 2002. Mixture of Experts Classification Using a Hierarchical Mixture Model. *Neural Computation* **14**:9, 2221-2244. [Abstract] [PDF] [PDF Plus]

263. A. Torralba, A. Oliva. 2002. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**:9, 1226-1238. [CrossRef]

264. Ji Ming, P. Jancovic, F.J. Smith. 2002. Robust speech recognition using probabilistic union models. *IEEE Transactions on Speech and Audio Processing* **10**:6, 403-414. [CrossRef]

265. GIULIANO ARMANO, ANDREA MURRU, FABIO ROLI. 2002. STOCK MARKET PREDICTION BY A MIXTURE OF GENETIC-NEURAL EXPERTS. *International Journal of Pattern Recognition and Artificial Intelligence* **16**:05, 501-526. [CrossRef]

266. E. Mizutani, K. Nishio. 2002. Multi-illuminant color reproduction for electronic cameras via CANFIS neuro-fuzzy modular network device characterization. *IEEE Transactions on Neural Networks* **13**:4, 1009-1022. [CrossRef]

267. F Acernese, F Barone, M de Rosa, R De Rosa, A Eleuteri, L Milano, R Tagliaferri. 2002. A neural network-based approach to noise identification of interferometric GW antennas: the case of the 40 m Caltech laser interferometer. *Classical and Quantum Gravity* **19**:12, 3293-3307. [CrossRef]

268. Akihiro Minagawa, Norio Tagawa, Toshiyuki Tanaka. 2002. SMEM Algorithm Is Not Fully Compatible with Maximum-Likelihood Framework. *Neural Computation* **14**:6, 1261-1266. [Abstract] [PDF] [PDF Plus]

269. M. Pardo, G. Sberveglieri. 2002. Learning from data: a tutorial with emphasis on modern pattern recognition methods. *IEEE Sensors Journal* **2**:3, 203-217. [CrossRef]

270. A Sierra. 2002. High-order Fisher's discriminant analysis. *Pattern Recognition* **35**:6, 1291-1302. [CrossRef]

271. Z Zhou. 2002. Ensembling neural networks: Many could be better than all. *Artificial Intelligence* **137**:1-2, 239-263. [CrossRef]

272. Sheng-Uei Guan, Shanchun Li. 2002. Parallel growing and training of neural networks using output parallelism. *IEEE Transactions on Neural Networks* **13**:3, 542-550. [CrossRef]

273. Antonio Ciampi, Andr# Couturier, Shaolin Li. 2002. Prediction trees with soft nodes for binary outcomes. *Statistics in Medicine* **21**:8, 1145-1165. [CrossRef]

274. E. Moreau, C. Mallet, S. Thiria, B. Mabboux, F. Badran, C. Klapisz. 2002. Atmospheric Liquid Water Retrieval Using a Gated Experts Neural Network. *Journal of Atmospheric and Oceanic Technology* **19**:4, 457-467. [CrossRef]

275. C.C. Chibelushi, F. Deravi, J.S.D. Mason. 2002. A review of speech-based bimodal recognition. *IEEE Transactions on Multimedia* **4**:1, 23-37. [CrossRef]

276. S. C. Mcloone, S. Mcginnity, G. W. Irwin. 2002. Comparison of two construction algorithms for local model networks. *International Journal of Systems Science* **33**:13, 1059-1072. [CrossRef]

277. R. Polikar, L. Upda, S.S. Upda, V. Honavar. 2001. Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* **31**:4, 497-508. [CrossRef]

278. M. Pardo, G. Sberveglieri, A. Taroni, F. Masulli, G. Valentini. 2001. Decompositive classification models for electronic noses. *Analytica Chimica Acta* **446**:1-2, 221-230. [CrossRef]

279. Tze Leung Lai, Samuel Po-Shing Wong. 2001. Stochastic Neural Networks With Applications to Nonlinear Time Series. *Journal of the American Statistical Association* **96**:455, 968-981. [CrossRef]

280. D. G. T. Denison, P. Dellaportas, B. K. Mallick. 2001. Wind speed prediction in a complex terrain. *Environmetrics* **12**:6, 499-515. [CrossRef]

281. N.S.V. Rao. 2001. On fusers that perform better than best sensor. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**:8, 904-909. [CrossRef]

282. G Bontempi. 2001. The local paradigm for modeling and control: from neuro-fuzzy to lazy learning. *Fuzzy Sets and Systems* **121**:1, 59-72. [CrossRef]

283. W Soh. 2001. Modular neural networks for multi-service connection admission control. *Computer Networks* **36**:2-3, 181-202. [CrossRef]

284. Yuhong Yang. 2001. Adaptive Regression by Mixing. *Journal of the American Statistical Association* **96**:454, 574-588. [CrossRef]

285. Marcos M Campos, Gail A Carpenter. 2001. S-TREE: self-organizing trees for data clustering and online vector quantization. *Neural Networks* **14**:4-5, 505-525. [CrossRef]

286. J. Cid-Sueiro, A.R. Figueiras-Vidal. 2001. On the structure of strict sense Bayesian cost functions and its applications. *IEEE Transactions on Neural Networks* **12**:3, 445-455. [CrossRef]

287. J PERES, R OLIVEIRA, S FEYODEAZEVEDO. 2001. Knowledge based modular networks for process modelling and control. *Computers & Chemical Engineering* **25**:4-6, 783-791. [CrossRef]

288. Z. Yang, A. Shimpi, D. Purves. 2001. A wholly empirical explanation of perceived motion. *Proceedings of the National Academy of Sciences* **98**:9, 5252-5257. [CrossRef]

289. I. M. GALVÁN, P. ISASI, R. ALER, J. M. VALLS. 2001. A SELECTIVE LEARNING METHOD TO IMPROVE THE GENERALIZATION OF MULTILAYER FEEDFORWARD NEURAL NETWORKS. *International Journal of Neural Systems* **11**:02, 167-177. [CrossRef]

290. H. Li, Y. Wang, K.J.R. Liu, S.-C.B. Lo, M.T. Freedman. 2001. Computerized radiographic mass detection. II. Decision support by featured database visualization and modular neural networks. *IEEE Transactions on Medical Imaging* **20**:4, 302-313. [CrossRef]

291. C Harris. 2001. State estimation and multi-sensor data fusion using data-based neurofuzzy local linearisation process models. *Information Fusion* **2**:1, 17-29. [CrossRef]

292. Thomas Shultz, Francois Rivest. 2001. Knowledge-based cascade-correlation: Using knowledge to speed learning. *Connection Science* **13**:1, 43-72. [CrossRef]

293. Gabriel Huerta, Wenxin Jiang, Martin A Tanner. 2001. Discussion. *Journal of Computational and Graphical Statistics* **10**:1, 82-89. [CrossRef]

294. Yuan-Fu Liao, Sin-Horng Chen. 2001. A modular RNN-based method for continuous Mandarin speech recognition. *IEEE Transactions on Speech and Audio Processing* **9**:3, 252-263. [CrossRef]

295. Hsin-Chia Fu, Yen-Po Lee, Cheng-Chin Chiang, Hsiao-Tien Pao. 2001. Divide-and-conquer learning and modular perceptron networks. *IEEE Transactions on Neural Networks* **12**:2, 250-263. [CrossRef]

296. ZOUBIN GHAHRAMANI. 2001. AN INTRODUCTION TO HIDDEN MARKOV MODELS AND BAYESIAN NETWORKS. *International Journal of Pattern Recognition and Artificial Intelligence* **15**:01, 9-42. [CrossRef]

297. Lei Xu. 2001. Best Harmony, Unified RPCL and Automated Model Selection for Unsupervised and Supervised Learning on Gaussian Mixtures, Three-Layer Nets and ME-RBF-SVM Models. *International Journal of Neural Systems* **11**:01, 43-69. [CrossRef]

298. X. Dai. 2001. CMA-based nonlinear blind equaliser modelled by a two-layer feedforward neural network. *IEE Proceedings - Communications* **148**:4, 243. [CrossRef]

299. Qiang Gan, C.J. Harris. 2001. A hybrid learning scheme combining EM and MASMOD algorithms for fuzzy local linearization modeling. *IEEE Transactions on Neural Networks* **12**:1, 43-53. [CrossRef]

300. J PENG. 2001. Local discriminative learning for pattern recognition. *Pattern Recognition* **34**:1, 139-150. [CrossRef]

301. Laurent Girin, Jean-Luc Schwartz, Gang Feng. 2001. Audio-visual enhancement of speech in noise. *The Journal of the Acoustical Society of America* **109**:6, 3007. [CrossRef]

302. Holger Luczak, Christopher Schlick, Alexander Kuenzer, Frank Ohmann. 2001. Syntactic user modelling with stochastic processes. *Theoretical Issues in Ergonomics Science* **2**:2, 97-123. [CrossRef]

303. C.J. Harris, X. Hong. 2001. Neurofuzzy mixture of experts network parallel learning and model construction algorithms. *IEE Proceedings - Control Theory and Applications* **148**:6, 456. [CrossRef]

304. M.A. Tanner, R.A. JacobsNeural Networks and Related Statistical Latent Variable Models 10526-10534. [CrossRef]

305. N-J Huh, J-H Oh, K Kang. 2000. *Journal of Physics A: Mathematical and General* **33**:48, 8663-8672. [CrossRef]

306. Jinwen Ma, Lei Xu, Michael I. Jordan. 2000. Asymptotic Convergence Rate of the EM Algorithm for Gaussian Mixtures. *Neural Computation* **12**:12, 2881-2907. [Abstract] [PDF] [PDF Plus]

307. Dirk Husmeier. 2000. The Bayesian Evidence Scheme for Regularizing Probability-Density Estimating Neural Networks. *Neural Computation* **12**:11, 2685-2717. [Abstract] [PDF] [PDF Plus]

308. T. Higuchi, Xin Yao, Yong Liu. 2000. Evolutionary ensembles with negative correlation learning. *IEEE Transactions on Evolutionary Computation* **4**:4, 380-387. [CrossRef]

309. H.-T. Pao, Yeong Yuh Xu, Hung-Yuan Chang, Hsin-Chia Fu. 2000. User adaptive handwriting recognition by self-growing probabilistic decision-based neural networks. *IEEE Transactions on Neural Networks* **11**:6, 1373-1384. [CrossRef]

310. Yasuo Matsuyama. 2000. The ?-EM algorithm and its basic properties. *Systems and Computers in Japan* **31**:11, 12-23. [CrossRef]

311. RAJEEV KUMAR. 2000. ANCHOR — A CONNECTIONIST ARCHITECTURE FOR PARTITIONING FEATURE SPACES AND HIERARCHICAL NESTING OF NEURAL NETS. *International Journal on Artificial Intelligence Tools* **09**:03, 397-416. [CrossRef]

312. A. Yan, X. Chen, R. Zhang, M. Liu, Z. Hu, B. T. Fan. 2000. Predicting the Standard Enthalpy (ΔH o f) and Entropy (S o) of Alkanes by Artificial Neural Networks. *SAR and QSAR in Environmental Research* **11**:3-4, 235-244. [CrossRef]

313. S. Gutta, J.R.J. Huang, P. Jonathon, H. Wechsler. 2000. Mixture of experts for classification of gender, ethnic origin, and pose of human faces. *IEEE Transactions on Neural Networks* **11**:4, 948-960. [CrossRef]

314. Wenxin Jiang. 2000. The VC Dimension for Mixtures of Binary Classifiers. *Neural Computation* **12**:6, 1293-1301. [Abstract] [PDF] [PDF Plus]

315. Naonori Ueda, Ryohei Nakano. 2000. EM algorithm with split and merge operations for mixture models. *Systems and Computers in Japan* **31**:5, 1-11. [CrossRef]

316. Yue Wang, Lan Luo, M.T. Freedman, Sun-Yuan Kung. 2000. Probabilistic principal component subspaces: a hierarchical finite mixture model for data visualization. *IEEE Transactions on Neural Networks* **11**:3, 625-636. [CrossRef]

317. Wenxin Jiang, M.A. Tanner. 2000. On the asymptotic normality of hierarchical mixtures-of-experts for generalized linear models. *IEEE Transactions on Information Theory* **46**:3, 1005-1013. [CrossRef]

318. D Husmeier. 2000. Learning non-stationary conditional probability distributions. *Neural Networks* **13**:3, 287-290. [CrossRef]

319. Bin Zhang, Rao S. Govindaraju. 2000. Prediction of watershed runoff using Bayesian concepts and modular neural networks. *Water Resources Research* **36**:3, 753-762. [CrossRef]

320. Shiro Ikeda. 2000. Acceleration of the EM algorithm. *Systems and Computers in Japan* **31**:2, 10-18. [CrossRef]

321. QingLin Ma, Aixia Yan, Zhide Hu, Zuixong Li, Botao Fan. 2000. Principal component analysis and artificial neural networks applied to the classification of Chinese pottery of neolithic age. *Analytica Chimica Acta* **406**:2, 247-256. [CrossRef]

322. Chapter 8 Design issues — Neural networks **1**, 89-102. [CrossRef]

323. Azriel Rosenfeld, Harry Wechsler. 2000. Pattern recognition: Historical perspective and future directions. *International Journal of Imaging Systems and Technology* **11**:2, 101-116. [CrossRef]

324. A.K. Jain, P.W. Duin, Jianchang Mao. 2000. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**:1, 4-37. [CrossRef]

325. Mohammed Ouali, Ross D. King. 2000. Cascaded multiple classifiers for secondary structure prediction. *Protein Science* **9**:6, 1162-1176. [CrossRef]

326. J. Peres, R. Oliveira, S. Feyo de AzevedoKnowledge based modular networks for process modelling and control **8**, 247-252. [CrossRef]

327. Mike SchusterNeural Nets for Speech Processing . [CrossRef]

328. Yair BartalDivide-and-Conquer Methods . [CrossRef]

329. HSIN-CHIA FU, Y. Y. XU, H. Y. CHANG. 1999. RECOGNITION OF HANDWRITTEN SIMILAR CHINESE CHARACTERS BY SELF-GROWING PROBABILISTIC DECISION-BASED NEURAL NETWORK. *International Journal of Neural Systems* **09**:06, 545-561. [CrossRef]

330. KE CHEN, HUISHENG CHI. 1999. A MODULAR NEURAL NETWORK ARCHITECTURE FOR PATTERN CLASSIFICATION BASED ON DIFFERENT FEATURE SETS. *International Journal of Neural Systems* **09**:06, 563-581. [CrossRef]

331. A. Baraldi, P. Blonda. 1999. A survey of fuzzy clustering algorithms for pattern recognition. I. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **29**:6, 778-785. [CrossRef]

332. L. Hadjiiski, B. Sahiner, Heang-Ping Chan, N. Petrick, M. Helvie. 1999. Classification of malignant and benign masses based on hybrid ART2LDA approach. *IEEE Transactions on Medical Imaging* **18**:12, 1178-1187. [CrossRef]

333. A.N. Srivastava, R. Su, A.S. Weigend. 1999. Data mining for features using scale-sensitive gated experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**:12, 1268-1279. [CrossRef]

334. A. Suarez, J.F. Lutsko. 1999. Globally optimal fuzzy decision trees for classification and regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**:12, 1297-1311. [CrossRef]

335. K Chen. 1999. Improved learning algorithms for mixture of experts in multiclass classification. *Neural Networks* **12**:9, 1229-1252. [CrossRef]

336. W Jiang. 1999. On the identifiability of mixtures-of-experts. *Neural Networks* **12**:9, 1253-1258. [CrossRef]

337. Bao-Liang Lu, H. Kita, Y. Nishikawa. 1999. Inverting feedforward neural networks using linear and nonlinear programming. *IEEE Transactions on Neural Networks* **10**:6, 1271-1290. [CrossRef]

338. Jen-Tzung Chien. 1999. Online hierarchical transformation of hidden Markov models for speech recognition. *IEEE Transactions on Speech and Audio Processing* **7**:6, 656-667. [CrossRef]

339. H Ando. 1999. Unsupervised visual learning of three-dimensional objects using a modular network architecture. *Neural Networks* **12**:7-8, 1037-1051. [CrossRef]

340. J Hansen. 1999. Combining predictors: comparison of five meta machine learning methods. *Information Sciences* **119**:1-2, 91-105. [CrossRef]

341. Sheng Ma, Chuanyi Ji. 1999. Performance and efficiency: recent advances in supervised learning. *Proceedings of the IEEE* **87**:9, 1519-1535. [CrossRef]

342. Sun-Yuan Kung, J. Taur, Shang-Hung Lin. 1999. Synergistic modeling and applications of hierarchical fuzzy neural networks. *Proceedings of the IEEE* **87**:9, 1550-1574. [CrossRef]

343. P. Frasconi, M. Gori, G. Soda. 1999. Data categorization using decision trellises. *IEEE Transactions on Knowledge and Data Engineering* **11**:5, 697-712. [CrossRef]

344. Wenxin Jiang, Martin A. Tanner. 1999. On the Approximation Rate of Hierarchical Mixtures-of-Experts for Generalized Linear Models. *Neural Computation* **11**:5, 1183-1198. [Abstract] [PDF] [PDF Plus]

345. J Zhang. 1999. Inferential estimation of polymer quality using bootstrap aggregated neural networks. *Neural Networks* **12**:6, 927-938. [CrossRef]

346. R Sun. 1999. Multi-agent reinforcement learning: weighting and partitioning. *Neural Networks* **12**:4-5, 727-753. [CrossRef]

347. Ori Rosen, Martin Tanner. 1999. Mixtures of proportional hazards regression models. *Statistics in Medicine* **18**:9, 1119-1131. [CrossRef]

348. 1999. Developing robust non-linear models through bootstrap aggregated neural networks. *Neurocomputing* **25**:1-3, 93-113. [CrossRef]

349. F.M. Candocia, J.C. Principe. 1999. Super-resolution of images based on local correlations. *IEEE Transactions on Neural Networks* **10**:2, 372-380. [CrossRef]

350. V. Ramamurti, J. Ghosh. 1999. Structurally adaptive modular networks for nonstationary environments. *IEEE Transactions on Neural Networks* **10**:1, 152-160. [CrossRef]

351. Robert A. Jacobs. 1999. Computational studies of the development of functionally specialized neural modules. *Trends in Cognitive Sciences* **3**:1, 31-38. [CrossRef]

352. D T Pham, R J Alcock. 1999. Synergistic classification systems for wood defect identification. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering* **213**:2, 127-133. [CrossRef]

353. Yue Wang, Shang-Hung Lin, Huai Li, Sun-Yuan Kung. 1998. Data mapping by probabilistic modular networks and information-theoretic criteria. *IEEE Transactions on Signal Processing* **46**:12, 3378-3397. [CrossRef]

354. Akio Utsugi. 1998. Density Estimation by Mixture Models with Smoothing Priors. *Neural Computation* **10**:8, 2115-2135. [Abstract] [PDF] [PDF Plus]

355. C.L. Fancourt, J.C. Principe. 1998. Competitive principal component analysis for locally stationary time series. *IEEE Transactions on Signal Processing* **46**:11, 3068-3081. [CrossRef]

356. Sin-Horng Chen, Yuan-Fu Liao. 1998. Modular recurrent neural networks for Mandarin syllable recognition. *IEEE Transactions on Neural Networks* **9**:6, 1430-1441. [CrossRef]

357. R. Sun, T. Peterson. 1998. Autonomous learning of sequential tasks: experiments and analyses. *IEEE Transactions on Neural Networks* **9**:6, 1217-1234. [CrossRef]

358. Y. Gotoh, M.M. Hochberg, H.F. Silverman. 1998. Efficient training algorithms for HMMs using incremental estimation. *IEEE Transactions on Speech and Audio Processing* **6**:6, 539-548. [CrossRef]

359. A.D.J. Cross, E.R. Hancock. 1998. Graph matching with a dual-step EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**:11, 1236-1253. [CrossRef]

360. K. Rose. 1998. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE* **86**:11, 2210-2239. [CrossRef]

361. D Wolpert. 1998. Multiple paired forward and inverse models for motor control. *Neural Networks* **11**:7-8, 1317-1329. [CrossRef]

362. Hugh A. Chipman, Edward I. George, Robert E. McCulloch. 1998. Bayesian CART Model Search. *Journal of the American Statistical Association* **93**:443, 935-948. [CrossRef]

363. D Knill. 1998. Ideal observer perturbation analysis reveals human strategies for inferring surface orientation from texture. *Vision Research* **38**:17, 2635-2656. [CrossRef]

364. R. Kumar, P. Rockett. 1998. Multiobjective genetic algorithm partitioning for hierarchical learning of high-dimensional pattern spaces: a learning-follows-decomposition strategy. *IEEE Transactions on Neural Networks* **9**:5, 822-830. [CrossRef]

365. V. Maiorov, R.S. Meir. 1998. Approximation bounds for smooth functions in C(R/sup d/) by neural and mixture networks. *IEEE Transactions on Neural Networks* **9**:5, 969-978. [CrossRef]

366. Aixia Yan, Ruisheng Zhang, Mancang Liu, Zhide Hu, M.A. Hooper, Zhengfeng Zhao. 1998. Large artificial neural networks applied to the prediction of retention indices of acyclic and cyclic alkanes, alkenes, alcohols, esters, ketones and ethers. *Computers & Chemistry* **22**:5, 405-412. [CrossRef]

367. Sheng Ma, Chuanyi Ji. 1998. A unified approach on fast training of feedforward and recurrent networks using EM algorithm. *IEEE Transactions on Signal Processing* **46**:8, 2270-2274. [CrossRef]

368. James A. Reggia, Sharon Goodall, Yuri Shkuro. 1998. Computational Studies of Lateralization of Phoneme Sequence Generation. *Neural Computation* **10**:5, 1277-1297. [Abstract] [PDF] [PDF Plus]

369. Ke Chen. 1998. A connectionist method for pattern classification with diverse features. *Pattern Recognition Letters* **19**:7, 545-558. [CrossRef]

370. Y. Shimshoni, N. Intrator. 1998. Classification of seismic signals by integrating ensembles of neural networks. *IEEE Transactions on Signal Processing* **46**:5, 1194-1201. [CrossRef]

371. A.J. Zeevi, R. Meir, V. Maiorov. 1998. Error bounds for functional approximation and estimation using mixtures of experts. *IEEE Transactions on Information Theory* **44**:3, 1010-1025. [CrossRef]

372. N Ueda. 1998. Deterministic annealing EM algorithm. *Neural Networks* **11**:2, 271-282. [CrossRef]

373. R. Rae, H.J. Ritter. 1998. Recognition of human head orientation based on artificial neural networks. *IEEE Transactions on Neural Networks* **9**:2, 257-265. [CrossRef]

374. J ZHANG, A MORRIS, E MARTIN, C KIPARISSIDES. 1998. Prediction of polymer quality in batch polymerisation reactors using robust neural networks. *Chemical Engineering Journal* **69**:2, 135-143. [CrossRef]

375. C.M. Bishop, M.E. Tipping. 1998. A hierarchical latent variable model for data visualization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**:3, 281-293. [CrossRef]

376. Lloyd P. M. Johnston, Mark A. Kramer. 1998. Estimating state probability distributions from noisy and corrupted data. *AIChE Journal* **44**:3, 591-602. [CrossRef]

377. A. Verikas, K. Malmqvist, L. Bergman, M. Signahl. 1998. Colour classification by neural networks in graphic arts. *Neural Computing & Applications* **7**:1, 52-64. [CrossRef]

378. David J. Miller, Hasan S. Uyar. 1998. Combined Learning and Use for a Mixture Model Equivalent to the RBF Classifier. *Neural Computation* **10**:2, 281-293. [Abstract] [PDF] [PDF Plus]

379. E Bax. 1998. Validation of average error rate over classifiers. *Pattern Recognition Letters* **19**:2, 127-132. [CrossRef]

380. D Husmeier. 1998. Neural Networks for Predicting Conditional Probability Densities: Improved Training Scheme Combining EM and RVFL. *Neural Networks* **11**:1, 89-116. [CrossRef]

381. Jouko Lampinen, Jorma Laaksonen, Erkki OjaPattern recognition **5**, 1-59. [CrossRef]

382. Kishan Mehrotra, Chilukuri K. MohanModular neural networks **3**, 147-181. [CrossRef]

383. H HUAYANG, N MURATA, S AMARI. 1998. Statistical inference: learning in artificial neural networks. *Trends in Cognitive Sciences* **2**:1, 4-10. [CrossRef]

384. Monica Bianchini, Paolo Frasconi, Marco Gori, Marco MagginiOptimal learning in artificial neural networks: A theoretical view **2**, 1-51. [CrossRef]

385. Sheng Ma, Chuanyi Ji. 1998. Fast training of recurrent networks based on the EM algorithm. *IEEE Transactions on Neural Networks* **9**:1, 11-26. [CrossRef]

386. Yoram Singer. 1997. Adaptive Mixtures of Probabilistic Transducers. *Neural Computation* **9**:8, 1711-1733. [Abstract] [PDF] [PDF Plus]

387. Athanasios Kehagias, Vassilios Petridis. 1997. Time-Series Segmentation Using Predictive Modular Neural Networks. *Neural Computation* **9**:8, 1691-1709. [Abstract] [PDF] [PDF Plus]

388. V.P. Kumar, E.S. Manolakos. 1997. Unsupervised statistical neural networks for model-based object recognition. *IEEE Transactions on Signal Processing* **45**:11, 2709-2718. [CrossRef]

389. A.V. Rao, D. Miller, K. Rose, A. Gersho. 1997. Mixture of experts regression modeling by deterministic annealing. *IEEE Transactions on Signal Processing* **45**:11, 2811-2820. [CrossRef]

390. N INTRATOR, S EDELMAN. 1997. Competitive learning in biological and artificial neural computation. *Trends in Cognitive Sciences* **1**:7, 268-272. [CrossRef]

391. Robert A. Jacobs. 1997. Nature, nurture, and the development of functional specializations: A computational approach. *Psychonomic Bulletin & Review* **4**:3, 299-309. [CrossRef]

392. R. Langari, Liang Wang, J. Yen. 1997. Radial basis function networks, regression weights, and the expectation-maximization algorithm. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **27**:5, 613-623. [CrossRef]

393. Daniel M. Wolpert. 1997. Computational approaches to motor control. *Trends in Cognitive Sciences* **1**:6, 209-216. [CrossRef]

394. V. Petridis, A. Kehagias. 1997. Predictive modular fuzzy systems for time-series classification. *IEEE Transactions on Fuzzy Systems* **5**:3, 381-397. [CrossRef]

395. Sherif Hashem. 1997. Optimal Linear Combinations of Neural Networks. *Neural Networks* **10**:4, 599-614. [CrossRef]

396. L Feldkamp. 1997. Adaptive behavior from fixed weight networks. *Information Sciences* **98**:1-4, 217-235. [CrossRef]

397. T. Adali, X. Liu, M.K. Sonmez. 1997. Conditional distribution learning with neural networks and its application to channel equalization. *IEEE Transactions on Signal Processing* **45**:4, 1051-1064. [CrossRef]

398. S Ma. 1997. An Efficient EM-based Training Algorithm for Feedforward Neural Networks. *Neural Networks* **10**:2, 243-256. [CrossRef]

399. R Jacobs. 1997. A Bayesian Approach to Model Selection in Hierarchical Mixtures-of-Experts Architectures. *Neural Networks* **10**:2, 231-241. [CrossRef]

400. Ke Chen, Xiang Yu, Huisheng Chi. 1997. Combining linear discriminant functions with neural networks for supervised learning. *Neural Computing & Applications* **6**:1, 19-41. [CrossRef]

401. Robert A. Jacobs. 1997. Bias/Variance Analyses of Mixtures-of-Experts Architectures. *Neural Computation* **9**:2, 369-383. [Abstract] [PDF] [PDF Plus]

402. A Jain. 1997. Practicing vision: Integration, evaluation and applications. *Pattern Recognition* **30**:2, 183-196. [CrossRef]

403. Assaf J. Zeevi, Ronny Meir. 1997. Density Estimation Through Convex Combinations of Densities: Approximation and Estimation Bounds. *Neural Networks* **10**:1, 99-109. [CrossRef]

404. Gérard Bailly, Rafael Laboissière, Arturo GalvánLearning to speak: Speech production and sensori-motor representations **119**, 593-615. [CrossRef]

405. T Johansen. 1997. Operating regime based process modeling and identification. *Computers & Chemical Engineering* **21**:2, 159-176. [CrossRef]

406. D. Miller, A.V. Rao, K. Rose, A. Gersho. 1996. A global optimization technique for statistical classifier design. *IEEE Transactions on Signal Processing* **44**:12, 3108-3122. [CrossRef]

407. J. T. Connor. 1996. A robust neural network filter for electricity demand prediction. *Journal of Forecasting* **15**:6, 437-458. [CrossRef]

408. Ke Chen, Dahong Xie, Huisheng Chi. 1996. A modified HME architecture for text-dependent speaker identification. *IEEE Transactions on Neural Networks* **7**:5, 1309-1313. [CrossRef]

409. Fengchun Peng, Robert A. Jacobs, Martin A. Tanner. 1996. Bayesian Inference in Mixtures-of-Experts and Hierarchical Mixtures-of-Experts Models with an Application to Speech Recognition. *Journal of the American Statistical Association* **91**:435, 953-960. [CrossRef]

410. M SANCHEZ. 1996. Performance of multi layer feedforward and radial base function neural networks in classification and modelling. *Chemometrics and Intelligent Laboratory Systems* **33**:2, 101-119. [CrossRef]

411. J. Tani. 1996. Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* **26**:3, 421-436. [CrossRef]

412. Steven Gold, Anand Rangarajan, Eric Mjolsness. 1996. Learning with Preknowledge: Clustering with Point and Graph Matching Distance Measures. *Neural Computation* **8**:4, 787-804. [Abstract] [PDF] [PDF Plus]

413. Ming Zhang, J. Fulcher. 1996. Face recognition using artificial neural network group-based adaptive tolerance (GAT) trees. *IEEE Transactions on Neural Networks* **7**:3, 555-567. [CrossRef]

414. E. Alpaydin, M.I. Jordan. 1996. Local linear perceptrons for classification. *IEEE Transactions on Neural Networks* **7**:3, 788-794. [CrossRef]

415. S. Gold, A. Rangarajan. 1996. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**:4, 377-388. [CrossRef]

416. David Miller, Kenneth Rose. 1996. Hierarchical, Unsupervised Learning with Growing via Phase Transitions. *Neural Computation* **8**:2, 425-450. [Abstract] [PDF] [PDF Plus]

417. G. Mato, H. Sompolinsky. 1996. Neural Network Models of Perceptual Learning of Angle Discrimination. *Neural Computation* **8**:2, 270-299. [Abstract] [PDF] [PDF Plus]

418. C HARRIS. 1996. Advances in neurofuzzy algorithms for real-time modelling and control*1. *Engineering Applications of Artificial Intelligence* **9**:1, 1-16. [CrossRef]

419. V. Petridis, A. Kehagias. 1996. Modular neural networks for MAP classification of time series and the partition algorithm. *IEEE Transactions on Neural Networks* **7**:1, 73-86. [CrossRef]

420. Florence d'Alché-Buc, Jean-Pierre Nadal. 1995. Asymptotic performances of a constructive algorithm. *Neural Processing Letters* **2**:2, 1-4. [CrossRef]

421. Shun-ichi Amari. 1995. The EM Algorithm and Information Geometry in Neural Network Learning. *Neural Computation* **7**:1, 13-18. [Abstract] [PDF] [PDF Plus]

422. M Jordan. 1995. Convergence results for the EM approach to mixtures of experts architectures. *Neural Networks* **8**:9, 1409-1431. [CrossRef]

423. S Amari. 1995. Information geometry of the EM and em algorithms for neural networks. *Neural Networks* **8**:9, 1379-1408. [CrossRef]