A Deterministic Annealing Approach for Parsimonious Design of Piecewise Regression Models

Ajit V. Rao, *Student Member, IEEE*, David J. Miller, *Member, IEEE*, Kenneth Rose, *Member, IEEE*, and Allen Gersho, *Fellow, IEEE*

Abstract—A new learning algorithm is proposed for piecewise regression modeling. It employs the technique of deterministic annealing to design space partition regression functions. While the performance of traditional space partition regression functions such as CART and MARS is limited by a simple tree-structured partition and by a hierarchical approach for design, the deterministic annealing algorithm enables the joint optimization of a more powerful piecewise structure based on a Voronoi partition. The new method is demonstrated to achieve consistent performance improvements over regular CART as well as over its extension to allow arbitrary hyperplane boundaries. Comparison tests, on several benchmark data sets from the regression literature, are provided.

Index Terms—Statistical regression, piecewise regression, deterministic annealing, parsimonious modeling, generalization, nearest-prototype models.

1 Introduction

The problem of statistical regression is to approximate an unknown function from the observation of a limited sequence of (typically) noise-corrupted input-output data pairs. Regression is an important tool in diverse areas, including statistics, computer science and applied mathematics, various engineering disciplines, business administration, and the social sciences.

The regression problem is usually stated as the optimization of a cost that measures how well the regression function $g(\mathbf{x})$ approximates the output $\mathbf{y}$, over a set $\{(\mathbf{x}, \mathbf{y})\}$. Here, $\mathbf{x} \in \mathcal{R}^m$, $\mathbf{y} \in \mathcal{R}^n$, and $g : \mathcal{R}^m \to \mathcal{R}^n$. Perhaps the most commonly used objective is the least squares cost

$$D = \sum_{(\mathbf{x},\mathbf{y})} \|\mathbf{y} - g(\mathbf{x})\|^2 . \qquad (1)$$

The regression function $g(\cdot)$ is *learned* by minimizing the design cost, D, measured over a *training set*, $\mathcal{T} = \{\mathbf{x}_i, \mathbf{y}_i\}$, but with the ultimate performance evaluation based on the generalization cost, which is the error D measured over a *test set*. The mismatch between the design cost and the generalization cost is a fundamental difficulty which is the subject of much current research in statistics in general and in neural networks in particular. It is well-known that for most choices of D, the cost measured during design monotonically decreases as the size of the learned regression model is increased, while the generalization cost will start to increase when the model size grows beyond a certain point. In general, the optimal model size, or even a favorable regime of model sizes, is unknown prior to training the model. Thus, the search for the correct model size must naturally be undertaken as an integral part of the training. Many techniques for improving generalization in learning are inspired by the well-known principle of Occam's razor,[1] which essentially states that the simplest model that accurately represents the data is most desirable. From the perspective of the learning problem, this principle suggests that the design should take into account some measure of the simplicity, or parsimony, of the solution, in addition to performance on the training set. In one basic approach, penalty terms are added to the training cost, either to directly favor the formation of a small model [1], [25], or to do so indirectly via regularization/smoothness constraints or other costs which measure overspecialization. A second common approach is to build a large model, overspecialized to the training set, and then attempt to "undo" some of the training by retaining only the vital model structure, removing extraneous parameters that have only learned the nuances of a particular noisy training set. This latter approach is adopted in the pruning methods for regression trees [4] and in methods such as optimal brain surgeon [14] in the context of neural networks.

While these techniques provide a way of generating parsimonious models, there is an additional serious difficulty that most methods do not address directly, which can also severely limit the generalization achieved by learning. This difficulty is the problem of nonglobal optima of the cost surface, which can easily trap descent-based learning

- *A.V. Rao is with SignalCom Inc., 7127 Hollister Ave., Suite 109, Goleta, CA 93117. E-mail: ajit@dsp-signal.com.*
- *D.J. Miller is with Pennsylvania State University, University Park, PA 16802. E-mail: miller@pippin.ee.psu.edu.*
- *K. Rose and A. Gersho are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106. E-mail: {rose; gersho}@ece.ucsb.edu.*

Manuscript received 23 Sept. 1996; revised 6 Nov. 1998. Recommended for acceptance by A. Webb.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 108365.

[1] William of Occam (1285-1349): "Causes should not be multiplied beyond necessity."

methods. If the designed regression function performs poorly as a result of a shallow, local minimum trap, the typical recourse is to optimize a larger model, under the assumption that the model was not sufficiently powerful to characterize the data well. The larger model will likely improve the design cost but may result in overspecialization and hence suboptimal performance outside the training set. Clearly, a superior optimization method that finds better models of smaller size will enhance the generalization performance of the regression function. While conventional techniques for parsimonious modeling control the model size, they do not address this optimization difficulty. In particular, methods such as CART [4] for tree-structured regression employ greedy heuristics in the "growing" phase of the model design which might lead to poorly designed trees. The subsequent pruning phase is then restricted, in its search for parsimonious models, to choosing pruned subtrees of this initial, potentially suboptimal tree. Techniques which add penalty terms to the cost can also suffer from problems of local minima. In fact, in many cases the addition of a penalty term can actually increase the complexity of the cost surface and exacerbate the local minimum problem (e.g., [35]).

As an alternative approach, in this work we present an optimization technique for regression modeling which, through its formulation of the problem, simultaneously embeds the search for a parsimonious solution and for one that is optimal in the design cost. The method is an extension of the deterministic annealing (DA) algorithm, proposed originally for data clustering and vector quantization [28], [29]. In that work, it was shown that embedding the clustering problem within a statistical physics framework yields an annealing-based optimization method which avoids shallow local optima of the cost. An interesting property of the method is that the model size *grows* by bifurcations which occur at distinct temperatures during the optimization process. The DA clustering method was later extended to include additional constraints and costs [27], [5]. In recent work [21], the approach was given an important extension that allows the incorporation of structural constraints on the assignments of data, thus extending the applicability of the method to a larger variety of practical problems. This approach has since been applied to the design of statistical classifiers [21], generalized vector quantizers for compression applications [23], and mixture of experts models for regression [24]. The present work extends the DA approach to address the problem of designing piece-wise regression models. As in the original DA method, the growth in model size is achieved by bifurcations in the annealing process, which occur so as to directly minimize the physical quantity known as the free energy. Thus, the annealing method naturally facilitates the generation of a sequence of candidate parsimonious models of increasing size as the temperature is lowered to zero, while avoiding many local optima of the cost along the way. The sequence of candidate models thus obtained can then be evaluated on an independent validation set to select the model size. While DA is applicable to the design of a variety of piecewise regression structures, this paper will focus on a particular piecewise model which we will refer to as the Nearest Prototype (NP) model. The NP structure provides a good trade-off between simple, tree-structured models like standard CART and more powerful, yet less informative[2] neural network models such as multilayer perceptrons. The NP structure is a piecewise regression model like CART. However, whereas CART is typically restricted to forming nested partitions with, even more restrictively, splits parallel to coordinate axes, the NP model divides the input feature space by forming a Voronoi partition. While this more general model engenders a substantially challenging optimization problem, deterministic annealing facilitates the solution and exploits the model's potential for enhanced regression performance.

To summarize the contents of the rest of the paper: In the next section, we present an overview of the existing methodology in statistical regression and indicate the relevant shortcomings due to both structures and learning techniques. We describe the NP regression structure in Section 3 and derive the deterministic annealing method for its optimization. In Section 4, we compare the performance of DA with that of CART on both synthetic and real-world datasets. The results substantiate the superior performance of DA.

2 The Regression Problem

2.1 Conventional Approaches

The basic approach to regression is the technique of local averaging. In simple averaging methods, the output estimate $\hat{\mathbf{y}}$ for an input $\mathbf{x}$ is computed as a weighted average over training points $(\mathbf{x}_i, \mathbf{y}_i)$ whose input coordinates are "close" to $\mathbf{x}$ in the sense of a well-defined distance or dissimilarity measure. Although this basic method has excellent asymptotic properties [9], its practical usefulness is limited since one must have access to the entire training set to compute the regression estimate. Moreover, the method suffers from the notorious "curse of dimensionality" (COD), as its performance deteriorates rapidly with increased dimensionality of the feature space. One way to mitigate the damage caused by COD is to perform regression on a lower-dimensional projection subspace. This technique is adopted for example in the projection pursuit regression (PPR) [11], and alternating conditional expectations (ACE) [3] approaches.

Here, we adopt an alternative "space partitioning" approach to solve the COD problem. This approach is derived from the observation that, despite the large "volume" of the feature space, it is often the case that the data is localized to a few relatively dense "clusters." Simple local averaging techniques fail for such data sets, because the weighting functions used for averaging do not take into account the variation in the density of the training set population. To exploit this variation, one must adapt the size and shape of

[2] By an "informative" solution, what is meant is that the solution can be readily interpreted to determine, e.g., which features play a significant role. CART solutions directly provide this information, since "splits" into more regions occur along feature axes. More distributed neural network models are less informative in this sense.

the averaging regions to the local nature of the data as in variable kernel methods [31], [30]. A natural extension of this idea is to divide the input space into regions of different sizes and shapes and to use a suitable "local" regression model in each region. The space partitioning approach also eliminates the need to access the entire training set to compute a regression estimate, thus reducing the complexity of implementation. However, the input partition and the local models must be designed carefully in order to obtain a good regression function. As a simple example, consider the case where $\mathbf{X}$ is one-dimensional (scalar). We partition the axis into continuous sub-intervals, $\{R_k\}$, each associated with a simple linear model for the output $\mathbf{Y}$: $f(\mathbf{x}, \Lambda_k) = \lambda_{k1}\mathbf{x} + \lambda_{k0}$. To achieve good performance for this model, the optimal locations of the "knots" (points that separate successive intervals) must be found. For this one-dimensional problem, the optimal algorithm is well-known and is based on dynamic programming [2]. For higher dimensions, however, only suboptimal practical methods are known.

Some important examples of space-partitioning regression include classification and regression trees (CART) [4] and multiple adaptive regression splines (MARS) [10]. CART divides the feature space into a sequence of nested regions and uses simple local averaging models in each of the regions. MARS is similar to CART but with local averaging based on splines, which makes it more complex and more versatile. The use of splines allows smoothness of regression functions across region boundaries. An attractive feature of CART is the simplicity of the design algorithm, which can be used to build a sequence of regression functions of progressively increasing model order. With each step of the design, the model order is increased leading to a decrease in the approximation error over the training set. However, good generalization is only achieved for a limited regime of model sizes beyond which the generalization performance deteriorates. One straightforward way to select the model size, is to design CART functions of different model orders based on the training set, and then choose from this set the function which performs best on an independent "validation set." However, this simple scheme is severely limited in its candidate set of parsimonious models. A potentially more powerful method involves pruning back an initially grown tree, effectively searching over the entire (and very large) set of pruned subtrees for models that give the best performance over the validation set [4]. Indeed, pruning has the potential of producing better sub-trees than those that naturally appear in the growing phase. It is important to note that although the sequence of candidate pruned trees is quite large, this set is entirely determined and limited by the initially grown tree. In fact, in our experimental results of Section 4, we have observed that in many cases the sequence of trees produced by pruning is *exactly the same* sequence of the growing phase (but in reverse order.) Thus, in such cases, pruning does not provide an additional benefit and the quality of the ultimate solution rests solely on the growing phase of design and its sequence of trees.

An important drawback of both CART and MARS is that the shapes of the regions over which local averaging is performed are highly restrictive. In most CART and MARS implementations, the regions are constrained to be hyper-rectangles, with sides parallel to the coordinate axes. Recently, some extensions to the original methods have been proposed which allow for regions whose boundary orientation is not as limited [7], [33], [15], [19], [34].

Another serious drawback is the greedy nature of standard tree growing approaches. In the basic design approach, the partitions of the input space are designed in a hierarchical manner with the partition with N regions formed by subdividing one of the regions in the partition with $N - 1$ regions. However, the upper levels in the hierarchy cannot be re-optimized as more regions are introduced (awareness of a similar difficulty in the projection pursuit model [11] led to the backfitting mechanism). The following example illustrates the problem of suboptimality in hierarchical partitioning.

Consider a training set of 200 ($\mathbf{x}$, $\mathbf{y}$) pairs ($\mathbf{x}$ and $\mathbf{y}$ each one-dimensional) generated from three Gaussian clusters. Fig. 1a shows a regression function designed by CART for this data. The regression function is the solid line that is superimposed over a scatter plot of the training set. To obtain this function, in the first step, a boundary is introduced at $\mathbf{x} = 0.46$ (the vertical line that breaks one of the clusters). Next, a second boundary is introduced at $\mathbf{x} = 0.63$ to obtain the function in the figure. Note that the CART algorithm cannot re-optimize the boundary introduced in the first step. The process of recursively dividing the regions is continued until a full tree is grown. Subsequently, the large tree is pruned back. However, pruning offers no means to adjust the suboptimal boundary introduced in the first step. Contrast the CART solution (average squared approximation error = 0.068) with the partition into three regions obtained by the DA algorithm (to be described later in this paper) in Fig. 1b. The DA partition identifies the three clusters correctly and achieves a lower average squared approximation error (0.052). It should be emphasized here that this example only serves to illustrate the difficulties that arise due to the suboptimality of the high level splits of CART. Neither Fig. 1a nor Fig. 1b represents the optimal regression model obtained by the corresponding design method. In fact, a complete CART design performed by growing a full tree, pruning it to the root node and evaluating each pruned tree on an independent validation set, produced a solution with model order 20 and an average squared regression error of 0.036 on a large independent test set. In contrast, the DA design method produced a model with five regions which yielded an average squared regression error of 0.032 on the independent test set.

Motivated by the drawbacks of traditional regression methods, we next propose a novel regression strategy. The structure we use is more powerful than that of CART and MARS because of its ability to achieve complicated partitions of the input space with fewer parameters and at low computational complexity. More importantly, we propose a deterministic annealing approach, based on principles of information theory, for designing the regression function.

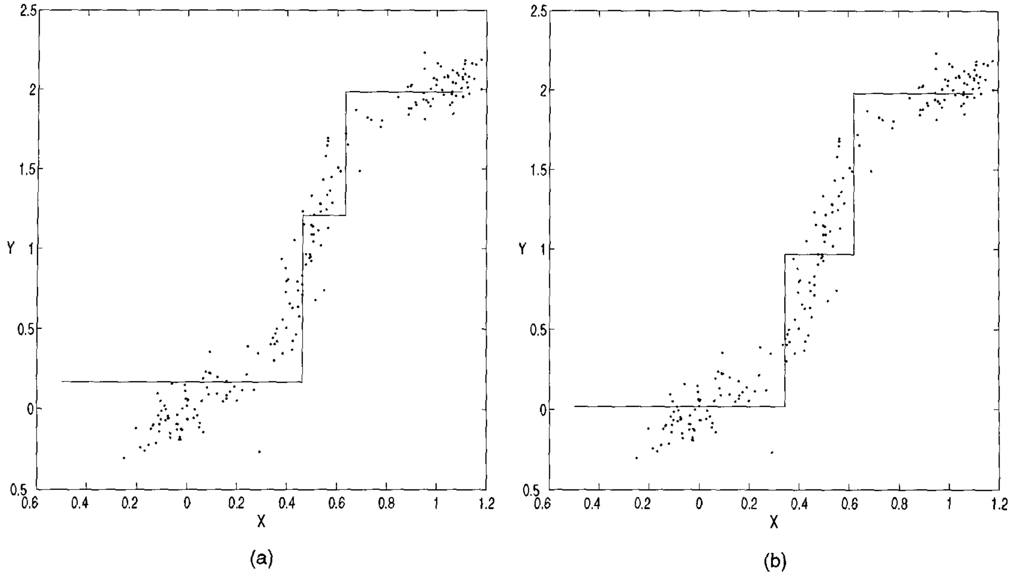


Fig. 1. An example demonstrating the suboptimality of the recursive tree growth approach of CART. (a) Regression function with three regions designed by CART (MSE = 0.068). (b) Function with three regions obtained by using DA (MSE = 0.052). Each function is superimposed over the scatter plot of the training data.

3 The Deterministic Annealing Approach for Regression

In this section, we attack the problem of designing space partition regression functions. We first consider "hard," deterministic space partitioning wherein each data point is uniquely assigned to a partition cell of the feature space. We then extend this notion to *randomized* space partitioning, wherein points are probabilistically associated with partition cells. While the paradigm of randomized space partitioning is the basis for several regression models such as normalized radial basis functions and hierarchical mixture of experts, in the present context randomized partitioning is used only as a tool for the design optimization, with the ultimate (designed) model structure a regular (hard) space-partitioned regression function.

Our design method builds on the deterministic annealing (DA) approach to clustering and related problems [28], [29], [27], its extension to introduce structural constraints which is derived in detail for the problem of statistical classification in [21], and its extension for generalized vector quantization and estimation [23]. The DA approach that we derive here is based on randomized space partitioning. The degree of randomization of the partition is measured by the Shannon entropy whose level is controlled during the design process. This framework serves two objectives: First, the cost which was a piecewise constant function of the partition parameters, becomes continuously differentiable, thus enabling a straightforward optimization using local gradients. Second, it provides a mechanism for an annealing method which has good potential for avoiding poor local optima. At the limit of low entropy, the randomized space partition that we design reduces to a structured, non-random partition. This method is applicable to virtually all space partitioning structures. However, in this paper we chose to employ the DA strategy to design the nearest prototype partition structure and exploit its potential.

3.1 Space Partitioning

In the basic space partitioning approach, the input (or feature) space, $\mathcal{R}^m$, is partitioned into K regions or cells, R_k. The regression function is given by:

$$g(\mathbf{x}) = f(\mathbf{x}, \Lambda_k), \quad \forall \mathbf{x} \in R_k, \quad k = 1, \ldots, K. \tag{2}$$

Typically, the local parametric model, $f(\mathbf{x}, \Lambda_k)$ has a simple form, e.g., linear or Gaussian, and is determined by the parameter set, Λ_k. The average approximation error measured over the training set is then

$$D = \frac{1}{N} \sum_{j=1}^{K} \sum_{i:\mathbf{x}_i \in R_j} d\big(\mathbf{y}_i, f(\mathbf{x}_i, \Lambda_j)\big) \tag{3}$$

where $d(\cdot,\cdot)$ is a distortion measure. A common choice is the squared difference error. For the moment, let us suppose that the local model parameters $\{\Lambda_k\}$ are fixed. Then, the remaining optimization problem is to choose, among all possible partitions, the one that leads to the best fit between the data and the fixed region models. (Of course, in reality the space partition and local model design problems are not separable, but this temporary assumption is useful to motivate our design approach.) Rather than seek the optimal hard partition directly, we will find it useful to generalize the domain of the search space by allowing *randomized* assignment of input samples. In this extended paradigm, input samples may be associated with each cell of the partition *in probability*. The randomized partition is specified by the distributions denoted $\{P(\mathbf{x} \in R_j)\}$.[3] Now, the approxi-

[3] Note that there is a distribution per domain point, x. $P(\mathbf{x} \in R_j)$ is the probability of the event that input x is assigned to cell R_j. Hence, $\sum_j P(\mathbf{x} \in R_j) = 1 \ \forall \mathbf{x}$.

mation error (3) over the training set is generalized to the *expected* approximation error:

$$D = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} P\left(\mathbf{x}_i \in R_j\right) d\left(\mathbf{y}_i, f\left(\mathbf{x}_i, \Lambda_j\right)\right). \tag{4}$$

Note that the standard ("hard") space partitioning cost (3) is the special case where $P\left(\mathbf{x} \in R_j\right) \in \{0, 1\}, \forall \mathbf{x}$.

So far we have let $\{R_j\}$ be any partition of the input space. While this strategy may appear to be useful for obtaining good regression functions, minimizing D without any constraint on the complexity of the partition often results in a function that is too complex for practical implementation and which generalizes poorly outside the training set. It is necessary, therefore, to impose constraints on the structure of the partition $\{R_j\}$. Depending on the imposed structural constraint, different classes of regression functions can be obtained. A common approach to imposing structural constraints on the partition is to require that it be consistent with that of a standard, parametrized classifier such as a nearest prototype (NP), decision tree, radial basis function or multilayer perception classifier. We have adopted this approach and found the nearest prototype (NP) structure to be useful in this context. It should however be reemphasized that although in this paper we elaborate the method only for the NP structure, it is applicable to other structured partitions.

3.2 The Nearest Prototype Partition

An NP partition is represented by a set of *prototype vectors*, $\{\mathbf{s}_j : j = 1, 2, 3, \ .. \ K\}$ in the feature (input) space. An input $\mathbf{x}$ belongs to region R_k if $\mathbf{s}_k$ is the prototype nearest to it: $\|\mathbf{x} - \mathbf{s}_k\| \leq \|\mathbf{x} - \mathbf{s}_j\|, \forall j$. We have chosen the convenient Euclidean distance as the proximity measure. Note that the prototypes completely specify the partition $\{R_j\}$. Given the set of prototypes, we may equivalently define the NP partition as the partition which minimizes the criterion

$$C = \frac{1}{N}\sum_{j=1}^{K}\sum_{i:\mathbf{x}_i \in R_j}\left\|\mathbf{x}_i - \mathbf{s}_j\right\|^2. \tag{5}$$

Such a space partition consists of K convex regions and is known as a Voronoi partition. The NP partition is a generalization of the one-dimensional piecewise model to higher dimensions. It is also referred to as the optimal vector quantizer or encoder partition in source coding [12]. In the pattern classification community, this structure is commonly associated with the learning vector quantizer [17] and the k-means algorithm [18].

In the context of regression, imposing the NP structure on the space partition results in a simple regression function that is robust to noise in the training set. However significant difficulties arise in the design optimization. While the one-dimensional piecewise model can be designed based on a dynamic programming approach [2], this technique cannot be extended to higher dimensions. Moreover, there are no known descent methods for direct optimization of the model parameters so as to minimize the cost, D. A simple explanation for this difficulty is that the variables $\{\mathbf{s}_j\}$ do not appear explicitly in the cost function (3), but do so indirectly through $\{R_j\}$. An infinitesimal change in the location of the prototypes does not change the region assignment of any vector in the training set (unless a training vector falls exactly on a region boundary—a set of zero measure if $\mathbf{X}$ is absolutely continuous). Consequently, the gradients of the regression cost D with respect to the prototype locations $\mathbf{s}_j$ are zero almost everywhere. In other words, almost every possible set of prototype locations appears as a local minimum to gradient based methods. One way to circumvent this difficulty is to optimize the partition by searching over the space of *randomized*, or *soft*, partitions, in such a way that ultimately enforces a hard partition at some limit. We next propose how to realize this approach.

Recall that the hard NP partition minimizes an objective function (5). In a similar fashion, we can characterize a randomized partition by the expected cost,

$$C = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} P\left(\mathbf{x}_i \in R_j\right)\left\|\mathbf{x}_i - \mathbf{s}_j\right\|^2. \tag{6}$$

Of course, minimization of C in (6) over the distributions $\left\{P\left(\mathbf{x}_i \in R_j\right)\right\}$ yields the "hard" NP partition, where each $\mathbf{x}_i$ is fully associated with the nearest prototype. While this is desirable eventually, during the design we wish to control the "randomness" in the assignments. Towards this goal, we introduce the *Shannon entropy*—an information theoretic measure of randomness. The entropy of a partition is given by

$$H = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j} P\left(\mathbf{x}_i \in R_j\right)\log P\left(\mathbf{x}_i \in R_j\right). \tag{7}$$

which implicitly assumes that the samples are independently assigned to the partition regions.

Given a fixed set of prototypes and models and a prespecified constraint on the randomness H_0, we obtain the probabilistic NP partition by minimizing C of (6) subject to $H = H_0$. Essentially, this constrained optimization problem seeks the partition which is nearest to the NP structure *in the sense of minimum* C while maintaining a specified level of randomness, as measured by the Shannon entropy. This constrained optimization problem may be rewritten as the minimization of the corresponding Lagrangian. The solution yields the Gibbs distribution as the optimal association probabilities:

$$P\left(\mathbf{x} \in R_j\right) = \frac{e^{-\gamma\left\|\mathbf{x}-s_j\right\|^2}}{\sum_{k} e^{-\gamma\left\|\mathbf{x}-s_k\right\|^2}}. \tag{8}$$

Here, γ is a nonnegative Lagrange multiplier which controls the "randomness" of the space partition. Note that at the limit $\gamma \to \infty$, the probabilistic NP partition reduces to a hard NP partition as required.[4] A layer of such neu-

[4] It is interesting to note at this stage that the functional form of (8) is identical to that of the neuron in a normalized radial basis function network [22].

rons may be viewed as generating the maximum entropy distribution for the association of the input to the "centers" of a layer at a prespecified variance level. Our entropy constrained formulation has thus yielded a parametric form for the randomized association of inputs to the partition regions. In (8), the degree of association with a point depends on the proximity of the region's prototype. The "peakiness" of the distributions depends on the scale parameter, γ, which we use to control the entropy of the distribution.

The result (8) is a structural form for randomized space partitioning which allows direct control of the randomness or "softness" of the partition, and which provides a useful probabilistic framework within which to search for an optimal hard partition regression function. The overall optimization approach based on this framework can now be described. Note first that since the expected approximation error of (4) depends in a continuous manner on the prototypes $\{\mathbf{s}_j\}$ via the association probabilities, we could simply define a gradient descent technique on this cost surface, which eliminates the original difficulty due to zero gradients. However the optimization must be consistent with the entropy constraint that was imposed on the probabilistic partition. Otherwise, it would encourage the parameters to settle down to a very poor local minimum, via a quick "hardening" of the associations. Alternatively, we suggest to minimize the expected cost (4) while simultaneously constraining the entropy of the associations in order to deter a quick hardening and resulting suboptimality of the solution. More specifically, we minimize the regression cost, D of (4) subject to an entropy constraint, $H = H_0$, where $P(\mathbf{x} \in R_j)$ takes the functional form of (8). Equivalently, we minimize the unconstrained Lagrangian,

$$\min_{\{\mathbf{s}_j\},\{\Lambda_j\},\gamma} L = D - TH \tag{9}$$

where the Lagrange parameter, T is referred to as the "temperature" to emphasize an intuitively compelling analogy to statistical physics.

3.3 Analogy to Statistical Physics—Helmholtz Free Energy

After close inspection of (9), one may recognize one of the cornerstone equations of thermodynamics. Let the regression cost D be the thermodynamic energy of a physical system, and recall that T is the temperature and H is the entropy. This implies that the Lagrangian

$$L = D - TH \tag{10}$$

is the Helmholtz free energy of the system. The temperature (Lagrange multiplier) determines a balance of energy (cost) and entropy (randomness). By minimizing the Lagrangian L we minimize the Helmholtz free energy and, in effect, seek isothermal equilibrium at the given temperature T. Of particular importance is the case of $T \to 0$ which corresponds to direct minimization of D, our ultimate objective. This suggests the possibility of implementing an annealing process, where the temperature is gradually lowered while maintaining the system at thermal equilibrium. Such a process allows one to avoid many of the local minima of the energy D. This interpretation of the Lagrangian in terms of the free energy in statistical physics has been offered in the context of our previous work [28], [21], [23]. The analogy of annealing to physical systems will provide more insight later in this section when we discuss *phase transitions* in the process.

3.4 Deterministic Annealing

Since careful annealing of physical systems ensures convergence to the ground state (global minimum of the energy), it gives motivation to develop a similar approach in the context of regression. Starting from a high value of T (high entropy of the random partition), we gradually reduce T while optimizing L at each step (maintaining the system at equilibrium). In the limit of $T = 0$, we are merely minimizing the regression cost D. It is, however, important to note the distinction between our approach and the method of simulated annealing (SA). SA is stochastic in nature, explicitly simulating the random evolution of the system to achieve stochastic equilibrium as the steady-state distribution over the states of a Markov chain. Our approach, on the other hand is a *deterministic* annealing (DA) approach, which replaces explicit stochastic simulations by expectations. Note that L, D, and H are all defined by expectation. Thus we replace the computationally intensive stochastic simulations by straightforward deterministic optimization of the Helmholtz free energy L. The resulting algorithm is considerably faster than comparable stochastic approaches. However, although the DA method has significant ability to avoid many local optima that trap descent methods, it is not guaranteed to find the global optimum.

We initialize the algorithm at $T \to \infty$ (in practice, T is simply chosen large enough; we will specify how large it should be when we discuss phase transitions). It is clear from (9) that the goal at this temperature is to maximize the entropy of associating inputs with regions. The solution is achieved by allowing all the prototypes to be located at the global centroid of the data. The distributions, $\{P(\mathbf{x} \in R_j)\}$, are consequently uniform. The same parameters, Λ_j, are used for the local regression models in all the regions—effectively, we have a single, global regression model. As the temperature is gradually lowered, in steps, optimization is carried out at each temperature to find the prototype locations $\{\mathbf{s}_j\}$, local model parameters, $\{\Lambda_j\}$, and scale parameter, γ, that minimize the Lagrangian, L. As $T \to 0$, the Lagrangian reduces to the regression cost, D. Further, since we have forced the entropy to go to zero, the randomized space partition that we obtain becomes a hard NP partition. In practice, we anneal the system to a low temperature, where the entropy of the random partition is very small ($H < H_f$). Further annealing will not move the prototypes significantly. Hence we fix the location of the prototypes at this point and jump to $T = 0$ (quench) to perform a "zero entropy iteration," where we partition the training set according to the "hard" nearest prototype rule and optimize the parameters of the local regression models $\{\Lambda_j\}$ to

TABLE 1
PSEUDO-CODE SKETCH OF THE DA IMPLEMENTATION USED IN THE SIMULATIONS

```
1) Initialize:
       $K = 2;\quad T = 1.1T_c;\quad \gamma = \gamma_{in};\quad \mathbf{s}_1 = \mathbf{s}_2 = \mu_{\mathbf{x}};$
2) Perturb:
       a)  $\mathbf{s}_j \leftarrow \mathbf{s}_j + v \ \forall j;\quad v$ =Gaussian perturbation with mean= 0, variance= $\epsilon^2$.
       b)  Initialize: $L_{old} = D - TH$
3) Thermal equilibrium:
       a)  $\mathbf{s}_j \leftarrow \mathbf{s}_j - \alpha\frac{\partial L}{\partial \mathbf{s}_j} \ \forall j$  ($\alpha$ selected by line search)
       b)  $\gamma \leftarrow \gamma - \beta\frac{\partial L}{\partial \gamma}$  ($\beta$ selected by line search)
       c)  $\Lambda \leftarrow \frac{\sum_i P(\mathbf{x}_i \in R_j)\mathbf{y}_i}{\sum_i P(\mathbf{x}_i \in R_j)} \ \forall j$  (Closed form solution for $\Lambda_j$ for the sqaured-error case)
       d)  Equilibrium Check: $L = D - TH$;  If $(\frac{L_{old}-L}{L_{old}} > L_{th})$ $\{L_{old} = L;$ Goto step 3(a)$\}$
4) Model Size determination:
       a)  $\{$ If $(|\mathbf{s}_j - \mathbf{s}_k| < \epsilon)$  Replace $\mathbf{s}_j$,$\mathbf{s}_k$ by a single prototype at $\frac{\mathbf{s}_j+\mathbf{s}_k}{2}\} \ \forall j, k$
       b)  $K$ = Number of prototypes after merging.
4) Freeze* and calculate the prediction error on the validation set.
5) If $(H < H_f)$ Stop.
6) Cooling: $T \leftarrow 0.95T$
7) Duplication: Replace each prototype by two prototypes at the same location. $K \leftarrow 2K$
8) Goto Step 2

   *Freezing
   a) Set $T = 0$
   b) $\gamma \leftarrow 1.1\gamma$
   c) Perform steps 3a,3c.
   d) If $(H > H_f)$ Goto Step b; Else Stop
```

The local models are constant (Λ_j), and a squared-error cost function is used.

minimize the regression cost, D. This approach is consistent with our ultimate goal of optimizing the regression cost constrained on using a (hard) structured space partition.

A brief sketch of the DA algorithm is as follows:

1) Initialize: $T = T_i$.
2) $\min_{\{\mathbf{s}_j\},\{\Lambda_j\},\gamma} L = D - TH$.
3) lower temperature: $T \leftarrow q(T)$.
4) If $H \geq H_f$ goto step 2.
5) Zero entropy iteration: Partition using Hard nearest prototype rule, $\min_{\{\Lambda_j\}} D$.

In our simulations, we used an exponential schedule for reducing T, i.e., $q(T) = \alpha T$, where $\alpha < 1$, but other annealing schedules are possible. The parameter optimization of step 3 may be performed by any local optimization method. In our simulations, this minimization is based on a gradient descent approach. A pseudo-code sketch of the steps used in our implementation is given in Table 1.

For the nearest prototype regression function, the gradients may be expressed as

$$\frac{\partial L}{\partial \mathbf{s}_j} = \frac{2\gamma}{N}\sum_{i=1}^{N} P\left(\mathbf{x}_i \in R_j\right)\left(L_{ij} - < L_{ik} >\right)\left(\mathbf{x}_i - \mathbf{s}_j\right), \tag{11}$$

$$\frac{\partial L}{\partial \Lambda_j} = \frac{1}{N}\sum_{i=1}^{N} P\left(\mathbf{x}_i \in R_j\right)\frac{\partial d\left(\mathbf{y}_i, f\left(\mathbf{x}_i, \Lambda_j\right)\right)}{\partial \Lambda_j}, \tag{12}$$

and

$$\frac{\partial L}{\partial \gamma} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j} P\left(\mathbf{x}_i \in R_j\right)\left(L_{ij} - < L_{ik} >\right)\left\|\mathbf{x}_i - \mathbf{s}_j\right\|^2. \tag{13}$$

Here, $L_{ij} = d(\mathbf{y}_i, f(\mathbf{x}_i, \Lambda_j)) - T\|\mathbf{x}_i - \mathbf{s}_j\|^2$ is the Lagrangian cost term associated with the ith training vector and region R_j, and $<L_{ik}> = \sum_k L_{ik} P(\mathbf{x}_i \in R_k)$ is the expected Lagrangian cost of the ith training vector averaged over its partition region assignments.

The prototype condition can be given an intuitive interpretation as a type of supervised learning rule. It suggests that moving a prototype vector $\mathbf{s}_j$ in the negative gradient direction means pushing the vector away from (towards) points that it "owns" probabilistically through $P(\mathbf{x}_i \in R_j)$ and for which the cost L_{ij} incurred by associating with region R_j is greater than (less than) the average cost, $<L_{ik}>$. The optimality condition for the scale parameter γ leads to a similar interpretation. Essentially, γ is either increased to solidify ownership of a point by a region if the cost is smaller than the average cost or is decreased to weaken ownership of a point by a region if the cost is larger than the average. Whereas supervised learning methods typically involve making hard classification decisions, the negative entropy component of the cost guarantees that the distribution $\{P[\cdot]\}$ remains "soft" for finite T.

3.5 Phase Transitions

An interesting phenomenon of the annealing method is the existence of discrete bifurcations, analogous to phase transitions in a physical system. Let us define the *effective model size* as the number of *distinct* prototypes. When $T \to \infty$, the solution dictates that all the prototypes are coincident. Correspondingly, the distributions $\{P(\mathbf{x} \in R_j)\}$ are uniform and the model parameters $\{\Lambda_j\}$ for all regions are identical. The effective model size is thus one. As T is gradually reduced, the model size grows through discrete bifurcations, i.e., there are distinct (critical) temperatures where the prototypes split into sub-groups and the effective size of the model increases. This is a useful phenomenon from the viewpoint of parsimony. Since model size is a good measure of parsimony, phase transitions correspond to discrete steps in the algorithm where parsimony is exchanged for lower regression cost. The result is a discrete sequence of regression functions which are progressively less parsimonious but perform better on the training set. Each function in this sequence is a potential candidate model for the data. We measure the average approximation error of each function in this sequence over an independent validation set and determine the ultimate regression function as the candidate model that gives the best performance on the validation set.

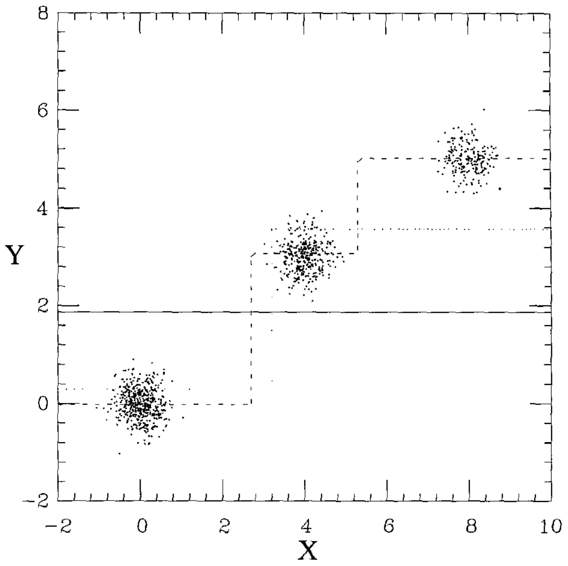


Fig. 2. Phase transitions in the annealing process. Solutions at effective model sizes 1 (solid line), 2 (dotted line), and 3 (dashed line).

Fig. 2 demonstrates a sequence of solutions for a simple regression example. Here **X** and **Y** are both scalars and the local regression models $\{f(\mathbf{x}, \Lambda_j)\}$ are constants. The solid line represents the regression function at a high temperature. All the prototypes are co-incident and the effective model size is one. The dotted and dashed lines indicate the optimal regression functions of model sizes two and three respectively. Note that the three functions progressively model the data better.

The phenomenon of phase transitions in the annealing process is also useful for speeding up the DA implementation. We will demonstrate in this section that we can analytically predict the temperatures at which discrete phase transitions take place in the system. Predicting the critical temperatures helps us to avoid the necessity of bringing the system to thermal equilibrium at each intermediate temperature. It allows us to "accelerate" in between consecutive critical temperatures of the system, thus speeding up our algorithm considerably. In particular, the annealing process may be initialized at a temperature which is slightly higher than the first critical temperature. For simplicity, we restrict the derivation of the critical temperatures in this section to the special case where the local regression models are constants. The results can easily be extended to other cases such as linear or polynomial models.

As $T \to \infty$, the globally optimal solution is easily found, where all prototype vectors coincide and where all the simple (constant) regression models coincide at the global centroid of the **Y** data set. This extremely simple solution will remain a stable minimum of L over an interval of decreasing T. However, when T reaches a critical value T_c, the minimum will turn into a saddle point or a local maximum. At this point, it becomes advantageous to "split" the prototypes, and the system undergoes a bifurcation wherein the free energy cost is decreased by increasing the effective size of the solution.

Insights into the mechanism for growth in the effective model size are obtained by analyzing the conditions for bifurcation or phase transition. In [29], it was shown for the clustering problem that the first bifurcation occurs at the critical temperature corresponding to $T_c = 2\lambda_{\max}$, where $\lambda_{\max}$ is the principal eigenvalue of the data covariance matrix, C_{xx}. Moreover, the split is initiated along the principal data axis. It thus related the critical temperature to the data variance. It was later shown in [26] that all subsequent bifurcations occur in a similar fashion, dependent on the a posteriori covariance of the data probabilistically "owned" by the cluster undergoing the split. Here, we present the necessary condition for the first bifurcation in the annealing process for regression function design. We note that a more general condition governing all bifurcations in the solution process can be derived using the calculus of variations. The first bifurcation increases the effective size of the solution from one to two. Initial splits into more than two distinct prototypes may also occur if certain symmetries exist, but we ignore them here for conciseness. Thus, we consider an optimization problem with two prototypes, each associated with a constant regression model, represented by the vector $\mathbf{y}_i \in \mathcal{R}^n$. Prior to the split, both the regression model vectors are at the global **Y** centroid μ_y, and the prototype vectors are at the global **X** centroid μ_x. The phase transition occurs when the Hessian matrix (consisting of second order partial derivatives) of L is no longer positive definite at the solution point. It can be shown that the Hessian is given by

$$H_e = \begin{bmatrix} \gamma^2 C_{xx} & -\gamma^2 C_{xx} & -\frac{\gamma}{T} C_{xy}^T & \frac{\gamma}{T} C_{xy}^T \\ -\gamma^2 C_{xx} & \gamma^2 C_{xx} & \frac{\gamma}{T} C_{xy}^T & -\frac{\gamma}{T} C_{xy}^T \\ -\frac{\gamma}{T} C_{xy} & \frac{\gamma}{T} C_{xy} & \frac{1}{T} I & 0 \\ \frac{\gamma}{T} C_{xy} & -\frac{\gamma}{T} C_{xy} & 0 & \frac{1}{T} I \end{bmatrix}, \tag{14}$$

where I and 0 are identity and zero matrices, respectively,

$$C_{xx} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mu_{\mathbf{x}})(\mathbf{x}_i - \mu_{\mathbf{x}})^T \tag{15}$$

is the empirical covariance matrix of **X**, and

$$C_{xy} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mu_{\mathbf{x}})(\mathbf{y}_i - \mu_{\mathbf{y}})^T \tag{16}$$

is the empirical cross-covariance matrix relating **X** and **Y**.

The critical temperature marks a point of transition from positive-definite Hessian. As the Hessian is evolving continuously, we seek the value of T for which its determinant vanishes. It can be easily established that this requires that

$$\left| C_{xx} - \frac{2}{T} C_{xy}^T C_{xy} \right| = 0. \tag{17}$$

The first (largest) value of T satisfying (17) is:

$$T_c = 2\lambda_{\max}\left(C_{xx}^{-1} C_{xy}^T C_{xy} \right), \tag{18}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue of its matrix argument. Moreover, the split is initiated along the direction of the principal eigenvector of $C_{xx}^{-1} C_{xy}^T C_{xy}$.

These results make an interesting connection with previous work in the area of data clustering. Consider a special degenerate case of the regression problem, where **X** = **Y**. While this may not be an interesting regression problem, a careful look at (3) reveals that the regression cost function for this case is exactly the cost of clustering (vector quantization) in the **X** space. The bifurcation condition of (18) for this special case reduces to $T_c = 2\lambda_{\max}(C_{xx})$ which, not surprisingly, is exactly the bifurcation condition reported previously for the problem of data clustering [29].

We can gain further intuition by specializing the results to scalar **X** and **Y**. Here, the condition in (18) reduces to

$$T_c = 2\rho^2 \sigma_y^2, \tag{19}$$

where σ_y^2 is the variance of **Y** and ρ is the correlation coefficient of **X** and **Y**. The "effective variance," $\rho^2 \sigma_y^2$, which determines critical temperatures in the annealing process, is reduced with respect to the clustering case as the pair (**X**, **Y**) becomes less correlated.

4 Results

In this section, we summarize our experiments comparing the performance of the proposed DA approach for NP-based regression with the conventional piecewise regression approach of CART. Recall that regular CART is severely restricted in that the partition is constrained to be tree-structured with partition boundaries that are parallel to the coordinate axes. The latter restriction which prevents regular CART from exploiting dependencies between the features of **X** can be overcome by adopting an extension of CART that allows the boundaries between regions to be arbitrary linear hyperplanes. While this extension admits a larger class of input space partitioning, and will therefore potentially reduce the approximation error, the complexity of the design method for the extended structure [15] grows as N^2, where N is the size of the training set. Consequently, the extended form of CART is impractical unless the training set is sufficiently short. In this section, whenever distinction is needed, we will refer to regular CART as CART1, and to its extended form as CART2. Our implementation of CART consists of growing a large "full tree" and then pruning it down to the root node using the BFOS algorithm [8]. The sequence of CART regression trees is obtained during the pruning process. It is known that the pruning phase is optimal given the fully grown tree.

The NP structure of the DA approach, unlike CART1, is capable of implementing more complex partitions of the input space and exploiting nonlinear dependencies between the components of the features vector. Unlike CART2, the complexity of the DA method grows linearly with the size of the training set. Moreover, the DA algorithm optimizes all the parameters of the regression function simultaneously, while avoiding many shallow local minima that trap greedy methods.

In all the simulation examples, we have used the simpler piecewise constant models. However, the DA regression method's improvements over CART are similarly obtained for other types of local models, e.g., linear or polynomial. In our implementation of the DA method (see pseudo-code in Table 1), we used the annealing schedule: $q(T) = 0.95T$. The initial temperature, T_i, is $1.1T_c$, where T_c is the first critical temperature given by (18). The annealing process is stopped and the quenching step is performed when the entropy of the partition (7) is less than $H_f = 0.01$. At such a small value of entropy, the partition is almost "hard," so that further annealing will not change the prototypes or the local regression models significantly. The steps to minimize the Helmholtz free energy at each temperature were performed until the fractional improvement[5] in the free energy was less than $L_{th} = 10^{-5}$.

We demonstrate the usefulness of our approach on several synthetic and real-world datasets from the regression literature.

4.1 Synthetic Examples

Through the following experiments on synthetic datasets, we will establish that the proposed DA method generates robust, low-complexity regression functions that outperform CART-generated regression functions. In all the synthetic examples, the input, $\mathbf{x} = (x_0, x_1)$, is two-dimensional and the output, $\mathbf{y}$, is one-dimensional.

First, we compare the performance of the CART and DA-based regression function design methods for the problem of approximating known functions from noisy input-output data. The functions used in this subsection have been used in the literature [16], [6] as benchmarks to compare regression function design methods.

We consider, first, the function given by

$$g^{(1)}(\mathbf{x}) = 10.39((x_0 - 0.4)\cdot(x_1 - 0.6) + 0.36). \qquad (20)$$

This function involves a simple interaction between the two features, x_0 and x_1. We generated a training set (size 200), a test set (size 3,000), and a validation set (size 1,000).[6] The training set in this experiment was deliberately made small to facilitate comparison with the CART2 design approach which becomes prohibitively complex when large training sets (>500) are involved. We added a zero-mean Gaussian noise with variance 0.1585 to the output variable in the training, test, and validation sets. CART1, CART2, and DA were used to design regression functions from the noisy data. The optimal regression models generated by the design methods were obtained and compared as follows: For each algorithm, we obtained a sequence of regression functions of increasing model order by the appropriate mechanism—greedy tree growing and pruning for CART1 and CART2 and phase transitions for DA. We then chose the best model for each design method by selecting the member of the corresponding sequence of functions that performed best over the independent validation set. The optimal regression functions for each method were then compared over the test set (which is independent of both the training and validation sets).

TABLE 2
OPTIMAL AVERAGE SQUARED APPROXIMATION ERROR MEASURED OVER THE TEST SET AND OPTIMAL MODEL ORDER FOR REGRESSION MODELS FOR NOISE CORRUPTED DATA

Dataset	CART (Model Order)	DA (Model Order)
$g^{(1)}()$	0.100 (59)	0.094 (57)
$g^{(2)}()$	0.112 (62)	0.106 (62)
$g^{(3)}()$	0.305 (214)	0.166 (59)

In the above experiment, the DA approach generates the best regression model with an average squared-error of 0.25. Further, the CART2 model which is marginally worse (error of 0.26) is considerably better than the CART1 function (error of 0.49). This is because the simple structure of the CART1 partition is inadequate when there is interaction between the feature variables. Note that the CART2 method performs quite well. This is often the case for small training set sizes. However, for larger training sets, CART2 is impractical because of its high design complexity.

In the following experiments, where we use larger training sets (size 1,000) we must abandon the CART2 method (in these examples, our attempt to simulate CART2 was aborted after 36 hours on a SPARC20 computer). Thus, the comparison is restricted to the CART1 and DA methods. We will return to consider CART2 when we experiment with smaller, real-world data sets.

The experiments for the larger training sets were performed over three functions. In addition to the function, $g^{(1)}$ of (20), we define

$$g^{(2)}(\mathbf{x}) = 24.234\mathbf{r}(0.75 - \mathbf{r}) \qquad (21)$$

where $\mathbf{r} = (x_0 - 0.5)^2 + (x_1 - 0.5)^2$ and

$$g^{(3)}(\mathbf{x}) = 42.659(0.1 + (x_0 - 0.5)(0.05 + (x_0 - 0.5)^4 - 10.0(x_0 - 0.5)^2(x_1 - 0.5)^2 + 5.0(x_1 - 0.5)^4)) \qquad (22)$$

For all three functions, a zero-mean noise with variance 0.063 was added to the output variable. The entire data set was divided into training (size 1,000), test (size 3,000), and validation (size 1,000) sets. The results tabulated in Table 2 show the average squared-error over the test set for each model and the corresponding model order in parentheses. The "model order" is measured by the number of regions

[5] Fractional improvement of a cost function is the ratio between the improvement in the cost resulting from an iteration and the absolute value of the cost before the iteration.

[6] All synthetic and real-world data used in this paper are available on the World Wide Web at http://scl.ece.ucsb.edu/datasets/.

TABLE 3
OPTIMAL AVERAGE SQUARED APPROXIMATION ERROR MEASURED OVER THE TEST SET AND OPTIMAL MODEL ORDER FOR MULTIMODAL GAUSSIAN DATASETS

Dataset	CART (Model Order)	DA (Model Order)
1	12.0 (21)	11.1 (8)
2	12.7 (30)	11.7 (10)
3	11.5 (22)	10.7 (13)
4	12.0 (33)	11.6 (14)
5	15.1 (59)	14.4 (9)
6	13.6 (47)	12.9 (11)
7	13.5 (46)	11.1 (20)
8	11.9 (27)	11.1 (14)

used by the best model produced by the method in terms of performance on the validation set. The DA approach consistently generates superior regression models (lower squared-approximation error and lower model order) compared to the CART approach.

Note that in the cases of the first two functions, which are relatively simple, the performance improvements of DA over CART are small. This is due to the fact that there is not much to be gained over CART for these simple functions. The function $g^{(3)}$, on the other hand, is more complex, and in this case, the DA method shows its real potential and outperforms CART by a factor of almost two, while maintaining a much simpler partition.

The second group of synthetic datasets also involves two-dimensional feature vectors $\mathbf{x} = (x_0, x_1)$ and one-dimensional output $\mathbf{y}$. The input components, $\mathbf{x}_0$ and $\mathbf{x}_1$, are uniformly distributed in the interval (0, 1). The output, $\mathbf{y}$, is a weighted sum of six normalized Gaussian-shaped functions of $\mathbf{x}$, each with an individual "center," variance, and weight. By choosing different centers, variances, and weights for the Gaussians, we created a number of data sets, each consisting of training and validation sets of size 1,000 each, and a test set of size 3,000. The output samples were corrupted by a zero-mean Gaussian noise with variance 10.0.

As with the first group of data sets, DA and CART processed the training set to design a sequence of regression functions of increasing complexity. Performance on the validation set was used to select a regression function from the sequence, and the overall performance was evaluated on the independent test set. The experiments were conducted over more than 40 different data sets. In Table 3, we have randomly chosen a sample of the data sets and tabulated the performance. Note that we only compare our method with standard CART1, since CART2 is too complex to use with training sets of this size. Clearly, in all the examples, DA demonstrates consistent improvement over CART1.

One notable advantage of CART over many other regression methods is that its decision tree structure offers an interesting and useful way to interpret the data. To a lesser degree, the NP-based regression function can also be used for such interpretation. In particular, the prototypes in the NP-regression function may be interpreted as "typical feature vectors," each associated with a particular, locally active regression model. The regression process effectively models the training data in terms of the "typical data," offering a way to interpret large volumes of data.

4.2 Real-World Examples

Here we present the results of experiments that compare the DA design approach against CART over data sets from real-world regression applications. This data is taken from the StatLib database,[7] which has been extensively used by researchers in benchmarking the relative performance of competing regression methods. In some of the following experiments, due to the unavailability of sufficient data for proper validation, we make use of cross-validation to determine the optimal model order.

In the first experiment, we consider the problem of predicting the value of homes in the Boston area from a variety of parameters [13]. The training set consists of data from 506 homes. The output in this case is the median price of a home, with the input consisting of a vector of 13 scalar features believed to influence the price. The objective is to minimize the average squared error in price prediction. Since the features have different dynamic ranges, we normalized each to have unit variance. We then applied the two CART design methods and the DA design method to the training set and obtained a sequence of regression functions for each design method. Each sequence of CART functions was obtained by growing a full tree and optimally pruning this tree using data from the training set. The sequence of DA solutions was obtained via the phenomenon of phase transitions.

The final model order for each design method was determined by a cross-validation method. Our implementation of cross-validation for CART1 and CART2 follows the approach outlined in [32]. The training set was divided into 10 (roughly) equal parts. Nine of these parts were used to grow and prune a sequence of trees. Each pruned tree obtained during design was tested on the remaining data. The division of the entire data into training and test sets in this manner can be done in ten ways. The test set results were averaged over the ten regression trees thus designed. Averaging was done for a fixed value of the cost-complexity slope (refered to as α in [32]). The model orders corresponding to the optimal value of α (that which result in the best averaged test set error) are 8 and 5, for CART1 and CART2, respectively. Cross-validation for the DA design method is implemented similarly—DA models of different orders are designed for each subdivision of the data into training and test sets. The error on the test sets are averaged for functions of the same model order designed on each training set and the best model order (corresponding to

[7] The StatLib data set archive is accessible on the World Wide Web at http://lib.stat.cmu.edu/data sets/.

TABLE 4
AVERAGE SQUARED PREDICTION ERROR FOR HOUSING PRICES IN THE BOSTON AREA

K	CART1	CART2	DA
1	84.4	84.4	84.4
2	46.2	43.2	34.4
3	31.8	33.0	25.0
4	25.7	26.1	16.9
5	20.7	23.2	14.4
6	17.9	21.9	11.0
7	15.6	20.8	10.8
8	13.6	19.7	10.7
9	12.5	18.8	8.6
10	11.8	18.1	8.5
Selected model	13.6 (8)	23.2 (5)	11.0(6)

Comparison of average squared-errors for the standard CART1, its extension CART2, and the proposed DA method. K is the number of partition regions for each model. The last row shows the squared-error and model order of the regression function selected by cross-validation.

the lowest averaged test set error) is determined to be 6 for the Boston data.

In Table 4, we have compared the squared-error in predicting the house price using the standard CART1 and its extended form CART2, with the performance of the proposed DA method. Although each design method generates a sequence of regression functions of increasing complexity, we have tabulated here, only the training errors for each method for model orders from 1 to 10. As mentioned earlier, the model orders chosen by cross-validation for the CART1, CART2, and DA methods are 8, 5, and 6, respectively. The performance of the optimal models is also shown in Table 4 and demonstrates the superior performance of the DA design method.

Also note that CART1 outperforms CART2 in several cases, despite the fact that CART2 is a potentially more powerful regression structure. These results are indicative of the prevalence of poor local optima which trap standard methods.

TABLE 5
MEAN-SQUARED ERROR FOR PREDICTION OF THE AGE-ADJUSTED MORTALITY RATE PER 100,000 PEOPLE FROM VARIOUS ENVIRONMENTAL FACTORS

K	CART1	CART2	DA
1	3805.13	3805.13	3805.13
2	2427.40	2087.0	2003.4
3	1786.90	1532.19	976.18
4	1381.08	1323.50	775.36
5	1122.68	1174.17	694.27
6	938.91	1050.55	603.46
7	792.91	917.2	551.85
Selected Model	1381.08 (4)	1532.19 (3)	976.18 (3)

Comparison of CART1, CART2, and DA. K is the number of partition regions for each model. For the model eventually selected by each design method, we quote the error with the corresponding model order in parentheses.

TABLE 6
MEAN-SQUARED APPROXIMATION ERROR FOR THE FAT CONTENT OF A MEAT SAMPLE FROM 100 SPECTROSCOPIC MEASUREMENTS

K	CART1		CART2		DA	
	TR	TE	TR	TE	TR	TE
1	159.03	168.20	159.03	168.20	159.03	168.20
2	114.03	143.72	62.85	85.06	37.56	38.60
3	97.54	129.90	51.16	75.23	22.80	21.90
4	83.17	148.40	41.41	73.56	16.08	15.17
5	73.28	131.83	37.70	66.68	15.46	15.93
6	62.29	126.44	34.24	72.33	15.33	16.66
7	55.51	117.24	30.83	65.81	12.90	24.80
8	48.18	121.85	28.54	74.91	12.42	14.98
9	42.58	122.04	25.69	84.03	10.95	18.41
10	37.62	119.44	23.56	73.64	11.56	13.87
Selected model	92.44 (35)		62.09 (20)		18.41 (9)	

The performance of CART1 and CART2 is compared with that of the proposed DA method, both inside the training set (TR) and on a test set (TE). K is the number of partition regions used to represent the data. For the model eventually selected by each design method, we quote the test set error with the corresponding model order in parentheses.

The data set for the second example was taken from the environmental sciences. We consider the problem of predicting the age-adjusted mortality rate per 100,000 people in a locality from 15 factors that may have possibly influenced it. Some of these factors are related to the levels of environmental pollution in the locality, while others are measurements of nonenvironmental/social parameters. This data set has been used by numerous researchers since its introduction in the early 1970s [20]. As there are data on only 60 localities, we have used a cross-validation method to determine the optimal model order of each method for this data set. Again, we use 10-fold cross-validation to obtain our results. In Table 5, we compare the regression error obtained by the DA-based method with CART1 and CART2. The CART results were obtained by growing and pruning a full tree from the training set. The selected model order which is obtained by cross-validation is determined to be 4 for CART1 and

TABLE 7
APPROXIMATE COMPUTATION TIME (IN MINUTES) ON A SUN SPARC20 SYSTEM, FOR REGRESSION FUNCTION DESIGN FOR THE TRAINING DATA EXAMPLES OF SECTION 4.2

Dataset	CART1	CART2	DA
Boston Data	30	510	580
Pollution Data	3	6	35
Tecator data	60	200	500

3 for CART2. The corresponding average squared-error results are 1,381.08 and 1,532.19. We note that the best cross-validated DA model is of order 3 and its average error is 976.18.

The third regression data set is drawn from an application in the food sciences. The problem is that of efficient estimation of the fat content of a sample of meat. (Techniques of analytical chemistry can be used to measure this quantity directly, but it is a slow and time-consuming process.) We used a data set of quick measurements by the Tecator Infratec Food and Feed Analyzer which measures the absorption of electro-magnetic waves in 100 different frequency bands, and the corresponding fat content as determined by analytical chemistry. As suggested by the data providers, we divided the data into a training set of size 129, a validation set of 43, and a test set of size 43. We then applied CART1, CART2, and DA to the training set for different model sizes. In Table 6, we compare the average squared approximation error obtained over the training and test sets for model orders from 1 to 10. We reemphasize here that in our experiment, we designed CART regression functions by growing a full tree by a sequence of splits of the training set and pruning the full tree back to the root node using the BFOS algorithm. The CART results in the table are for trees obtained during pruning. The optimal model order is chosen by chosing the tree in the pruned sequence that performs best over the validation set. The optimal model order for CART1 was 35 and its average error over the test set was 92.44. The more complex CART2 method performs substantially better. It chooses a model order of 20, which gives an average test error of 62.1. The DA method out-performs both the CART algorithms. The best validation performance was obtained for the DA-designed model with 9 regions. This model gives an average test error of 18.41. The excellent performance of the DA method both inside and outside the training set confirms its expected good generalization capabilities.

4.3 Note on Design Complexity

Although the DA method achieves substantial performance gains over the CART-based regression function, these gains are obtained at the expense of additional computational complexity of the design process. As noted earlier, the complexity of DA and CART1 grows linearly with the number of training vectors and with the dimensionality of the feature space.

Typically, the computation time DA varies between 8 and 20 times that of CART1, depending on the optimal model order for the given data set. In our simulations, the complexity ratio was 15 on the average.

The complexity of CART2, on the other hand, grows *quadratically* with the number of training vectors and linearly with the dimensionality of the feature space. Therefore, while CART2 is less complex than DA on small training data such as in the examples given in Section 4.2 (see Table 7), it becomes prohibitively complex on midsize to large training sets, as in the synthetic examples of Section 4.1.

Table 7 shows the approximate time (in minutes) spent by each one of the competing methods on regression function design for the real-world datasets of Section 4.2. The software was written in C, and the simulations were performed on a Sun microsystems SPARC20 computer.

5 CONCLUSIONS

In this paper, we have shown that the nearest prototype model, when optimized via the powerful technique of deterministic annealing, is an efficient and useful structure for regression modeling. We have presented a novel approach for the design which takes as its starting point the paradigm of randomized space partitioning. The method is based on information-theoretic principles and uses entropy-constrained optimization within a deterministic annealing framework. Further intuition into the workings of the method is conveyed through analogy to statistical physics. In the annealing process, we identified and analyzed the phenomenon of phase transitions. Each phase corresponds to a distinct model size with the transitions corresponding to increases in model size occurring at predictable "critical temperatures." Through this phenomenon, as well as through its ability to avoid many shallow local optima, the DA method provides efficient parsimonious models and thereby attacks the problem of generalization. Our experimentation shows that the DA method generalizes significantly better than both the standard CART approach and the extended form of CART that allows arbitrary hyperplane partitions. Further, the algorithm generates an optimal model whose complexity (model order) is lower than that of the optimal CART model.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Shivkumar Chandrasekaran for helpful discussions on local optimization methods.

This work was supported in part by the National Science Foundation under grant no. NCR-9314335, the University of California MICRO program, ACT Networks, Advanced Computer Communications, Cisco Systems, DSP Group, DSP Software Engineering, Fujitsu Laboratories of America, General Electric, Hughes Electronics, Intel, Nokia Mobile Phones, Qualcomm, Rockwell International, and Texas Instruments. David Miller was supported by NSF Career Award NSF IRI-9624870. Most of this work was done when A.V. Rao and D.J. Miller were graduate students in the Department of Electrical and Computer Engineering, University of California, Santa Barbara.

References

[1] H. Akaike, "A New Look at Statistical Model Identification," *IEEE Trans. Automatic Control*, vol. 19, pp. 716-723, Dec. 1974.

[2] R. Bellman and R. Roth, "Curve Fitting by Segmented Straight Lines," *J. Am. Statistical Assoc.*, vol. 64, pp. 1,079-1,084, 1969.

[3] L. Breiman and J.H. Friedman, "Estimating Optimal Transformations for Multiple Regression," *Computer Science and Statistics: Proc. 16th Symp. Interface*, pp. 121-134, 1985.

[4] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees.* Belmont, Calif.: Wadsworth, 1984.

[5] J. Buhmann and H. Kuhnel, "Vector Quantization With Complexity Costs," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1,133-1,145, 1993.

[6] V. Cherkassky, Y. Lee, and H. Lari-Najafi, "Self-Organizing Network for Regression: Efficient Implementation and Comparative Evaluation," *Proc. Int'l Joint Conf. Neural Networks*, vol. 1, pp. 79-84, 1991.

[7] P.A. Chou, "Optimal Partitioning for Classification and Regression Trees," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 340-354, Apr. 1991.

[8] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Optimal Pruning With Applications to Tree-Structured Source Coding and Modeling," *IEEE Trans. Inform. Theory*, vol. 35, pp. 299-315, 1989.

[9] T.M. Cover, "Estimation by the Nearest Neighbor Rule," *IEEE Trans. Inform. Theory*, vol. 14, pp. 50-55, 1968.

[10] J.H. Friedman, "Multiple Adaptive Regression Splines," *Ann. Stat.* vol. 19, pp. 1-141, 1991.

[11] J.H. Friedman and W. Stuetzle, "Projection Pursuit Regression," *J. Am. Statistical Assoc.* vol. 76, pp. 817-823, 1981.

[12] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression.* Boston: Kluwer Academic Publishers, 1992.

[13] D. Harrison and D.L. Rubinfeld, "Hedonic Prices and the Demand for Clean Air," *J. Environ. Economics & Management*, vol. 5, pp. 81-102, 1978.

[14] B. Hassibi and D.G. Stork, "Second Derivatives for Network Pruning: Optimal Brain Surgeon," *Advances in Neural Information Processing Systems*, vol. 5, pp. 164-171, 1993.

[15] G.E. Hinton and M. Revow, "Using Pairs of Data Points to Define Splits for Decision Trees," *Advances in Neural Information Processing Systems*, vol. 8, pp. 507-513, 1995.

[16] J.-H. Hwang, S.-R. Lay, M. Maechler, R.D. Martin, and J. Schimert, "Regression Modeling in Back-Propagation and Projection Pursuit Learning," *IEEE Trans. Neural Networks*, vol. 5, no. 3, pp. 342-353, 1994.

[17] T. Kohonen, "An Introduction to Neural Computing," *Neural Networks*, vol. 1, no. 1, pp. 3-16, 1988.

[18] Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," *IEEE Trans. Comm.*, vol. 28, pp. 84-95, 1980.

[19] W.-Y. Loh and N. Vanichsetakul, "Tree-Structured Classification via Generalized Discriminant Analysis (With Discussion)," *J. Am. Statistical Assoc.*, vol. 83, no. 403, pp. 715-727, 1988.

[20] G.C. McDonald and R.C. Schwing, "Instabilities of Regression Estimates Relating Air Pollution to Mortality," *Technometrics*, vol. 15, pp. 463-482, 1973.

[21] D. Miller, A. Rao, K. Rose, and A. Gersho, "A Global Optimization Technique for Statistical Classifier Design," *IEEE Trans. Signal Proc.*, vol. 44, no. 12, pp. 3,108-3,122, 1996.

[22] J. Moody and C.J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Computation*, vol. 1, no. 2, pp. 281-94, Summer 1989.

[23] A. Rao, D. Miller, K. Rose, and A. Gersho, "A Generalized VQ Method for Combined Compression and Estimation," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2,032-2035, 1996.

[24] A. Rao, D. Miller, K. Rose, and A. Gersho, "Mixture of Experts Regression Modeling by Deterministic Annealing," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2,811-2,820, Nov. 1997.

[25] J. Rissanen, "Stochastic Complexity and Modeling," *Ann. Stat.*, vol. 14, pp. 1,080-1,100, 1986.

[26] K. Rose, "A Mapping Approach to Rate-Distortion Computation and Analysis," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1,939-1,952, 1994.

[27] K. Rose, E. Gurewitz, and G.C. Fox, "Constrained Clustering as an Optimization Method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 785-794, 1993.

[28] K. Rose, E. Gurewitz, and G.C. Fox, "Statistical Mechanics and Phase Transitions in Clustering," *Phys. Rev. Lett.*, vol. 65, no. 8, pp. 945-948, 1990.

[29] K. Rose, E. Gurewitz, and G.C. Fox, "Vector Quantization by Deterministic Annealing," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1,249-1,258, 1992.

[30] B. Silverman, "Density Estimation for Statistics and Data Analysis," *Monographs on Statistics and Applied Probability.* London: Chapman and Hall, 1986.

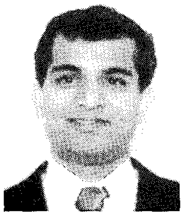
[31] G.R. Terrell and D.W. Scott, "Variable Kernel Density Estimation," *Ann. Stat.*, vol. 20, no. 3, pp. 1,236-1,265, 1992.

[32] W.N. Venables and W.B. Ripley, *Modern Applied Statistics With S-Plus.* New York: Springer-Verlag, 1994.

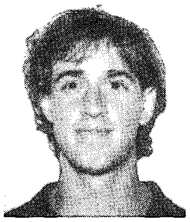
[33] S.M. Weiss, R.S. Galen, and P.V. Tadepalli, "Optimizing the Predictive Value of Diagnostic Decision Rules," *Proc. Nat'l Conf. Artificial Intelligence*, AAAI, pp. 18.1.1-14, Seattle, 1987.

[34] X. Wu and K. Zhang, "A Better Tree-Structured Vector Quantizer," *Proc. Data Compression Conf.*, pp. 392-401. Los Alamitos, Calif.: IEEE Computer Society Press, 1991.

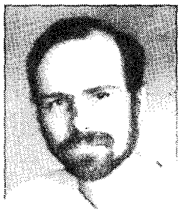
[35] J. Zhao and J. Shawe-Taylor, "Neural Network Optimization for Good Generalization Performance," *Proc. Int'l Conf. Artificial Neural Networks*, pp. 561-564, 1994.



Ajit V. Rao (S'93) received the BTech degree in electronics and communication engineering in 1992 from the Indian Institute of Technology, Madras, India, and the MS and PhD degrees in electrical and computer engineering in 1993 and 1998, respectively, from the University of California, Santa Barbara. He was an intern in the Speech Coding group at Texas Instruments Inc., Dallas, Texas, and is currently employed as a research engineer with SignalCom Inc., Santa Barbara, California. His research interests lie in speech and image coding, speech recognition, digital communications, and statistical pattern recognition.



David J. Miller (S'87, M'95) received the BSE degree from Princeton University in 1987, the MSE degree from the University of Pennsylvania in 1990, and the PhD degree from the University of California at Santa Barbara in 1995, all in electrical engineering. From January 1988 through January 1990, he was employed by General Atronics Corporation in Wyndmoor, Pennsylvania. Since August 1995, Dr. Miller has been an assistant professor of electrical engineering at the Pennsylvania State University. Dr. Miller's research interests include source coding and coding over noisy channels, image compression, statistical pattern recognition, and neural networks. In 1996, Dr. Miller received the National Science Foundation Career Award for the continuation of his research on statistical pattern recognition and learning algorithms for neural networks.



Kenneth Rose (S'85, M'91) received the BSc (summa cum laude) and MSc (magna cum laude) degrees in electrical engineering from Tel-Aviv University, Israel, in 1983 and 1987, respectively, and the PhD degree in electrical engineering from the California Institute of Technology in 1990. From July 1983 to July 1988, he was employed by Tadiran Ltd., Israel, where he carried out research in the areas of image coding, transmission through noisy channels, and general image processing. From September 1988 to December 1990, he was a graduate student at Caltech. In January 1991, he joined the Department of Electrical and Computer Engineering, University of California, Santa Barbara, where he is currently an associate professor. His research interests are in information theory, source and channel coding, pattern recognition, image coding and processing, and nonconvex optimization in general. Dr. Rose was corecipient of the William R. Bennett Prize Paper Award of the IEEE Communications Society (1990).

Allen Gersho (S'58, M'64, SM'78, F'81) is a professor of electrical and computer engineering at the University of California, Santa Barbara. He received his BS from MIT in 1960 and PhD from Cornell University in 1963. He was at Bell Laboratories from 1963 to 1980. His current research activities are in signal compression methodologies and algorithm development for speech, audio, image, and video coding. He holds patents on speech coding, quantization, adaptive equalization, digital filtering, and modulation and coding for voiceband data modems. He is coauthor with R.M. Gray of the book, *Vector Quantization and Signal Compression* (Kluwer Academic Publishers, 1992) and coeditor of two books on speech coding. He served as a member of the Board of Governors of the IEEE Communications Society from 1982 to 1985 and is a member of various IEEE technical, award, and conference management committees. He has served as Editor of *IEEE Communications Magazine* and Associate Editor of the *IEEE Transactions on Communications*. He received NASA "Tech Brief" awards for technical innovation in 1987, 1988, and 1992. In 1980, he was corecipient of the Guillemin-Cauer Prize Paper Award from the Circuits and Systems Society. In 1981, he became a Fellow of the IEEE. He received the Donald McClennan Meritorious Service Award from the IEEE Communications Society in 1983, and in 1984 he was awarded an IEEE Centennial Medal. In 1992, he was corecipient of the 1992 Video Technology Transactions Best Paper Award from the IEEE Circuits and Systems Society.