Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems

KENNETH ROSE, MEMBER, IEEE

Invited Paper

The deterministic annealing approach to clustering and its extensions has demonstrated substantial performance improvement over standard supervised and unsupervised learning methods in a variety of important applications including compression, estimation, pattern recognition and classification, and statistical regression. The method offers three important features: 1) the ability to avoid many poor local optima; 2) applicability to many different structures/architectures; and 3) the ability to minimize the right cost function even when its gradients vanish almost everywhere, as in the case of the empirical classification error. It is derived within a probabilistic framework from basic information theoretic principles (e.g., maximum entropy and random coding). The application-specific cost is minimized subject to a constraint on the randomness (Shannon entropy) of the solution, which is gradually lowered. We emphasize intuition gained from analogy to statistical physics, where this is an annealing process that avoids many shallow local minima of the specified cost and, at the limit of zero "temperature," produces a nonrandom (hard) solution. Alternatively, the method is derived within rate-distortion theory, where the annealing process is equivalent to computation of Shannon's rate-distortion function, and the annealing temperature is inversely proportional to the slope of the curve. This provides new insights into the method and its performance, as well as new insights into rate-distortion theory itself. The basic algorithm is extended by incorporating structural constraints to allow optimization of numerous popular structures including vector quantizers, decision trees, multilayer perceptrons, radial basis functions, and mixtures of experts. Experimental results show considerable performance gains over standard structure-specific and application-specific training methods. The paper concludes with a brief discussion of extensions of the method that are currently under investigation.

Manuscript received November 1, 1997; revised April 17, 1998. This work was supported in part by the National Science Foundation under Grant NCR-9314335, the University of California MICRO Program, ACT Networks, Inc., Advanced Computer Communications, Cisco Systems, Inc., DSP Group, Inc., DSP Software Engineering, Inc., Fujitsu Laboratories of America, Inc., General Electric Company, Hughes Electronics Corp., Intel Corp., Nokia Mobile Phone, Qualcomm, Inc., Rockwell International Corp., and Texas Instruments, Inc.

The author is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106 USA.

Publisher Item Identifier S 0018-9219(98)07860-8.

Keywords*—*Classification, clustering, compression, deterministic annealing, maximum entropy, optimization methods, regression, vector quantization.

I. INTRODUCTION

There are several ways to motivate and introduce the material described in this paper. Let us place it within the neural network perspective, and particularly that of learning. The area of neural networks has greatly benefited from its unique position at the crossroads of several diverse scientific and engineering disciplines including statistics and probability theory, physics, biology, control and signal processing, information theory, complexity theory, and psychology (see [45]). Neural networks have provided a fertile soil for the infusion (and occasionally confusion) of ideas, as well as a meeting ground for comparing viewpoints, sharing tools, and renovating approaches. It is within the ill-defined boundaries of the field of neural networks that researchers in traditionally distant fields have come to the realization that they have been attacking fundamentally similar optimization problems.

This paper is concerned with such a basic optimization problem and its important variants or derivative problems. The starting point is the problem of clustering, which consists of optimal grouping of observed signal samples (i.e., a training set) for the purpose of designing a signal processing system. To solve the clustering problem one seeks the partition of the training set, or of the space in which it is defined, which minimizes a prescribed cost function (e.g., the average cluster variance). The main applications of clustering are in pattern recognition and signal compression. Given training samples from an unknown source, in the former application the objective is to characterize the underlying statistical structure (identify components of the mixture), while in the latter case a quantizer is designed for the unknown source. This paper describes the deterministic annealing (DA) approach to

clustering and its extension via introduction of appropriate constraints on the clustering solution, to attack a large and important set of optimization problems.

Clustering belongs to the category of unsupervised learning problems, where during training we are only given access to input samples for the system under design. The desired system output is not available. The complementary category of supervised learning involves a "teacher" who provides, during the training phase, the desired output for each input sample. After training, the system is expected to emulate the teacher. Many important supervised learning problems can also be viewed as problems of grouping or partitioning and fall within the broad class that we cover here. These include, in particular, problems of classification and regression. We shall further see that the methods described herein are also applicable to certain problems that do not, strictly speaking, involve partitioning.

The design of a practical system must take into account its complexity. Here we must, in general, restrict the complexity of the allowed partitions. This is typically done by imposing a particular structure for implementing the partition. Rather than allowing any arbitrary partition of the training set (or of the input space) we require that the partition be determined by a prescribed parametric function whose complexity is determined by the number of its parameters. For example, a vector quantizer (VQ) structure implements a Voronoi (nearest neighbor) partition of space and its complexity may be measured by the number of codevectors or prototypes. Another example is the partition obtained by a multilayer perceptron, whose complexity is determined by the number of neurons and synaptic weights. It is evident, therefore, that the design method will normally be specific to the structure, and this is indeed the case for most known techniques. However, the approach we describe here is applicable to a large and diverse set of structures and problems.

It is always instructive to begin with the simplest nontrivial problem instance in order to obtain an unobstructed insight into the essentials. We therefore start with the problem of clustering for quantizer design, where we seek the optimal partition into a prescribed number of subsets, which minimizes the average cluster variance or the mean squared error (MSE). In this case, we need not even impose a structural constraint. The Voronoi partition is optimal and naturally emerges in the solution. (Structurally constrained clustering is still of interest whenever one wants to impose a different structure on the solution. One such example is the tree-structured VQ which is used when lower quantizer complexity is required.) Not having to explicitly impose the structure is a significant simplification, yet even this problem is not easy. It is well documented (e.g., [41]) that basic clustering suffers from poor local minima that riddle the cost surface. A variety of heuristic approaches have been proposed to tackle this difficulty, and they range from repeated optimization with different initialization, and heuristics to obtain good initialization, to heuristic rules for cluster splits and merges, etc. Another approach was to use stochastic gradient techniques [16], particularly in conjunction with self-organizing feature maps, e.g., [22] and [107]. Nevertheless, there is a substantial margin of gains to be recouped by a methodical, principled attack on the problem as will be demonstrated in this paper for clustering, classification, regression, and other related problems.

The observation of annealing processes in physical chemistry motivated the use of similar concepts to avoid local minima of the optimization cost. Certain chemical systems can be driven to their low-energy states by annealing, which is a gradual reduction of temperature, spending a long time at the vicinity of the phase transition points. In the corresponding probabilistic framework, a Gibbs distribution is defined over the set of all possible configurations which assigns higher probability to configurations of lower energy. This distribution is parameterized by the temperature, and as the temperature is lowered it becomes more discriminating (concentrating most of the probability in a smaller subset of low-energy configurations). At the limit of low temperature it assigns nonzero probability only to global minimum configurations. A known technique for nonconvex optimization that capitalizes on this physical analogy is stochastic relaxation or simulated annealing [54] based on the Metropolis algorithm [68] for atomic simulations. A sequence of random moves is generated and the random decision to accept a move depends on the cost of the resulting configuration relative to that of the current state. However, one must be very careful with the annealing schedule, i.e., the rate at which the temperature is lowered. In their work on image restoration, Geman and Geman [34] have shown that, in theory, the global minimum can be achieved if the schedule obeys $T \propto 1/\log n$, where n is the number of the current iteration (see also the derivation of necessary and sufficient conditions for asymptotic convergence of simulated annealing in [42]). Such schedules are not realistic in many applications. In [100] it was shown that perturbations of infinite variance (e.g., the Cauchy distribution) provide better ability to escape from minima and allow, in principle, the use of faster schedules.

As its name suggests, DA tries to enjoy the best of both worlds. On the one hand it is deterministic, meaning that we do not want to be wandering randomly on the energy surface while making incremental progress on the average, as is the case for stochastic relaxation. On the other hand, it is still an annealing method and aims at the global minimum, instead of getting greedily attracted to a nearby local minimum. One can view DA as replacing stochastic simulations by the use of expectation. An effective energy function, which is parameterized by a (pseudo) temperature, is derived through expectation and is deterministically optimized at successively reduced temperatures. This approach was adopted by various researchers in the fields of graph-theoretic optimization and computer vision [10], [26], [33], [37], [98], [99], [108]. Our starting point here is the early work on clustering by deterministic annealing which appeared in [86] and [88]–[90]. Although strongly motivated by the physical analogy, the approach is formally

based on principles of information theory and probability theory, and it consists of minimizing the clustering cost at prescribed levels of randomness (Shannon entropy).

The DA method provides clustering solutions at different scales, where the scale is directly related to the temperature parameter. There are "phase transitions" in the design process, where phases correspond to the number of effective clusters in the solution, which grows via splits as the temperature is lowered. If a limitation on the number of clusters is imposed, then at zero temperature a hard clustering solution, or a quantizer, is obtained. The basic DA approach to clustering has since inspired modifications, extensions, and related work by numerous researchers including [6], [14], [47], [64], [70], [72], [73], [82], [91], [103], [106].

This paper begins with a tutorial review of the basic DA approach to clustering, and then goes into some of its most significant extensions to handle various partition structures [69], as well as hard supervised learning problems including classifier design [70], piecewise regression [78], and mixture of experts [82]. Another important theoretical aspect is the connection with Shannon's rate distortion (RD) theory, which leads to better understanding of the method's contribution to quantization and yields additional contributions to information theory itself [87]. Some of the currently investigated extensions, most notably for hidden Markov models and speech recognition [80], [81], will be briefly discussed.

II. Deterministic Annealing for Unsupervised Learning

A. Clustering

Clustering can be informally stated as partitioning a given set of data points into subgroups, each of which should be as homogeneous as possible. The problem of clustering is an important optimization problem in a large variety of fields, such as pattern recognition, learning, source coding, image, and signal processing. The exact definition of the clustering problem differs slightly from field to field, but in all of them it is a major tool for the analysis or processing of data without *a priori* knowledge of the distribution. The clustering problem statement is usually made mathematically precise by defining a cost criterion to be minimized. In signal compression it is commonly referred to as the distortion. Let x denote a source vector, and let $y(x)$ denote its best reproduction codevector from codebook Y. Denoting the distortion measure (typically, but not necessarily, the squared Euclidean distance) by $d(\cdot, \cdot)$, the expected distortion is

$$D = \sum_x p(x)\, d(x, y(x)) \approx \frac{1}{N} \sum_x d(x, y(x)) \quad (1)$$

where the right-hand side assumes that the source distribution may be approximated by a training set of N independent vectors.[1] In this case, the clustering solution is specified in terms of the codebook Y and an encoding rule for selecting the codevector which best matches an input vector. Virtually all useful distortion functions are not convex and are instead riddled with poor local minima [41]. Thus, clustering is a nonconvex optimization problem. While exhaustive search will find the global minimum, it is hopelessly impractical for all nontrivial distributions and reasonably large data sets.

As the clustering problem appears in very diverse applications, solution methods have been developed in different disciplines. In the communications or information-theory literature, an early clustering method was suggested for scalar quantization, which is known as the Lloyd algorithm [60] or the Max quantizer [65]. This method was later generalized to vector quantization, and to a large family of distortion measures [59], and the resulting algorithm is commonly referred to as the generalized Lloyd algorithm (GLA). For a comprehensive treatment of the subject within the areas of compression and communications see [36]. In the pattern-recognition literature, similar algorithms have been introduced including the ISODATA [4] and the K-means [63] algorithms. Later, fuzzy relatives to these algorithms were derived [9], [25]. All these iterative methods alternate between two complementary steps: optimization of the encoding rule for the current codebook, and optimization of the codebook for the encoding rule. When operating in "batch" mode (i.e., where the cost due to the entire training set is considered before adjusting parameters), it is easy to show that this iterative procedure is monotone nonincreasing in the distortion. Hence, convergence to a local minimum of the distortion (or of its fuzzy variant, respectively) is ensured.

1) Principled Derivation of Deterministic Annealing: Various earlier versions of the principled derivation of DA appeared in [86], [89], and [90]. The derivation was revised here to include more recent insights and to provide the most natural foundation for the following sections. A probabilistic framework for clustering is defined here by randomization of the partition, or equivalently, randomization of the encoding rule. Input vectors are assigned to clusters in probability, which we call the association probability. This viewpoint bears similarity to fuzzy clustering, where each data point has partial membership in clusters. However, our formulation is purely probabilistic. While we consider clusters as regular (nonfuzzy) sets whose exact membership is the outcome of a random experiment, one may also consider the fuzzy sets obtained by equating degree of membership with the association probability in the former (probabilistic) model. It is, thus, possible to utilize DA for both fuzzy and "regular" clustering design. We will not, however, make any use of tools or methods from fuzzy sets theory in this paper. On the other hand, the traditional framework for clustering is the marginal special case where all association probabilities are either zero or one. In the pattern recognition literature this is called "hard" clustering in contradistinction with the more recent ("soft") fuzzy clustering.

[1] The approximation of expected distortion by empirical distortion is practically unavoidable. In the sequel, whenever such approximation is obvious from the context, it will be used without repeating this explicit statement.

For the randomized partition we can rewrite the expected distortion (1) as

$$D = \sum_x \sum_y p(x, y)\, d(x, y) = \sum_x p(x) \sum_y p(y|x)\, d(x, y) \quad (2)$$

where $p(x, y)$ is the joint probability distribution, and the conditional probability $p(y|x)$ is the association probability relating input vector x with codevector y. At the limit where the association probabilities are hard and each input vector is assigned to a unique codevector with probability one, (2) becomes identical with the traditional hard clustering distortion (1).

Minimization of D of (2) with respect to the free parameters $\{y, p(y|x)\}$ would immediately produce a hard clustering solution, as it is always advantageous to fully assign an input vector to the nearest[2] codevector. However, we recast this optimization problem as that of seeking the distribution which minimizes D subject to a specified level of randomness. The level of randomness is, naturally, measured by the Shannon entropy

$$H(X, Y) = -\sum_x \sum_y p(x, y) \log p(x, y). \quad (3)$$

This optimization is conveniently reformulated as minimization of the Lagrangian

$$F = D - TH \quad (4)$$

where T is the Lagrange multiplier, D is given by (2), and H is given by (3). Clearly, for large values of T we mainly attempt to maximize the entropy. As T is lowered we trade entropy for reduction in distortion, and as T approaches zero, we minimize D directly to obtain a hard (nonrandom) solution.

At this point, it is instructive to pause and consider an equivalent derivation based on the principle of maximum entropy. Suppose we fix the level of expected distortion D and seek to estimate the underlying probability distribution. The objective is to characterize the random solution at gradually diminishing levels of distortion until minimal distortion is reached. To estimate the distribution we appeal to Jaynes's maximum entropy principle [52] which states: of all the probability distributions that satisfy a given set of constraints, choose the one that maximizes the entropy. The informal justification is that while this choice agrees with what is known (the given constraints), it maintains maximum uncertainty with respect to everything else. Had we chosen another distribution satisfying the constraints, we would have reduced the uncertainty and would have therefore implicitly made some extra restrictive assumption. For the problem at hand, we seek the distribution which maximizes the Shannon entropy while satisfying the expected distortion constraint. The corresponding Lagrangian to maximize is $H - \beta D$, with β the Lagrange multiplier. The equivalence of the two derivation is obvious—both Lagrangians are simultaneously optimized by the same solution configuration for $\beta = 1/T$.

To analyze further the Lagrangian F of (4) we note that the joint entropy can be decomposed into two terms: $H(X, Y) = H(X) + H(Y|X)$, where $H(X) = -\sum p(x) \log p(x)$ is the source entropy, which is independent of clustering. We may therefore drop the constant $H(X)$ from the Lagrangian definition, and focus on the conditional entropy

$$H(Y|X) = -\sum_x p(x) \sum_y p(y|x) \log p(y|x). \quad (5)$$

Minimizing F with respect to the association probabilities $p(y|x)$ is straightforward and gives the Gibbs distribution

$$p(y|x) = \frac{\exp\left(-\frac{d(x, y)}{T}\right)}{Z_x} \quad (6)$$

where the normalization is

$$Z_x = \sum_y \exp\left(-\frac{d(x, y)}{T}\right) \quad (7)$$

(which is the partition function of statistical physics). The corresponding minimum of F is obtained by plugging (6) back into (4)

$$F^* = \min_{\{p(y|x)\}} F = -T \sum_x p(x) \log Z_x = -T \sum_x p(x) \log \sum_y \exp\left(-\frac{d(x, y)}{T}\right). \quad (8)$$

To minimize the Lagrangian with respect to the codevector locations $\{y\}$, its gradients are set to zero yielding the condition

$$\sum_x p(x, y) \frac{d}{dy} d(x, y) = 0 \qquad \forall y \in Y. \quad (9)$$

Note that the derivative notation here stands, in general, for gradients. After normalization by $p(y) = \sum_x p(x, y)$ the condition can be rewritten as a centroid condition

$$\sum_x p(x|y) \frac{d}{dy} d(x, y) = 0 \quad (10)$$

(where $p(x|y)$ denotes the posterior probability calculated using Bayes's rule), which for the squared error distortion case takes the familiar form

$$y = \sum_x p(x|y) x. \quad (11)$$

While the above expressions convey most clearly the "centroid" aspect of the result, the practical approximation of the general condition (9) is

$$\frac{1}{N} \sum_x p(y|x) \frac{d}{dy} d(x, y) = 0 \qquad \forall y \in Y \quad (12)$$

where $p(y|x)$ is the Gibbs distribution of (6).

[2] The term "nearest" is used in the sense of the distortion measure $d(\cdot, \cdot)$, which is not necessarily the Euclidean distance.

The practical algorithm consists, therefore, of minimizing F^* with respect to the codevectors, starting at high value of T and tracking the minimum while lowering T. The central iteration consists of the following two steps:

1) fix the codevectors and use (6) to compute the association probabilities;
2) fix the associations and optimize the codevectors according to (12).

Clearly, the procedure is monotone nonincreasing in F^* and converges to a minimum. At high levels of T, the cost is very smooth and, under mild assumptions,[3] can be shown to be convex, which implies that the global minimum of F^* is found. As T tends to zero the association probabilities become hard and a hard clustering solution is obtained. In particular it is easy to see that the algorithm itself becomes the known GLA method [59] at this limit.

Some intuitive notion of the workings of the system can be obtained from observing the evolution of the association probabilities (6). At infinite T, these are uniform distributions, i.e., each input vector is equally associated with all clusters. These are extremely fuzzy associations. As T is lowered, the distributions become more discriminating and the associations less fuzzy. At zero temperature the classification is hard with each input sample assigned to the nearest codevector with probability one.[4] This is the condition in which traditional techniques such as GLA work. From the DA viewpoint, standard methods are "zero temperature" methods. It is easy to visualize how the zero temperature system cannot "sense" a better optimum farther away, as each data point exercises its influence only on the nearest codevector. On the other hand, by starting at high T and slowly "cooling" the system, we start with each data point equally influencing all codevectors and gradually localize the influence. This gives us some intuition as to how the system senses, and settles into, a better optimum.

Another important aspect of the algorithm is seen if we view the association probability $p(y|x)$ as the expected value of the random binary variable $V_{xy} \in \{0, 1\}$ which take the value one if input x is assigned to codevector y, and zero if not. From this perspective one may recognize the known expectation maximization (EM) algorithm [21] in the above two step iteration. The first step, which computes the association probabilities, is the "expectation" step, and the second step which minimizes F^* is the "maximization" (of $-F^*$) step. Note further that the EM algorithm is applied here at each given level of T. The emergence of EM is not surprising given that for many choices of distortion measure, F^* can be given an interpretation as a negative likelihood function. For example, in the case of squared error distortion, the optimization of F^* is equivalent to maximum likelihood estimation of means in a normal mixture, where the assumed variance is determined by T. It is important, however, to note that in general we do not necessarily assume an underlying probabilistic model for the data. Our distributions are derived from the distortion measure. In compression applications, in particular, the distortion measure attempts to quantify the perceptual significance of reconstruction error, independent of the source statistics.

2) Statistical Physics Analogy: The above probabilistic derivation is largely motivated by analogies to statistical physics. In this section, we develop this analogy and indicate more precisely how the method produces an annealing process. Moreover, we will demonstrate that the system undergoes a sequence of "phase transitions," and thereby we will obtain further insights into the process.

Consider a physical system whose energy is our distortion D and whose Shannon entropy is H. The Lagrangian, $F = D - TH$, which is central to the DA derivation, is exactly the Helmholtz free energy of this system (strictly speaking it is the Helmholtz thermodynamic potential). The Lagrange multiplier T is accordingly the temperature of the system which governs its level of randomness. Note that our choice of notation (with the exception of D, which stands for distortion) was made to agree with the traditional notation of statistical mechanics, and it emphasizes this direct analogy. A fundamental principle of statistical mechanics (often called the principle of minimal free energy) states that the minimum of the free energy determines the distribution at thermal equilibrium. Thus, F^* is achieved by the system when it reaches equilibrium, at which point the system is governed by the Gibbs (or canonical) distribution. The chemical procedure of annealing consists of maintaining the system at thermal equilibrium while carefully lowering the temperature. Compare this with the computational procedure of DA: track the minimum of the free energy while gradually lowering the temperature! In chemistry, annealing is used to ensure that the ground state of the system, that is, the state of minimum energy, is achieved at the limit of low temperature. The method of simulated annealing [54] directly simulates the stochastic evolution of such a physical system. We, instead, derive its free energy as the corresponding expectation, and deterministically (and quickly) optimize it to characterize the equilibrium at the given temperature.

In summary, the DA method performs annealing as it maintains the free energy at its minimum (thermal equilibrium) while gradually lowering the temperature; and it is deterministic because it minimizes the free energy directly rather than via stochastic simulation of the system dynamics.

But there is much more to the physical analogy. We shall next demonstrate that, as the temperature is lowered, the system undergoes a sequence of "phase transitions," which consists of natural cluster splits where the clustering model grows in size (number of clusters). This phenomenon is highly significant for a number of reasons. First, it provides a useful tool for controlling the size of the clustering model and relating it to the scale of the solution, as will be explained below. Second, these phase transitions are the critical points of the process where one needs to be careful

[3] If $d(x, y)$ is a differentiable, convex function of y for all x, then F^* has a unique minimum, asymptotically, at high temperature T.

[4] More precisely, each input sample is uniformly associated with the set of equidistant nearest representatives. We will ignore the pathologies of encoding "ties" as they are of no significance in DA.

with the annealing (as is the case in physical annealing). The "critical temperatures" are computable, as will be shown next. This information allows us to accelerate the procedure in between phase transitions without compromising performance. Finally, the sequence of solutions at various phases, which are solutions of increasing model size, can be coupled with validation procedures to identify the optimal model size for performance outside the training set.

Let us begin by considering the case of very high temperature ($T \to \infty$). The association probabilities (6) are uniform, and the optimality condition (12) is satisfied by placing all codevectors at the same point—the centroid of the training set determined by

$$\frac{1}{N} \sum_x \frac{d}{dy} d(x, y) = 0. \tag{13}$$

(In the case of squared error distortion this optimal y is the sample mean of the training set.) Hence, at high temperature, the codebook Y collapses on a single point. We say, therefore, that there is effectively one codevector and one cluster—the entire training set. As we lower the temperature the cardinality of the codebook changes. We consider the effective codebook cardinality, or model size, as characterizing the phases of the physical system. The system undergoes phase transitions as the model size grows. An analysis of the phase transitions is fundamental in obtaining an understanding of the evolution of the system.

In order to explicitly derive the "critical temperatures" for the phase transitions, we will assume the squared error distortion $d(x, y) = |x - y|^2$. The bifurcation occurs when a set of coincident codevectors splits into separate subsets. Mathematically, the existing solution above the critical temperature is no longer the minimum of the free energy as the temperature crosses the critical value. Although it is natural to define this as the point at which the Hessian of F^* loses its positive definite property, the notational complexity of working with this large and complex matrix motivates the equivalent approach of variational calculus. Let us denote by $Y+\epsilon\Psi = \{y+\epsilon\psi_y\}$ a perturbed codebook, where ψ_y is the perturbation vector applied to codevector y, and where the nonnegative scalar ϵ is used to scale the magnitude of the perturbation. We can rewrite the necessary condition for optimality of Y

$$\frac{d}{d\epsilon} F^*(Y + \epsilon\Psi)|_{\epsilon=0} = 0 \tag{14}$$

for all choices of finite perturbation Ψ. This variational statement of the optimality condition leads directly to the earlier condition of (9). But we must also require a condition on the second-order derivative

$$\frac{d^2}{d\epsilon^2} F^*(Y + \epsilon\Psi)|_{\epsilon=0} \geq 0 \tag{15}$$

for all choices of finite perturbation Ψ. Bifurcation occurs when equality is achieved in (15) and hence the minimum is no longer stable.[5] Applying straightforward differentiation we obtain the following condition for equality in (15)

$$\sum_x \sum_y p(x, y)\psi_y^t \left[I - \frac{2}{T}(x-y)(x-y)^t \right] \psi_y + \sum_x p(x) \left[\sum_y p(y|x)\psi_y^t (x-y) \right]^2 = 0 \tag{16}$$

where I denotes the identity matrix. The first term can be rewritten more compactly and the equation becomes

$$\sum_y p(y)\psi_y^t \left[I - \frac{2}{T} C_{x|y} \right] \psi_y + \sum_x p(x) \left[\sum_y p(y|x)(x-y)^t \psi_y \right]^2 = 0 \tag{17}$$

where

$$C_{x|y} = \sum_x p(x|y)(x-y)(x-y)^t \tag{18}$$

is the covariance matrix of the posterior distribution $p(x|y)$ of the cluster corresponding to codevector y. We claim that the left-hand side of (17) is positive for all perturbations Ψ if and only if the first term is positive. The "if" part is trivial since the second term is obviously nonnegative. To show the "only if" part, we first observe that the first term is nonpositive only if there exists some $y_0 \in Y$ with positive probability such that the matrix $I - (2/T)C_{x|y_0}$ is not positive definite. In fact, we assume that there are several coincident codevectors at this point to allow bifurcation. We next show that in this case there always exists a perturbation that makes the second term vanish. Select a perturbation Ψ satisfying

$$\psi_y = 0, \qquad \forall y \neq y_0 \tag{19}$$

and

$$\sum_{y \in Y:\, y = y_0} \psi_y = 0. \tag{20}$$

With this perturbation the second term becomes

$$\sum_x p(x) \left[p(y_0|x)(x-y_0)^t \sum_{y \in Y:\, y = y_0} \psi_y \right]^2$$

which equals zero by (20). Thus, whenever the first term is nonpositive we can construct a perturbation such that the second term vanishes. Hence, there is strict inequality in (15) if and only if the first term of (17) is positive for all choices of finite perturbation Ψ.

In conclusion to the above derivation, the condition for phase transition requires that the coincident codevectors

[5] For simplicity we ignore higher order derivatives, which should be checked for mathematical completeness, but which are of minimal practical importance. The result is a necessary condition for bifurcation.

at some y_0 have a (posterior) data distribution $p(x|y)$ satisfying

$$\det\left[I - \frac{2}{T} C_{x|y_0}\right] = 0. \tag{21}$$

The critical temperature T_c is therefore determined as

$$T_c = 2\lambda_{\max} \tag{22}$$

where $\lambda_{\max}$ is the largest eigenvalue of $C_{x|y_0}$. In other words, phase transitions occur as the temperature is lowered to twice the variance along the principal axis of the cluster. It can be further shown that the split (separation of codevectors) is along this principal axis. We summarize this result in the form of a theorem.

Theorem 1: For the squared error distortion measure, a cluster centered at codevector y undergoes a splitting phase transition when the temperature reaches the critical value $T_c = 2\lambda_{\max}$, where $\lambda_{\max}$ is the cluster's principal component.

Fig. 1 illustrates the annealing process with its phase transitions on a simple example. The training set is generated from a mixture of six randomly displaced, equal variance Gaussians whose centers are marked by X. At high temperature, there is only one effective cluster represented by one codevector, marked by O, at the center of mass of the training set. As the temperature is lowered, the system undergoes phase transitions which increase the number of effective clusters as shown in the figure. Note that as the partition is random, there are no precise boundaries. Instead, we give "isoprobability curves," or contours of equal probability (typically 1/3) of belonging to the cluster. In Fig. 2 we give the corresponding "phase diagram," which describes the variation of average distortion (energy) with $\beta = 1/T$. Note, in particular, that when the temperature reaches the value which corresponds to the variance of all the isotropic Gaussians we get an "explosion" as there is no preferred direction (principal component) for the split. More on the analysis of phase transitions is given in a later section on RD theory, but for a deeper treatment of the condition for explosion, or continuum of codevectors, and the special role of Gaussian distribution, see [87].

3) Mass-Constrained Clustering: The mass constrained clustering approach [91] is the preferred implementation of the DA clustering algorithm, and a detailed sketch of the algorithm will be given at the end of this section. We shall show that the annealing process, as described so far, has a certain dependence on the number of coincident codevectors in each effective cluster. This weakness is not desirable and can be eliminated, leading to a method that is totally independent of initialization.

We start by recalling a central characteristic of the DA approach: no matter how many codevectors are "thrown in," the effective number emerges at each temperature. This number is the model size, and it defines the phase of the system. For example, even if we have thousands of codevectors, there is only one single effective codevector at very high temperature. However, after a split occurs, the result

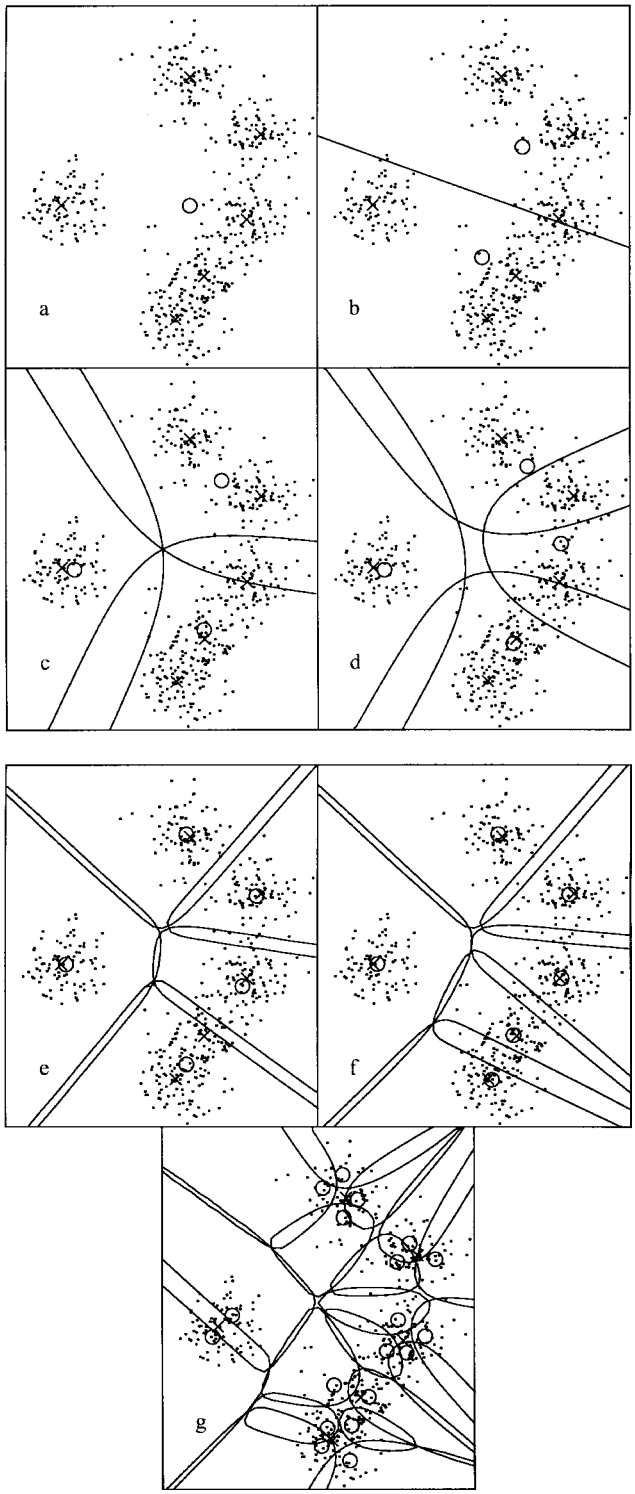


Fig. 1. Clustering at various phases. The lines are equiprobable contours, $p = 1/2$ in (b), and $p = 1/3$ elsewhere. (a) one cluster ($\beta = 0$), (b) two clusters ($\beta = 0.0049$), (c) three clusters ($\beta = 0.0056$), (d) four clusters ($\beta = 0.0100$), (e) five clusters ($\beta = 0.0156$), (f) six clusters ($\beta = 0.0347$), and (g) 19 clusters ($\beta = 0.0605$). From [90].

may differ somewhat depending on the number of codevectors in each of the resulting subgroups. Clearly, the initial partition into subgroups depends on the perturbation. In order to fix this shortcoming, let us reformulate the method

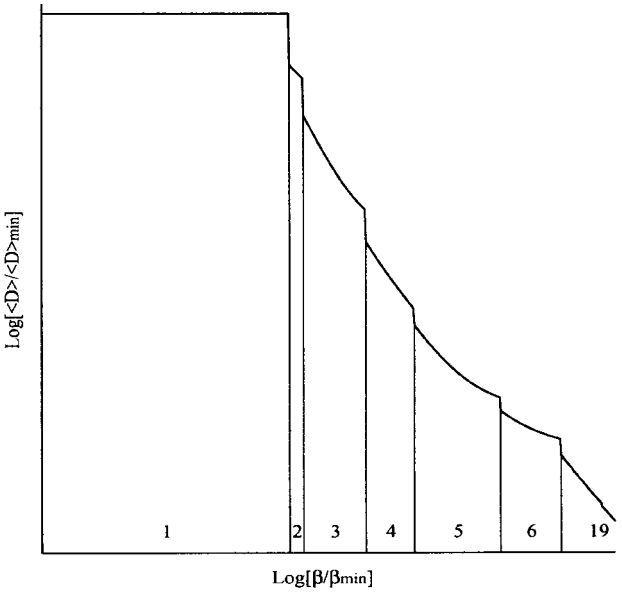


Fig. 2. Phase diagram for the distribution shown in Fig. 1. The number of effective clusters is shown for each phase. From [90].

in terms of effective clusters (or distinct codevectors). Let us assume that there is an unlimited supply of codevectors, and let p_i denote the fraction of codevectors which represent effective cluster i and are therefore coincident at position y_i. Using this notation, the partition function of (7) is rewritten equivalently as

$$Z_x = \sum_i p_i e^{-(d(x, y_i)/T)} \tag{23}$$

where the summation is over distinct codevectors. The probability of association (6) with distinct codevector y_j is the so-called tilted distribution

$$p(y_j|x) = \frac{p_j e^{-(d(x, y_j)/T)}}{Z_x} \tag{24}$$

and the free energy (8) is

$$\begin{aligned} F^* &= -T \sum_x p(x) \log Z_x \\ &= -T \sum_x p(x) \log \sum_i p_i e^{-(d(x, y_i)/T)}. \end{aligned} \tag{25}$$

The free energy is to be minimized under the obvious constraint that $\sum_i p_i = 1$. The optimization is performed as unconstrained minimization of the Lagrangian

$$F' = F^* + \lambda \left(\sum_i p_i - 1 \right) \tag{26}$$

with respect to the cluster parameters y_i and p_i. Note that although we started with a countable number of codevectors to be distributed among the clusters, we effectively view them now as possibly uncountable, and p_i is not required to be rational. One may therefore visualize this as a "mass of codevectors" which is divided among the effective clusters, or simply as a distribution over the codevector space. (The notion of inducing a possibly continuous distribution over the codevector space provides a direct link to rate-distortion theory, which is pursued in a later section.)

The optimal set of codevectors $\{y_j\}$ must satisfy

$$\frac{\partial}{\partial y_j} F' = \frac{\partial}{\partial y_j} F^* = 0 \tag{27}$$

where the left equality is because the constraint is independent of the positions $\{y_j\}$. We thus get again the condition of (9)

$$\sum_x p(x) p(y_j|x) \frac{\partial}{\partial y_j} d(x, y_j) = 0 \tag{28}$$

with the important distinction that now the association probabilities are tilted according to (24).

On the other hand, the set $\{p_i\}$ which minimizes F' satisfies

$$\frac{\partial}{\partial p_i} F' = -T \sum_x p(x) \frac{e^{-(d(x, y_i)/T)}}{Z_x} + \lambda = 0 \tag{29}$$

which yields

$$\frac{\lambda}{T} = \sum_x p(x) \frac{e^{-(d(x, y_i)/T)}}{Z_x}. \tag{30}$$

Taking the expectation of (30) with respect to the distribution p_i we obtain

$$\frac{\lambda}{T} = \sum_i p_i \sum_x p(x) \frac{e^{-(d(x, y_i)/T)}}{Z_x} = 1 \tag{31}$$

where the last equality uses the definition of Z_x in (23). Thus

$$\lambda = T. \tag{32}$$

Substituting (32) in (30) we see that the optimal distribution p_i must satisfy

$$\sum_x p(x) \frac{e^{-(d(x, y_i)/T)}}{Z_x} = 1 \tag{33}$$

where $\{p_i\}$ are implicit in Z_x (23). Equation (33) is thus the equation we solve while optimizing over $\{p_i\}$. This equation also arises from the Kuhn–Tucker conditions of rate-distortion theory [7], [12], [40].

It is instructive to point out that (24) and (33) imply that

$$\begin{aligned} p_i &= \sum_x p(x) \frac{p_i e^{-(d(x, y_i)/T)}}{Z_x} \\ &= \sum_x p(x) p(y_i|x) = p(y_i). \end{aligned} \tag{34}$$

In other words, the optimal codevector distribution mimics the training data set partition into the clusters. The distribution p_i is identical to the probability distribution induced on the codevectors via the encoding rule which we have denoted by $p(y) = \sum_x p(x) p(y|x)$. As an aside, note that the results are also in perfect agreement with estimation of priors (mixing coefficients) in parametric estimation of mixture of densities. However, it must be kept in mind that the mass-constrained algorithm is applicable to solving the

simple VQ or clustering design problem where minimum distortion is the ultimate objective.

The mass-constrained formulation only needs as many codevectors as there are effective clusters at a given temperature. The process is computationally efficient and increases the model size only when it is needed, i.e., when a critical temperature has been reached. The mechanism can be implemented by maintaining and perturbing pairs of codevectors at each effective cluster so that they separate only when a phase transition occurs. Another possibility is to compute the critical temperature and supply an additional codevector only when the condition is satisfied.

It should also be noted that at the limit of low temperature ($T = 0$), both the unconstrained DA method and the mass-constrained DA method, converge to the same descent process, namely, GLA [59] (or basic ISODATA [4] for the sum of squared distances). This is because the association probabilities of the two DA methods are identical at the limit, and they assign each data point to the nearest codevector with probability one. The difference between the two is in their behavior at intermediate T, where the mass-constrained clustering method takes the cluster populations into account. This is illustrated by a simple example in Fig. 3.

4) Preferred Implementation of the DA Clustering Algorithm: We conclude the treatment of the basic clustering problem with a sketch of a preferred implementation of the clustering algorithm. This version incorporates the mass-constrained approach. The squared error distortion is assumed for simplicity, but the description is extendible to other distortion measures. It is also assumed that the objective is to find the hard (nonfuzzy) clustering solution for a given number of clusters.

1) *Set Limits*: number of codevectors $K_{\max}$, minimum temperature $T_{\min}$.
2) *Initialize*: $T > 2\lambda_{\max}(C_x)$, $K = 1$, $y_1 = \sum_x xp(x)$, and $p(y_1) = 1$.
3) *Update for* $i = 1, \ldots, K$

$$y_i = \frac{\sum_x xp(x)p(y_i|x)}{p(y_i)}$$

where

$$p(y_i|x) = \frac{p(y_i)e^{-((x-y_i)^2/T)}}{\sum_{j=1}^{K} p(y_j)e^{-((x-y_j)^2/T)}}$$

$$p(y_i) = \sum_x p(x)p(y_i|x).$$

4) *Convergence Test*: If not satisfied go to 3).
5) If $T \leq T_{\min}$, perform last iteration for $T = 0$ and STOP.
6) *Cooling Step*: $T \leftarrow \alpha T$, ($\alpha < 1$).
7) If $K < K_{\max}$, check condition for phase transition for $i = 1, \ldots, K$. If critical T is reached for cluster j, add a new codevector $y_{K+1} = y_j + \delta$, $p(y_{K+1}) = p(y_j)/2$, $p(y_j) \leftarrow p(y_j)/2$ and increment K.
8) Go to 3).

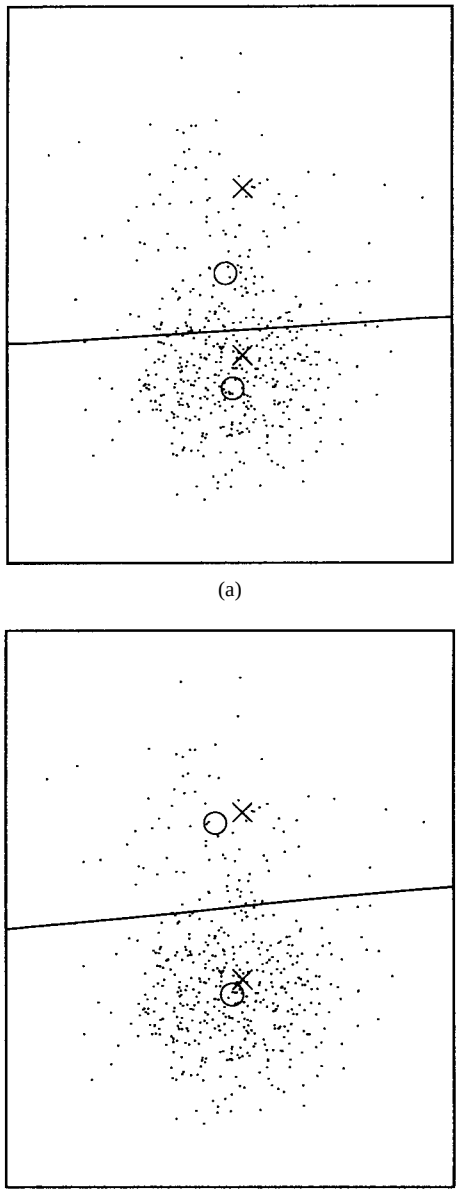


Fig. 3. The effect of cluster mass (population) at intermediate β. The data is sampled from two normal distributions whose centers are marked by X. The computed representatives are marked by O. (a) Nonconstrained clustering and (b) mass-constrained clustering. From [91].

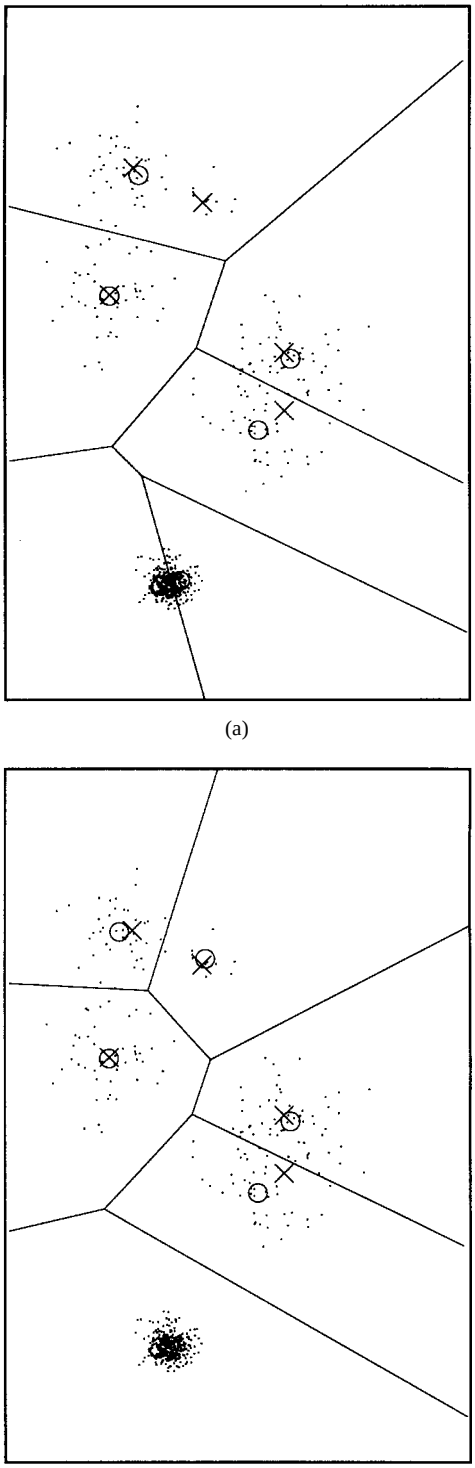
Note that the test for critical T in 7) may be replaced by a simple perturbation if considered expensive for high dimensions. In this case we always keep two codevectors at each location and perturb them when we update T. Until the critical T is reached they will be merged together by the iterations. At phase transition they will move further apart.

The relation to the Lloyd algorithm for quantizer design is easy to see. At a given T, the iteration is a generalization of the nearest neighbor and the centroid conditions. The relation to maximum likelihood estimation of parameters

in normal mixtures is also obvious. For a treatment of the problem of covariance matrix estimation in DA see [55].

While the algorithm sketch was given for the typical case of VQ design, it is easy to modify it to produce cluster analysis solutions. In particular, fuzzy clustering solutions are produced naturally at a sequence of scales (as determined by the temperature). One simple approach is to combine the algorithm with cluster validation techniques to select a scale (and solution) from this sequence. It is also easy to produce hard clustering solutions at the different scales by adding quick "quenching" at each phase to produce the required hard solutions which can then be processed for validation.

5) Illustrative Examples: To further illustrate the performance of mass-constrained clustering, we consider the example shown in Fig. 4. This is a mixture of six Gaussian densities of different masses and variances. We compare the result of DA with the well known GLA method. Since GLA yields results that depend on the initialization, we have run it 25 times, each time with a different initial set of representatives (randomly extracted from the training set). In Fig. 4(a), we show the best result obtained by GLA, where the MSE is 6.4. This result was obtained only once, while for ≈80% of the runs it got trapped in local optima of ≈12.5 MSE. In Fig. 4(b), we show the result obtained by DA. The MSE is 5.7, and this solution is, of course, independent of the initialization. The process of "annealing" is illustrated in Fig. 5. Here we have a mixture of nine overlapping Gaussian densities. The process undergoes a sequence of phase transitions as $\beta = 1/T$ is increased. We show the results at some of the phases. Equiprobable contours are used to emphasize the fuzzy nature of the results at intermediate β. At the limit of high β, the MSE is 32.854. Repeated runs of GLA on this example yielded a variety of local optima with MSE from 33.5 to 40.3.



(a)

(b)

Fig. 4. GLA versus DA: (a) best result of GLA out of 25 runs with random initialization: $D = 6.4$ and (b) mass-constrained clustering: $D = 5.7$. From [91].

B. Extensions and Applications

In this section we consider several direct extensions of the DA clustering method. First, we consider extensions motivated by compression and communications applications including VQ design for transmission through noisy channels, entropy-constrained vector quantization, and structurally constrained clustering, which addresses the encoding and storage complexity problem. Finally, we briefly discuss straightforward extensions via constraints on the codevectors and identify as special cases approaches to the "traveling salesman problem" and self-organizing feature maps.

1) Vector Quantization for Noisy Channels: The area of source-channel coding is concerned with the joint design of communication systems while taking into account the distortion due to both compression and transmission over a noisy channel. In the particular case of VQ-based communications systems, it is advantageous to optimize the quantizer while taking into account the effects of the channel. A noisy channel is specified here by its transition probabilities $p(z|y)$, which denote the probability that the decoder decides on codevector z given that the encoder transmitted the index of codevector y.[6] (As an aside, it may be noted that there exist applications [61], [62], where such noise models are used to model the distortion due

[6]The implicit simplifying assumption is that the channel is memoryless or at least does not have significant temporal dependencies reaching beyond the transmission of a codevector index.

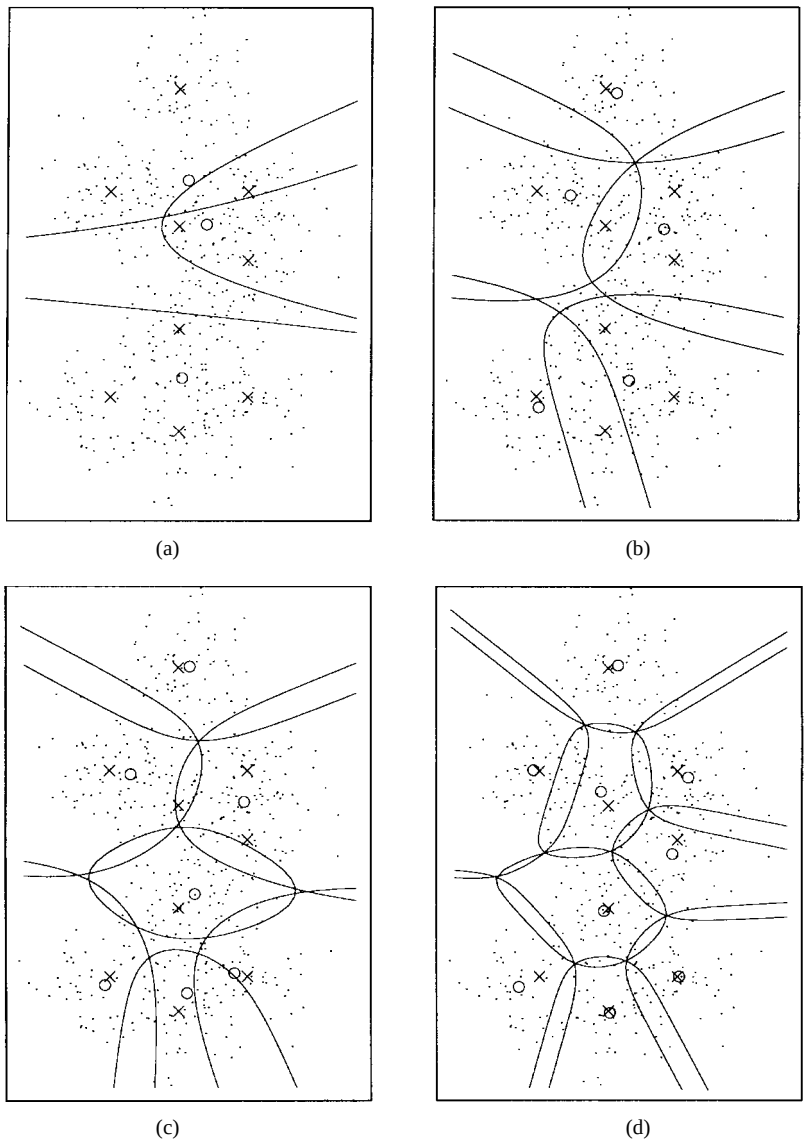


Fig. 5. Various phases in the annealing process. Random partition represented by equiprobable contours (p) to emphasize the fuzzy nature of the results at intermediate β: (a) three clusters at $\beta = 0.005$ contours at $p = 0.45$, (b) five clusters at $\beta = 0.009$, $p = 0.33$, (c) seven clusters at $\beta = 0.013$, $p = 0.33$, and (d) nine clusters at $\beta = 0.03$, $p = 0.33$. From [91].

to hierarchical or topological constraints rather than a real communication channel.)

A simple but important observation is the following: the noisy-channel VQ design problem is in fact identical to the regular VQ design, but with the modified distortion measure

$$d'(x, y) = \sum_z p(z|y)\, d(x, z) \tag{35}$$

which measures the expected distortion when codevector y is selected by the encoder. This observation allows direct extension of the known VQ design algorithms to this case. There is a long history of noisy-channel quantizer design. In the 1960's, a basic method was proposed for scalar quantizers [58] and was extended in many papers since [3], [24], [28], [30], [57], [109]. These papers basically describe GLA-type methods which alternate between enforcing the encoder and centroid (decoder) optimality conditions. One can similarly extend the DA approach to the noisy channel case [14], [72].

We can write the expected overall source-channel distortion as

$$\begin{aligned} D &= \sum_x \sum_y p(x, y)\, d'(x, y) \\ &= \sum_x \sum_y \sum_z p(x)p(y|x)p(z|y)\, d(x, z) \end{aligned} \tag{36}$$

where $p(y|x)$ defines the encoder, which is random during the DA design. Note that we exploit the fact that (X, Y, Z) form a Markov chain. Alternatively, we may write that

$$D = \sum_x \sum_z p(x)p(z|x)\, d(x, z) \tag{37}$$

where $p(z|x) = \sum_y p(y|x)p(z|y)$. The entropy is defined over the encoding probabilities (the probabilities which are

under our control)

$$H = -\sum_x \sum_y p(x, y) \log p(y|x). \tag{38}$$

Following the standard DA derivation we obtain the optimal encoding probability at a given temperature T

$$p(y|x) = \frac{\exp\left(-\frac{d'(x, y)}{T}\right)}{Z_x} \tag{39}$$

and the free energy

$$F^* = -T \sum_x p(x) \log \sum_y \exp\left(-\frac{d'(x, y)}{T}\right). \tag{40}$$

Optimizing F^* with respect to the parameters yields the centroid rule

$$\sum_x p(x)p(z|x) \frac{d}{dz} d(x, z) = 0 \tag{41}$$

which simplifies for the squared error distortion to

$$z = \sum_x p(x|z)x. \tag{42}$$

For a phase transition analysis of the DA process in the noisy-channel VQ case see [38]. In [72] it is shown that the DA approach avoids many local optima of the design and outperforms standard design. Of particular interest is how the design process converges to explicit error control coding (ECC) at the limit of very noisy channels. At this limit, certain encoding regions become empty, so that less codevectors are used by the encoder. Some of the available bit rate is thus reserved for channel protection. Note that the system itself finds the optimal error correcting code which need not (and does not, in general) satisfy any algebraic constraints such as linearity, as typically required in ECC design.

2) Entropy-Constrained VQ Design: Variable-length coding (also called entropy coding) is commonly used to further reduce the rate required to transmit quantized signals. It results in rates that are close to the quantized signal entropy, which is the fundamental limit. The basic VQ design approach, however, assumes fixed-length coding, thereby simplifying the optimization problem to that of minimizing distortion for the given codebook size. It is, however, of much interest to derive a method for optimizing the VQ for use in conjunction with variable-length coding. Here the optimization must take into account both the distortion and the rate costs of selecting a codevector. This fundamental extension of the VQ problem was obtained by incorporation of an entropy constraint within the design to produce quantizers optimized for subsequent entropy coding. The earlier work was concerned with scalar quantizers [8], [29]. The VQ design method was proposed by Chou *et al.* [19]. We refer to this paradigm as the entropy-constrained VQ (ECVQ).

The cost function is the weighted cost

$$D + \lambda H = \sum_x \sum_y p(x, y)[d(x, y) - \lambda \log p(y)] \tag{43}$$

where λ determines the penalty for increase in rate relative to increase in distortion. It can also be used as a Lagrange multiplier for imposing a prescribed rate constraint while minimizing the distortion (or, alternatively, imposing a prescribed distortion while minimizing the rate). It follows that this problem reverts to the regular VQ problem with a modified cost function

$$d'(x, y) = d(x, y) - \lambda \log p(y) \tag{44}$$

subject to the additional constraint $\sum_y p(y) = 1$. This observation leads directly to the ECVQ algorithm of Chou *et al.* [19].

It can also be incorporated in a DA method for ECVQ design, as was first pointed out by Buhmann and Kuhnel [14]. The free energy is

$$F^* = -T \sum_x p(x) \log \sum_y \exp\left(-\frac{d'(x, y)}{T}\right). \tag{45}$$

Note that in ECVQ we do not use a mass-constrained version of DA since the masses are already implicit in the modified distortion measure (44). Moreover, the above becomes equivalent to mass-constrained operation in the special case $\lambda = T$.

The resulting update rules are derived from the free energy. The following assumes, for simplicity, the squared error distortion measure. The encoding rule is

$$p(y|x) = \frac{\exp\left(-\frac{d(x, y) - \lambda \log p(y)}{T}\right)}{\sum_y \exp\left(-\frac{d(x, y) - \lambda \log p(y)}{T}\right)}. \tag{46}$$

The centroid rule is

$$y = \sum_x p(x|y)x \tag{47}$$

where $p(x|y) = p(x)p(y|x)/p(y)$, and the mass rule is

$$p(y) = \sum_x p(x)p(y|x). \tag{48}$$

At $T = 0$ the DA iteration becomes identical to the standard ECVQ algorithm of [19].

3) Structurally Constrained VQ Design: A major stumbling block in the way of VQ applications is the problem of encoding complexity. The size of the codebook grows exponentially with the vector dimension and rate (in bits per sample). As the encoder has to find the best codevector in the codebook, its complexity grows linearly with the codebook size, and hence exponentially with dimension and rate. In many practical applications, such encoding complexity is not acceptable and low-complexity alternatives are needed. The most common approach for reducing encoding complexity involves the imposition of a structural constraint on the VQ partition. A tree-structured partition is a typical such structure consisting of nested decision boundaries which can be represented by a decision tree. Sibling nodes in the tree define a VQ that partitions the region associated with their parent node. The

reproduction codebook of the tree-structured VQ (TSVQ) is, in fact, the set of leaves. The role of the internal nodes is to provide a mechanism for fast encoding search. The encoding operation is not an exhaustive search through the leaves. Instead, one starts at the root of the tree and determines a path to a leaf by a sequence of local decisions. At each layer the decision is restricted to selecting one of the descendants of the winning node in the previous layer. Thus, the encoding search grows linearly, rather than exponentially, with the dimension and the rate.

The design of TSVQ is, in general, a harder optimization problem than the design of regular VQ. Typical approaches [15], [44], [84] employ a greedy sequential design, optimizing a local cost to grow the tree one node (or layer) at a time. The reason for the greedy nature of standard approaches is that, whereas in the unstructured case an optimal partition design step is readily specified by the nearest neighbor rule, in the tree structured case an optimal partition is determined only by solving a formidable multiclass risk discrimination problem [23]. Thus, the heuristically determined high-level boundaries may severely constrain the final partition at the leaf layer, yet they are not readjusted when lower layers are being designed.

The DA approach to clustering offers a way to optimize the partition (i.e., all the parameters which define the final partition) directly and, moreover, to escape many shallow local minima traps. It should be noted that the new difficulty here is the need to impose the structure on the partition. Earlier work on this problem [73] appealed to the principle of minimum cross-entropy (or minimum divergence) which is a known generalization of the principle of maximum entropy [97]. Minimum cross-entropy provides a probablistic tool to gradually enforce the desired consistency between the leaf layer, where the quantization cost is calculated, and the rest of the tree—thereby imposing the structural constraint on the partition at the limit of zero temperature. This approach provided consistent substantial gains over the standard approaches. Although this method worked very well in all tests, it has two theoretical disadvantages. First, alternate minimization of the cross-entropy and of the cost at the leaf layer is not ensured to converge, though in practice this has not been a problem. The second undesired aspect is that it lacks the direct simplicity of basic DA. More recent developments of the DA approach in the context of supervised learning [69], [70] have since opened the way for a simpler, more general way to impose a structure on the partition, and which is also a more direct extension of the basic DA approach to clustering. A detailed description of this extension will be given in the section on supervised learning. Here we shall only cover the minimum required to develop the DA approach for TSVQ design. It is appropriate to focus on the latter derivation since it has none of the theoretical flaws of the earlier approach. However, no simulation results for it exist as of the time this paper was written. To illustrate the annealing and the type of gains achievable we will include some simulation results of the earlier approach [73] which, in spite of its theoretical shortcomings, approximates closely the optimal annealing process and achieves (apparently) globally optimal solutions for these nontrivial examples.

We replace the hard encoding rule with a randomized decision. The probability of encoding input x with a leaf (codevector) y is in fact the probability of choosing the entire path starting at the root of the tree and leading to this leaf. This is a sequence of decisions where each node on the path competes with its siblings. The probability of choosing node s given that its parent was chosen is Gibbs[7]

$$p(s|x,\ \text{parent(s)}) = \frac{\exp\left(-\gamma\, d(x,\ s)\right)}{\sum_{s' \in \text{sibling(s)}} \exp\left(-\gamma\, d(x,\ s')\right)} \tag{49}$$

where γ is a scale parameter. Thus, the selection of nodes at sequential layers is viewed as a Markov chain, where the transition probabilities obey the Gibbs distribution. Note in particular that as $\gamma \to \infty$, this Markov chain becomes a hard decision tree, and the resulting partition corresponds to a standard TSVQ. We have thus defined a randomized tree partition which, at the limit, enforces the desired structure on the solution.

We next wish to minimize the objective which is the overall distortion at a specified level of randomness. We again define the Lagrangian (the Helmholtz free energy)

$$F = D - TH \tag{50}$$

where D is the distortion at the leaf layer and H is the entropy of the Markov chain, both computed while employing the explicit Gibbs form of (49). Then, by minimizing the free energy over the tree parameters $\{s,\ y,\ \gamma\}$ we obtain the optimal random tree at this temperature. As the temperature is lowered, reduction in entropy is traded for reduction in distortion, and at $T = 0$ the tree becomes hard. This process also involves a sequence of phase transitions as the tree grows, similar to the case of unconstrained VQ design. In Fig. 6 we show the performance of the DA method from [73] on a mixture example, as well as the sequence of phase transitions and the manner in which the tree grows. In Fig. 7 we show how TSVQ designed by DA outperforms the unstructured VQ designed by standard methods. This demonstrates the significant impact of poor local optima which cause worse degradation than the structural constraint itself.

Beside the tree structure, on which we focused here, there are other important structures that are used in signal compression and for which DA design methods have been developed. One commonly used structure, particularly in speech coding, is the multistage vector quantizer (see [71] for an early DA approach). Another very important structure is the trellis quantizer (as well as the trellis vector quantizer) for which a DA approach has been proposed in [74].

[7]The choice of the Gibbs distribution is not arbitrary and will be explained in a fundamental and general setting in Section III. At this point let it simply be noted that it is directly obtainable from the maximum entropy principle.

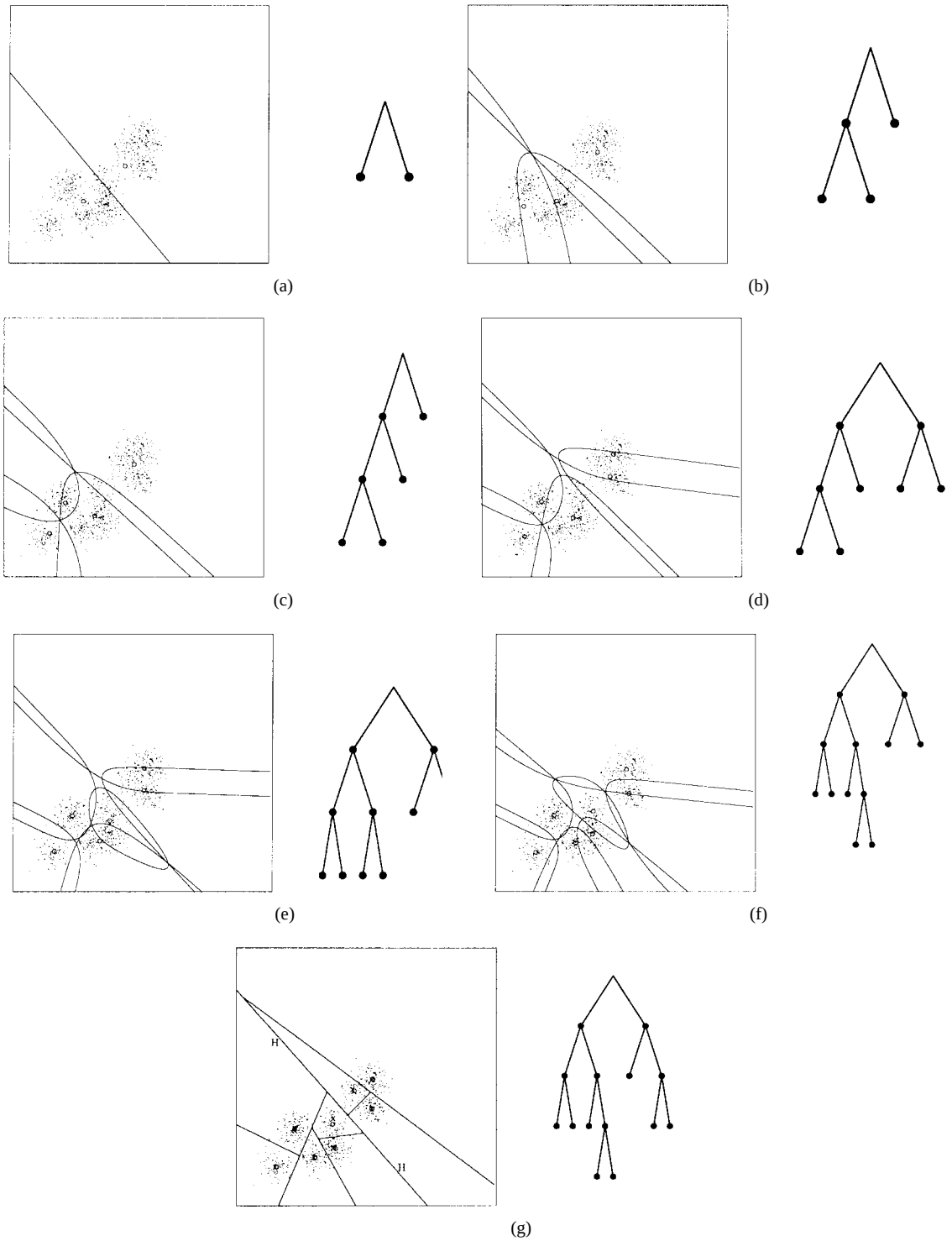


Fig. 6. A hierarchy of tree-structured solutions generated by the annealing method for increasing β. The source is a Gaussian mixture with eight components. To the right of each figure is the associated tree structure. The lines in the figure are equiprobable contours with membership probability of $p = 0.33$ in a given partition region, except for (a) and (g), for which $p = 0.5$. "H" denotes the highest level decision boundary in (g). From [73].

4) Graph-Theoretic and Other Optimization Problems: In the deterministic annealing clustering algorithm, if we throw in enough codevectors and let $T \rightarrow 0$, then each data point will become a natural cluster. This can be viewed as a process of data association, where each data point is exclusively associated with a "codevector." As it stands, there is no preference as to which codevector is associated with which data point. However, by adding appropriate constraints, which are easy to incorporate in the Lagrangian derivation, we can encourage the process to obtain associations that satisfy additional requirements which embody the actual data assignment problem we

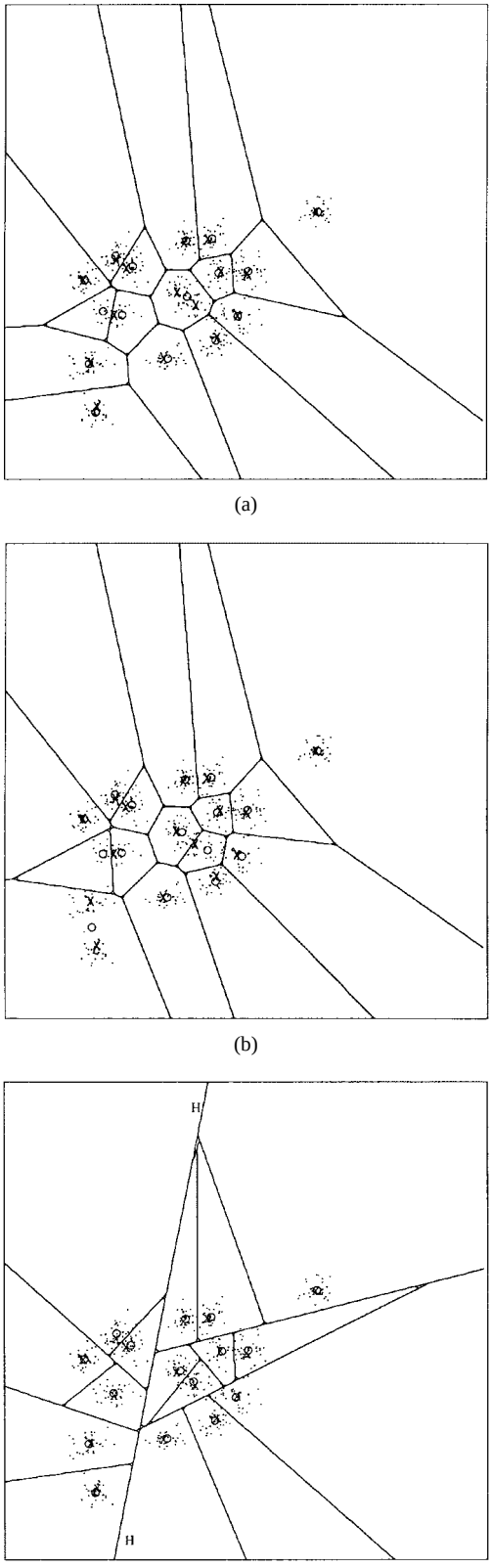


Fig. 7. A Gaussian mixture example with 16 components. (a) The best unconstrained GLA solution after 30 random initializations within the data set, with $D = 0.51$. (b) A typical unconstrained GLA solution, with $D = 0.72$. (c) The unbalanced tree-structured DA solution with maximal depth of five and $D = 0.49$. From [73].

wish to solve. This allows exploiting DA as a framework for solving a variety of hard graph-theoretic problems. As an example, when applied to the famous "traveling salesman" problem, such DA derivation becomes identical to the "elastic net" method [26], [27]. The approach has been applied to various data assignment problems such as the module placement problem in computer-aided design (CAD) and graph partitioning. Another variant with a different constraint yields a DA approach for batch optimization of self-organizing feature maps. For more details see [91] and [92].

III. Deterministic Annealing for Supervised Learning

In this section, we develop the deterministic annealing approach for problems of supervised learning. We consider first the general supervised learning or function approximation problem where, from a given parametric class of functions, a function is selected which best captures the input–output statistics of a training set. In the learning literature, the given parametric class of functions is viewed as the set of transfer functions implementable on a given system structure by varying its parameters (e.g., the weights of a multilayer perceptron). The ultimate performance evaluation is commonly performed on an independent test set of samples.

The fundamental extension of DA to supervised learning by inclusion of structural constraints [69], [70] led first to the DA method for classification [70], then to the DA method for piecewise regression [78], and was later extended to address regression by mixture of experts [82]. Here, after formulating the problem in its general setting and deriving a DA method for its solution, we show how it specializes to various classical regression and classification problems by defining the structure (parametric form of the function/system) and the cost criterion. We demonstrate that the DA approach results in a powerful technique for classifier and regression function design for a variety of popular system structures.

A. Problem Formulation

We define the objective of supervised learning as that of approximating an unknown function from the observation of a limited sequence of (typically noise-corrupted) input–output data pairs. Such function approximation is typically referred to as regression if the output is continuous, and as classification if the output is discrete or categoric. Regression and classification are important tools in diverse areas of signal processing, statistics, applied mathematics, business administration, computer science, and the social sciences.

For concreteness, the problem formulation here will employ the terminology and notation of the regression problem. We will later show how the solution is, in fact, applicable to classification and other related problems. The regression problem is usually stated as the optimization of an expected cost that measures how well the regression

function $g(X)$, applied to random vector X, approximates the output Y over the joint distribution $p(x, y)$, or in practice, over a sample set $\{(x, y)\}$. Let us reformulate the cost as

$$D = \sum_x \sum_y p(x, y)\, d(y, g(x)) \tag{51}$$

where the distortion measure $d(\cdot, \cdot)$ is general, though the squared error is most often in use. The cost is optimized by searching for the best regression function g within a given parametric class of functions. We shall first restrict ourselves to space-partitioning regression functions. These functions are often called piecewise regression functions because they approximate the desired function by partitioning the space and matching a simple local model to each region. Space-partitioning regression functions are constructed of two components: a parametric space partition (structured partition) and a parametric local model per partition cell. Let the partition parameter set be denoted by Ω. It may consist of the nodes of a decision tree, the codevectors or prototypes of a vector quantizer, the weights of a multilayer perceptron, etc. Let $\Lambda = \{\Lambda_j\}$ denote the set of local model parameters, where Λ_j is the subset of parameters specifying the model for region R_j.

We can now write the regression function as

$$g(x) = f(x, \Lambda_k), \qquad \forall x \in R_k \tag{52}$$

where $f(x, \Lambda_k)$ denotes the local model. Note that Ω is implicit in the partition into cells $\{R_k\}$. Typically, the local parametric model $f(x, \Lambda_k)$ has a simple form such as constant, linear, or Gaussian. The average regression error measured over a sample set is then

$$D = \frac{1}{N} \sum_j \sum_{(x, y):\, x \in R_j} d(y, f(x, \Lambda_j)). \tag{53}$$

The regression function $g(\cdot)$ is learned by minimizing the design cost, D, measured over a training set, $\mathcal{T} = \{x, y\}$, 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 decreases as the complexity (size) of the learned regression model is allowed to increase, 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. Most techniques for improving generalization in learning are inspired by the well-known principle of Occam's razor,[8] 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], [85], 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 CART [13] and in methods such as optimal brain surgeon [110] 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 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 to the training set and 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, standard methods such as classification and regression trees (CART) [13] for tree-structured classification and 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., [110]).

As an alternative approach, let us consider the DA 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.

B. Basic Derivation

The design objective is minimization of the expected regression cost D of (53) which is repeated here for convenience

$$D = \frac{1}{N} \sum_j \sum_{(x, y):\, x \in R_j} d(y, f(x, \Lambda_j)) \tag{54}$$

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

over the partition parameters Ω (which are implicit in the partition $\{R_j\}$) and the local model parameters Λ. To begin the derivation let us assume that the local model parameters Λ are known and fixed and focus on the more difficult problem of optimizing the partition. The partition is structurally constrained, that is, it is selected from a family of partitions by assigning values to the set of parameters Ω. Note that a partition is, in fact, a classifier as it assigns to each input a label indicating to which partition cell it belongs. There are many popular structures for the partition such as the vector quantizer, the decision tree, the multilayer perceptron, and the radial basis functions partitions. The operation of each of the above distinct structures (or classifiers) is consistent with that of the general (canonical) maximum discriminant model [23]: given input x the system produces competing outputs (one per partition cell) via the discriminant functions $\{F_j(x)\}$, and the input is assigned to the largest, "winning" output. It thus uses the "winner-take-all" partition rule

$$R_j \equiv \{x\colon F_j(x) \geq F_k(x),\ \forall k\}. \tag{55}$$

Any partition can be represented by this model, albeit possibly with complicated discriminant functions. Note that the discriminant functions $F_j(x)$ in our case are specified by the set of parameters Ω, although we have suppressed this dependence in the notation of (55). We thus employ the maximum discriminant model to develop a general optimization approach for regression. We will later specialize the results to specific popular structures and learning costs and give experimental results to demonstrate the performance.

Let us write an objective function whose maximization determines the hard partition for given Ω

$$S_h = \frac{1}{N} \sum_{j \in \mathcal{I}} \sum_{x \in R_j} F_j(x). \tag{56}$$

Note, in particular, that the winner-take-all rule (55) is optimal in the sense of S_h. Specifically, maximizing (56) over all possible partitions captures the decision rule of (55).

To derive a DA approach we wish to randomize the partition similar to the earlier derivation for the problem of clustering. The probabilistic generalization of (56) is

$$S = \frac{1}{N} \sum_x \sum_j p(j|x) F_j(x) \tag{57}$$

where the partition is now represented by association probabilities $\{p(j|x)\}$ and the corresponding entropy is

$$H = -\frac{1}{N} \sum_x \sum_j p(j|x)\ \log p(j|x). \tag{58}$$

It is emphasized that H measures the average level of uncertainty in the partition decisions. We determine our assignment distribution at a given level of randomness as the one which maximizes S while maintaining H at a prescribed level $\hat{H}$

$$\max_{\{p(j|x)\}} S \qquad \text{subject to } H = \hat{H}. \tag{59}$$

The result is the best probabilistic partition, in the sense of the structural objective S, at the specified level of randomness. For $\hat{H} = 0$ we naturally revert to the hard partition which maximizes (56) and thus employs the winner-take-all rule. At any positive $\hat{H}$, the solution of (59) is the Gibbs distribution

$$p(j|x) = \frac{e^{\gamma F_j(x)}}{\sum_k e^{\gamma F_k(x)}} \tag{60}$$

where γ is the Lagrange multiplier controlling the level of entropy. For $\gamma \to 0$ the associations become increasingly uniform, while for $\gamma \to \infty$ they revert to the hard partition, equivalent to application of the rule in (55). Thus, (60) is a probabilistic generalization of the winner-take-all rule which satisfies its structural constraint, specified by (55), for the choice $\gamma \to \infty$. Note that beside the obvious dependence on the parameter γ, the discriminant functions $\{F_j(x)\}$ are determined by $\Omega = \{\Omega_j\}$.

So far, we have formulated a controlled way of introducing randomness into the partition while enforcing its structural constraint. However, the derivation assumed that the model parameters were given, and thus produced only the form of the distribution $p(j|x)$, without actually prescribing how to choose the values of its parameter set. Moreover, the derivation did not consider the ultimate goal of minimizing the expected regression cost D. We next remedy both shortcomings.

To apply the basic principles of DA design similar to our treatment of clustering, we need to introduce randomness into the partition while enforcing the required structure, only now we must also explicitly minimize the expected regression cost. *A priori*, satisfying these multiple objectives may appear to be a formidable task, but the problem is greatly simplified by restricting the choice of random partitions to the set of distributions $\{p(j|x)\}$ as given in (60)—these random partitions naturally enforce the structural constraint of (55) through γ, as explained earlier. Thus, from the parameterized set $\{p(j|x)\}$ (determined by the implicit Ω), we seek that distribution which minimizes the expected regression cost while constraining the entropy

$$\min_{\Omega, \Lambda} D \equiv \min_{\Omega, \Lambda} \frac{1}{N} \sum_{(x,\, y)} \sum_j p(j|x)\, d(y,\, f(x,\, \Lambda_j)) \tag{61}$$

subject to

$$H = \hat{H}. \tag{62}$$

The solution yields the best random partition and model parameters in the sense of minimum D for a given entropy level $\hat{H}$. At the limit of zero entropy, we should get a hard partition which minimizes D, yet has the desired structure, as specified by (55).

We naturally reformulate (61) and (62) as minimization of the unconstrained Lagrangian, or free energy

$$F = D - TH \tag{63}$$

where the Lagrange parameter T is the "temperature" and emphasizes the intuitively compelling analogy to statistical

physics, in parallel to the DA derivation in the earlier sections. Virtually all the discussion on the analogy to statistical physics which appeared in the context of clustering holds here too, and it provides strong motivation for use of the DA method. For conciseness we shall not elaborate on the analogy here.

We initialize the algorithm at $T \to \infty$ (in practice, T is simply chosen large enough to be above the first critical temperature). It is clear from (63) that the goal at this temperature is to maximize the entropy of the partition. The distributions $\{p(j|x)\}$ 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, optimization is carried out at each temperature to find the partition parameters $\{\Omega_j\}$ and local model parameters $\{\Lambda_j\}$ that minimize the Lagrangian F. 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 partition satisfying the imposed structure. In practice, we anneal the system to a low temperature, where the entropy of the random partition is sufficiently small. Further annealing will not change the partition parameters significantly. Hence we fix the partition parameters at this point and jump (quench) to $T = 0$ to perform a "zero entropy iteration," where we partition the training set according to the "hard" partition rule and optimize the parameters of the local models $\{\Lambda_j\}$ to minimize the regression cost D. This approach is consistent with our ultimate goal of optimizing the 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) $\min_{\Omega,\Lambda} F = D - TH$.
3) *Lower Temperature*: $T \leftarrow q(T)$.
4) If $H \geq H_f$ go to 2).
5) *Zero Entropy Iteration*: Partition using hard partition 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 2) may be performed by any local optimization method.

C. Generality and Wide Applicability of the DA Solution

1) Regression, Classification, and Clustering: In Section III-B, we derived a DA method to design a regression function subject to structural constraints on the partition. In this section we pause to appreciate the general applicability of the DA solution. We show that special cases of the problem we defined include the problems of clustering and vector quantization, as well as statistical classifier design. These special cases are obtained by specifying appropriate cost functions and local models. We also review a number of popular structures from data compression and neural networks and show how they are special cases of the general maximum discriminant structure, and hence directly handled by the DA approach we have derived.

Let us first restate the learning problem. Given a training set of pairs (x, y), we wish to design a function which takes in x and estimates y. The estimator function is constructed by partitioning the input space and fitting a local model within each region. The learning cost is defined as

$$D = \frac{1}{N} \sum_j \sum_{(x,y): x \in R_j} d(y, f(x, \Lambda_j)) \qquad (64)$$

where $\{R_j\}$ is the set of partition regions and $\Lambda = \{\Lambda_j\}$ is the set of parameters which determine the local models. Beside the obvious and direct interpretation of the above as a regression problem with applications in function approximation and curve fitting, it is easy to see that the important problem of classifier design is another special case of this learning problem: if the local model $f(x, \Lambda_j)$ is simply the class label assigned to the region, and if we define the distortion measure as the error indicator function $d(u, v) = 1 - \delta(u, v)$, where the δ function takes the value one when its arguments are equal and vanishes otherwise, then the learning cost is exactly the rate of misclassification or error rate of the classifier. Thus, statistical classifier design is a special case of the general learning problem we considered, albeit with a particularly difficult cost to optimize due to its discrete nature. This is a very important problem with numerous "hot" applications in conjunction with various structures, and we will devote more space to it in the sequel.

A somewhat contrived, yet important, special case of the above regression problem is that of unsupervised clustering. Here we consider the degenerate case of $y = x$ where the local regression models are constant. In other words, we approximate the training set with a piecewise-constant function. We partition the space into regions, and each region is represented by a constant vector (the codevector) so as to minimize the cost (e.g., MSE). This is clearly the vector quantization problem. If we apply our DA regression method to this problem, and assume a vector quantizer structure, we will get exactly the clustering approach we had derived directly, and more simply, earlier on. The simpler derivation of a DA clustering method was only possible because the VQ structure emerges by itself from minimization of the clustering distortion, and need not be externally imposed as in the case of the general DA regression method. Although the latter derivation seems unnecessarily cumbersome for clustering problems, it does in fact open the door to important clustering applications. We are often interested in solving the clustering problem while imposing a different structure. The typical motivation is that of reducing encoding or storage complexity, but it may also be that of discerning hierarchical information, or because a certain structure better fits prior information on the underlying distribution. We have already considered in detail tree-structured clustering within the unsupervised learning section, however we had to postpone the complete description of the mechanism for enforcing the structure until the supervised learning section.

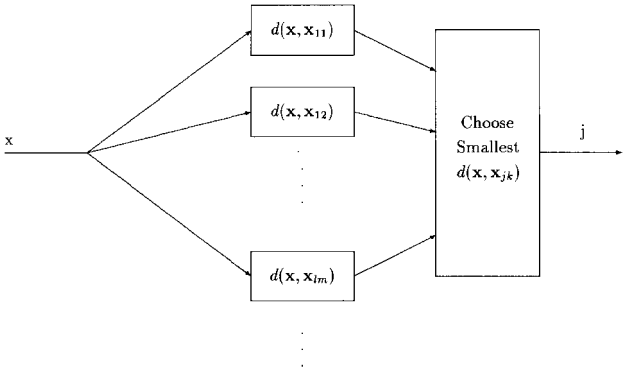


Fig. 8. The VQ classifier architecture. From [70].

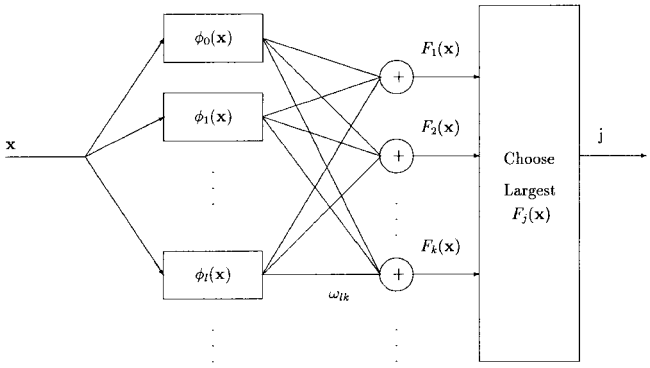


Fig. 9. The RBF classifier architecture. From [70].

2) Structures: We next consider the applicability of the approach to a variety of structures. Recall that the approach was generally derived for the maximum discriminant partition structure which is defined by

$$R_j \equiv \{x: F_j(x) \geq F_k(x) \; \forall k\}. \tag{65}$$

This general structure can be specialized to specific popular structures such as the vector quantizer (or nearest prototype classifier), the multilayer perceptron, and the radial basis functions classifier. It is important to note that known design methods are structure specific, while the DA approach is directly applicable to virtually all structures. In the remainder of this section we describe these three structures. The choice of presentation is such that the applicability of the general design procedure is evident. For detailed structure-specific derivation see [70].

a) The VQ classifier: The VQ structure is shown in Fig. 8. The partition is specified by the parameter set $\Omega = \{\mathbf{x}_{jk}\}$ where $\mathbf{x}_{jk} \in \mathcal{R}^n$ is the kth prototype associated with class j. The VQ classifier maps a vector in $\mathcal{R}^n$ to the class associated with the nearest prototype, specifying a partition of $\mathcal{R}^n$ into the regions

$$R_j \equiv \bigcup_k S_{jk}$$

with

$$S_{jk} \equiv \{\mathbf{x} \in \mathcal{R}^n: d(\mathbf{x}, \mathbf{x}_{jk}) \leq d(\mathbf{x}, \mathbf{x}_{lm}) \; \forall l, m\} \tag{66}$$

i.e., each region R_j is the union of Voronoi cells S_{jk}. Here, $d(\cdot, \cdot)$ is the "distance measure" used for classification. For consistency with the maximum discriminant classifier ("winner takes all") we note trivially that the classification rule can also be written as

$$R_j \equiv \bigcup_k S_{jk}$$

with

$$S_{jk} \equiv \{\mathbf{x} \in \mathcal{R}^n: F_{jk}(\mathbf{x}) \geq F_{lm}(\mathbf{x}) \; \forall l, m\} \tag{67}$$

by choosing $F_{jk}(\mathbf{x}) \equiv -d(\mathbf{x}, \mathbf{x}_{jk})$.

b) The radial basis functions (RBF) classifier: The RBF classifier structure is shown in Fig. 9. The classifier is specified by a set of Gaussian receptive field functions $\{e^{-(|x-\mu_k|^2/\sigma_k^2)}\}$ and by a set of scalar weights $\{\omega_{kj}\}$ which connect each of the receptive fields to the class outputs of the network. Thus, $\Omega = \{\{\mu_k\}, \{\sigma_k^2\}, \{\omega_{kj}\}\}$. The parameter μ_k is the "center" vector for the receptive field and σ_k^2 is its "width." In the "normalized" representation for RBF's [75] which we will adopt here, the network output for each class is written in the form

$$F_j(x) = \sum_k \omega_{kj} \phi_k(x) \tag{68}$$

where

$$\phi_k(x) = \frac{e^{-\left(|x-\mu_k|^2/\sigma_k^2\right)}}{\sum_l e^{-\left(|x-\mu_l|^2/\sigma_l^2\right)}}. \tag{69}$$

Since $\phi_k(\cdot)$ can be viewed as a probability mass function, each network output is effectively an average of weights emanating from each of the receptive fields. The classifier maps the vector x to the class with the largest output

$$R_j \equiv \{x \in \mathcal{R}^n: F_j(x) \geq F_k(x) \; \forall k \in \mathcal{I}\}. \tag{70}$$

c) The multilayer perceptron (MLP) classifier: The MLP classifier structure is shown in Fig. 10. We restrict ourselves to the MLP structure with a binary output unit per class. The classification rule for MLP's is the same as that for RBF's (70), but the output functions $\{F_j(\cdot)\}$ are parametrized differently.

The input x passes through K layers with M_k neurons in layer k. We define u_{kj} to be the output of hidden unit j in layer k, with the convention that layer zero is the input layer $u_{0j} = x_j$ and layer K is the output layer $u_{Kj} = F_j(\mathbf{x})$. To avoid special notation for thresholds, we define the augmented output vector of layer k as $\tilde{\mathbf{u}}_k = [u_{k1} \; u_{k2} \; \ldots \; u_{kM_k} \; 1]^T$. This is a standard notation allowing us to replace thresholds by synaptic weights which multiply a fixed input value of unity. The weight matrix $\mathbf{W}_k$ connects the augmented outputs of layer $k-1$ and the neurons of layer k. The activation function of the kth layer is the vector valued function $\mathbf{f}_k$: $\mathcal{R}^{M_k} \rightarrow \mathcal{R}^{M_k+1}$ defined as $\mathbf{f}_k(\mathbf{v}) = [f_k(v_1) \; f_k(v_2) \; \ldots \; f_k(v_{M_k}) \; 1]^T$ where

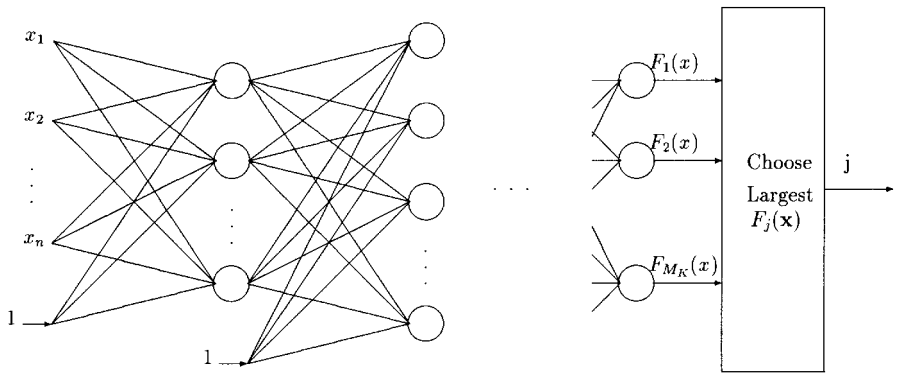


Fig. 10. The MLP classifier architecture. From [70].

$f_k(\cdot)$ is the scalar activation function used by all neurons in the kth layer. In our experiments, we used the logistic activation function $(f_k(v) = (1/1 + e^{-v}))$ for the hidden layers $k = 1, \ldots, K-1$, and the linear activation function $(f_K(v) = v)$ for the output layer. The activity level at the input of the kth layer is given by

$$\mathbf{v}_k = \mathbf{W}_k \tilde{\mathbf{u}}_{k-1}. \tag{71}$$

Thus, the network's operation can be described by the following recursion formula

$$\tilde{\mathbf{u}}_k = \mathbf{f}_k(\mathbf{v}_k) = \mathbf{f}_k(\mathbf{W}_k \tilde{\mathbf{u}}_{k-1}) \qquad k = 1, 2, \ldots, K. \tag{72}$$

D. Experimental Results

The general DA method for supervised learning has been specialized to specific design problems and tested. In particular, results are given for classifier design for the VQ, RBF, and MLP structures, piecewise regression, and mixture of experts regression. The exact equations used in the iteration depend on the structure, and can be derived in a straightforward manner from the general design approach. For more details on specific DA design refer to [70] for classifier design, [78] for piecewise regression, and [82] for mixture of experts.

1) VQ Classifier Design: The DA approach to VQ classifier design [70] is compared with the learning VQ (LVQ) method [56]. Note that here LVQ will refer narrowly to that design method, not to the structure itself which we call VQ. The first simulation result is on the "synthetic" example from [83], where DA design achieved $P_e = 8.9\%$ on the test set using eight prototypes and $P_e = 8.6\%$ using 12 prototypes, in comparison to LVQ's $P_e = 9.5\%$ based on 12 prototypes. (For general reference, an MLP with six hidden units achieved $P_e = 9.4\%$.) For complicated mixture examples, with possibly 20 or more overlapping mixture components and multiple classes, the DA method was found to consistently achieve substantial performance gains over LVQ. As an example, consider the training data for a four-class problem involving 24 overlapping, nonisotropic mixture components in two dimensions, shown in Figs. 11 and 12. VQ-classifiers with 16 prototypes (four per class) were designed using both LVQ and DA. Figs. 11(a) and 12(a) display the data and partitions formed by the two methods. Figs. 11(b) and 12(b) display the prototype locations along with the partitions. The best LVQ solution based on ten random initializations, shown in Fig. 11, achieved $P_e = 31\%$. Note that the method has failed to distinguish a component of class 0 in the upper left of Fig. 11(a), as well as a component of class 1 near the lower right of the figure. By contrast, the DA solution shown in Fig. 12 succeeds in discriminating these components, and achieves $P_e = 23\%$.

Another benchmark test data is the Finnish phoneme data set that accompanies the standard LVQ package. The training set consists of 1962 vectors of 20 dimensions each. Each vector represents speech attributes extracted from a short segment of continuous Finnish speech. These vectors are labeled according to the phoneme uttered by the speaker. There are 20 classes of phonemes in the training set. In both LVQ and DA approaches, the number of prototypes associated with a particular class was set to be proportional to the relative population of that class in the training set. This is referred to as the *propinit* initialization in the standard LVQ package. The experimental results are shown in Table 1. Note that the DA method consistently outperformed LVQ over the entire range.

2) RBF Classifier Design: The DA approach to RBF design [70] is compared here with the method of Moody and Darken [75] (MD-RBF), with a method described by Tarassenko and Roberts [10] (TR-RBF), and with the gradient method of steepest descent on $\langle P_e \rangle$ (G-RBF). MD-RBF combines unsupervised learning of receptive field parameters with supervised learning of $\{\omega_{kj}\}$ to minimize the squared distance to target class outputs. The primary advantage of this approach is its modest design complexity. However, the receptive fields are not optimized in a supervised fashion, which can cause performance degradation. TR-RBF, one of the methods described in [101], optimizes all of the RBF parameters to approximate target class outputs in a squared error sense. This design is more complex than MD-RBF and achieves better performance for a given model size (the number of receptive fields the classifier uses). However, the TR-RBF design objective is not equivalent to minimizing P_e, but as in the case of back propagation, it effectively aims to approximate the Bayes-optimal discriminant. While direct descent on

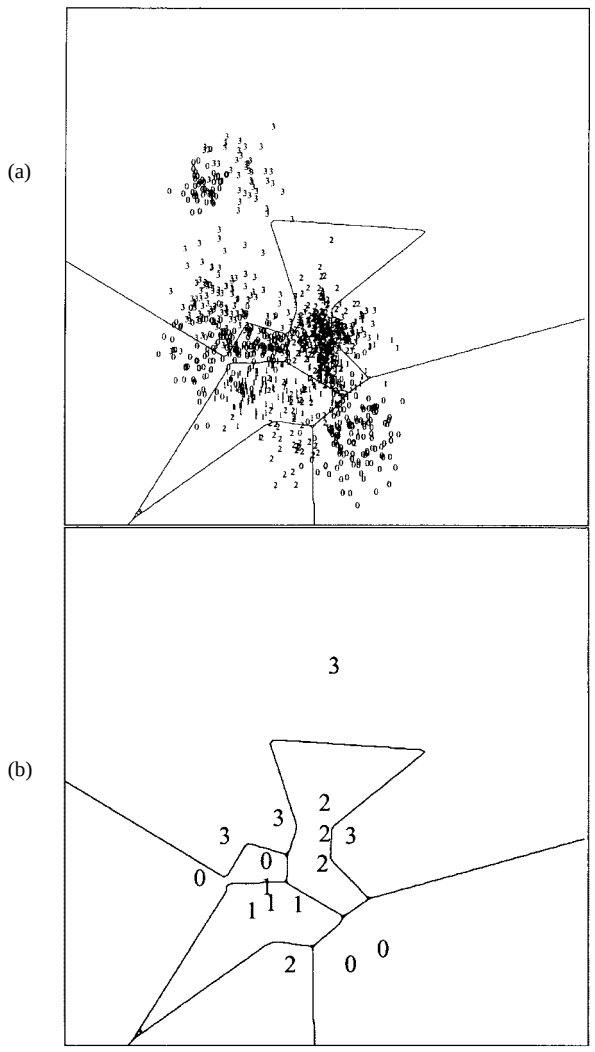


Fig. 11. (a) A four-class Gaussian mixture training set for a VQ classifier design and the partition produced by LVQ and (b) the LVQ partition, with the 16-class prototype. Locations of prototypes shown. The error rate is $P_e = 31\%$. From [70].

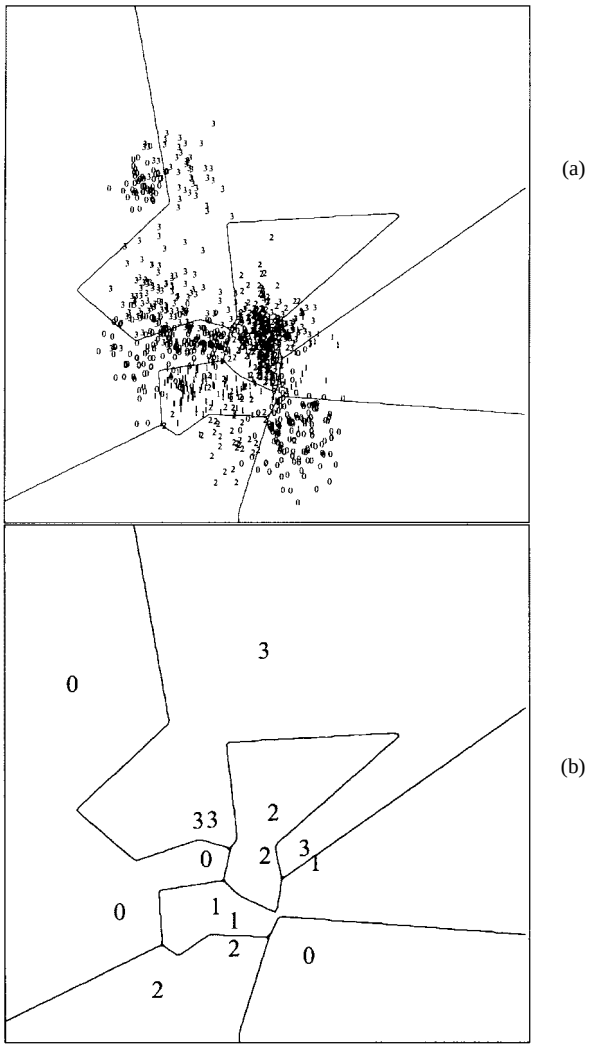


Fig. 12. (a) The four-class Gaussian mixture training set for a VQ classifier design and the partition produced by DA and (b) the DA partition with the 16-class prototypes shown. The error rate is $P_e = 23\%$. From [70].

$\langle P_e \rangle$ may minimize the "right" objective, problems of local optima may be quite severe. In fact, we have found that the performance of all of these methods can be quite poor without a judicious initialization. For all of these methods, we have employed the unsupervised learning phase described in [75] (based on Isodata clustering and variance estimation) as model initialization. Then, steepest descent was performed on the respective cost surface. We have found that the complexity of our design is typically 1–5$\times$ that of TR-RBF or G-RBF (though occasionally our design is actually faster than G-RBF). Accordingly, we have chosen the best results based on five random initializations for these techniques and compared with the single DA design run.

To illustrate that increasing M may not help to improve performance on the test set, we compared DA with the

Table 1 Error Probability Comparison of the DA and LVQ Methods for the Design of VQ Classifiers on the 20-Dimensional, 20-Class Finnish Phoneme Data Set That Accompanies the Standard LVQ Package. M Represents the Total Number of Prototypes

M (# of units)	20	30	40	50	80	100	200
P_e (LVQ)	13.25	12.44	11.47	10.96	10.09	8.17	6.78
P_e (DA)	11.67	9.99	8.36	5.55	4.83	4.23	3.26

results reported in [76] for two-dimensional (2-D) and eight-dimensional (8-D) mixture examples. For the 2-D example, DA achieved $P_{e_{\text{train}}} = 6.0\%$ for a 400-point training set and $P_{e_{\text{test}}} = 6.1\%$ on a 20 000-point test set, using $M = 3$ units. (These results are near-optimal, based on the Bayes rate.) By contrast, the method in [76] used 86 receptive fields and achieved $P_{e_{\text{test}}} = 9.26\%$. For the 8-D example and $M = 5$, DA achieved $P_{e_{\text{train}}} = 8\%$ and

Table 2 Error Probability Comparison of DA and Other Design Techniques for RBF Classification on the 21-Dimensional Waveform Data from [13]. M is the Number of Receptive Fields. DA is compared with TR-RBF [101], MD-RBF [75], and with G-RBF, Which Is Gradient Descent on $\langle P_e \rangle$. The Test Set Performance Results Have 95% Confidence Intervals of Half-Length Less Than 2%

Method	DA	TR-RBF				MD-RBF		G-RBF		
M (# of units)	3	3	5	15	25	10	30	5	10	15
P_e (training set)	14.0	38.0	15.0	14.0	10.0	25.0	18.0	48.0	21.0	14.0
P_e (test set)	16.0	38.0	22.0	18.0	17.0	26.0	19.0	47.0	19.0	16.0

Table 3 Error Probability Comparison of DA with Known Design Techniques for RBF Classification on the 40-Dimensional Noisy Waveform Data from [13]. The Test Set Performance Results Have 95% Confidence Intervals of Half-Length Less Than 3.0%

Method	DA		TR-RBF				MD-RBF		G-RBF
M (# of units)	4	30	4	10	30	50	10	50	10
P_e (training set)	11.0	2.8	33.0	16.2	14.5	12.9	30.0	19.0	18.0
P_e (test set)	13.0	16.7	35.0	16.5	16.8	17.9	37.0	18.0	20.0

$P_{e_{\text{test}}} = 9.4\%$ (again near-optimal), while the method in [76] achieved $P_{e_{\text{test}}} = 12.0\%$ using $M = 128$.

More comprehensive tests on higher dimensional data have also been performed. Two examples reported here are the 21-dimensional (21-D) waveform data and the 40-dimensional (40-D) "noisy" waveform data used in [13] (obtained from the UC-Irvine machine learning database repository). Rather than duplicate the experiments conducted in [13], we split the 5000 vectors into equal size training and test sets. Our results in Tables 2 and 3 demonstrate quite substantial performance gains over all the other methods, and performance quite close to the estimated Bayes rate of 14% [13]. Note in particular that the other methods perform quite poorly for small M and need to increase M to achieve training set performance comparable to our approach. However, performance on the test set does not necessarily improve, and may degrade for large M.

3) MLP Classifier Design: The DA approach for designing MLP's [70] is compared with two other approaches—the standard back propagation (BP) algorithm of [94] and gradient descent on the $\langle P_e \rangle$ cost surface (G-MLP). The BP weights were initialized to random numbers uniformly distributed between ± 0.01. A total of 50 000 epochs of a batch gradient descent algorithm were run to minimize the MSE between the desired and actual outputs of the MLP. BP, however, descends on a cost surface mismatched to the minimum P_e objective. Further, its performance is dependent on the choice of initial weights. In G-MLP the performance of BP is improved by taking the BP solution as initialization and then descending on $\langle P_e \rangle$. However, in practice, the gains achieved by G-MLP over BP are only marginal, as the optimization performance sensitively depends on the choice of initialization.

The performance is tested on the 19-dimensional 7-class image segmentation data from the University of California-Irvine machine learning database. The training set contains 210 vectors and the test set contains 2100 vectors, each 19-dimensional. The features represent various attributes of a 3×3 block of pixels. The classes correspond to textures (brickface, sky, foliage, cement, window, path, grass). A sequence of single hidden layer neural networks was designed based on this data set. Table 4 summarizes the results for various hidden layer sizes (M). Networks designed by DA significantly outperformed the other approaches over the entire range of network sizes.

Table 4 Error Probability Comparison of the BP and G-MLP Design Approaches on the 19-Dimensional 7-Class Segmentation Data Example. The Test Set Performance Results Have 95% Confidence Intervals of Half-Length Less Than 2.1%

	DA			BP				G-MLP			
M	4	6	8	4	6	8	10	4	6	8	10
P_e (training set)	19.1	8.1	6.2	45.7	31.9	20.0	6.1	45.7	31.4	19.6	6.0
P_e (test set)	20.1	11.2	10.5	48.1	31.7	25.3	13.3	47.2	34.4	23.2	13.0

An important concern is the issue of design complexity. In the above experiments the DA learning complexity was roughly 4–8× higher than that of back propagation and roughly the same as that of G-MLP. This suggests that the potential for performance improvement would, in typical applications, greatly outweigh the somewhat higher design complexity of the DA approach.

4) Piecewise Regression: Here we summarize experiments comparing the performance of the DA approach for VQ-based piecewise regression [78] with the conventional piecewise regression approach of CART [13]. Note that regular CART is severely restricted in that its 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 $\mathbf{X}$ can be overcome by adopting an extension of CART that allows the boundaries between regions to be arbitrary linear hyperplanes. While this extension allows better partitioning of the input space and hence smaller approximation error, the complexity of the design method for the extended structure [46] 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 short. In this section, we will refer to regular CART as CART1, and 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 Breiman–Friedman–Olshen–Stone algorithm (see e.g., [18]). 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. Unlike CART2, the complexity of the DA method is linear in the size of the training set. Further, 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 following comparisons, the models are piecewise constant, which is the simplest example of piecewise regression. In our implementation of the DA method we used an annealing schedule given by $q(T) = 0.95T$.

The first experiment involves synthetic datasets, where the regression input $x = (x_0, x_1)$ is 2-D and the output y is one-dimensional. The input components x_0, x_1 are each uniformly distributed in the interval, (0, 1). The output y is a sum of six normalized Gaussian-shaped functions of

Table 5 Mean Squared Approximation Error Measured over the Test Set and Model Order for the Best Solutions Produced by Cart1 and by DA for Multimodal Gaussian Data Sets

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)

x each with an individual center, variance, and magnitude. By choosing different sets of parameters (centers, variances, and magnitudes) for the Gaussians, we created a number of data sets, each consisting of a training and a validation set of size 1000 each and a test set of size 3000. The output samples were corrupted by a zero-mean Gaussian noise with variance 10.0. To compare the design approaches, DA and CART were applied to design regression functions for each dataset using the training and validation sets (validation was used to select the best model size for generalization) and the performance was evaluated on the independent test sets. The experiments were conducted over more than 40 different data sets. Table 5 provides a randomly selected subset of the results. Note that in this case DA is only compared with standard CART1, since CART2 is too complex for training sets of this size. Clearly, for all the examples, DA demonstrates consistent improvements over CART1.

We next compare CART with DA over a few data sets from real-world regression applications. This data is taken from the StatLib database[9] and has been extensively used by researchers in benchmarking the relative performance of competing regression methods. However, due to the unavailability of sufficient data for proper validation, we simply compare the performance of the two regression models versus model size.

One benchmark problem is concerned with predicting the value of homes in the Boston area from a variety of parameters [43]. 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, they were normalized to unit variance prior to application of DA and CART. Piecewise constant regression models of different model sizes were generated by the design methods. Table 6 compares 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. Clearly, the DA method substantially outperforms both CART1 and CART2 over the entire range of model sizes. This example illustrates that DA find substantially better solutions for the design objective. 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 difficulties due to local minima.

[9] Available WWW: http://lib.stat.cmu.edu/data sets/.

Table 6 Mean Squared Prediciton Error for Housing Price in the Boston, MA Area. Comparison of Training Set Errors for the Standard Cart1, Its Extension Cart2, and the DA Method. K Is the Number of Regions Allowed for Each Model

Method	K=1	K=2	K=4	K=6	K=9
CART1	84.42	46.2	25.7	17.86	12.53
CART2	84.42	43.17	26.10	21.87	18.74
DA	84.42	34.35	16.88	11.00	8.61

Table 7 Mean-Squared Prediction Error for the Age-Adjusted Mortality Rate Per 100 000 Inhabitants from Various Environmental Factors. Comparison of Cart1, Cart2, and DA. K Is the Number of Regions Allowed for Each Model

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

The data set for the next example was taken from the environmental sciences. This problem is concerned with predicting the age-adjusted mortality rate per 100 000 inhabitants of a locality from 15 factors that have presumably influenced it. Some of these factors are related to the levels of environmental pollution in the locality, while others are measurements of nonenvironmental, mainly social parameters. This data set has been used by numerous researchers since its introduction in the early 1970's [66]. As data are only available for 60 localities, they were not divided into separate training and test sets. We only show performance on the training set. Table 7 shows that the VQ-based regression function designed by DA offers a consistent substantial advantage over CART for the entire range of model sizes.

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 electromagnetic 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 172 and a test set of size 43. We then applied CART1, CART2, and DA to the training set for different model sizes. Table 8 compares the mean-squared approximation error obtained over the training and test sets for all the methods. DA significantly outperformed the CART regression functions that used the same number of regions in the input space. In fact, using five prototype DA's produced a regression function that outperformed both of the CART regression function with ten regions. The excellent performance of the DA method outside the training set confirms its expected good generalization capabilities. Note also that the CART2 method exhibits overfitting with the test set performance deteriorating from $K = 5$ to $K = 10$.

Table 8 Mean-Squared Approximation Error for the Fat Content of a Meat Sample from 100 Spectroscopic Measurements. The Performance of Cart1 and Cart2 Is Compared with that of the Proposed DA Method, Both Inside (TR) and Outside (TE) the Training Set. K Is the Number of Regiouns Used to Represent the Data

Method	tr/te	K=1	K=2	K=3	K=4	K=5	K=10
CART1	tr	159.89	113.86	106.93	101.66	85.27	48.20
	te	168.25	141.34	142.74	140.54	109.54	107.28
CART2	tr	159.89	67.35	49.84	37.15	28.08	16.22
	te	168.25	84.98	69.92	57.99	39.44	50.63
DA	tr	159.89	38.05	38.05	26.67	15.55	8.11
	te	168.25	39.85	37.03	26.47	14.27	14.10

5) Mixture of Experts: Mixture of experts is an important type of structures that was inspired by mixture models from statistics [67], [102]. This class includes the structures known as "mixture of experts" [51] and "hierarchical mixture of experts" (HME) [53], as well as normalized radial basis functions (NRBF) [75]. We refer to this class generally as mixture of experts (ME) models. ME's have been suggested for a variety of problems, including classification [48], [51], control [50], [53], and regression tasks [53], [104], [105].

We define the "local expert" regression function $f(x, \Lambda_j)$, where Λ_j is the set of model parameters for local model j. The ME regression function is defined as

$$g(x) = \sum_j p[j|x] f(x, \Lambda_j) \tag{73}$$

where, $p[j|x]$ is a nonnegative weight of association between input x and expert j that effectively determines the degree to which expert j contributes to the overall model output. In the literature, these weights are often called gating units [51], and obey some prespecified parametric form. We further impose $\sum_j p[j|x] = 1$, which leads to the natural interpretation of the weight of association or gating unit as a probability of association.

ME is an effective compromise between purely piecewise regression models discussed earlier, such as CART [13], and "global" models such as the MLP [94]. By "purely piecewise" it is meant that the input space is hard-partitioned to regions, each with its own exclusive expert model. Effectively, the piecewise regression function is composed of a patchwork of local regression functions that collectively cover the input space. In addition to partitioning the input space, the model parameter set is partitioned into submodels which are only "active" for a particular local input region. By contrast, in global models such as MLP's there is a single regression function that must fit the data well everywhere, with no explicit partitioning of the input space nor subdivision of the parameter set. ME exploits a partition of the space but produces a function which combines the contributions of the various experts with some appropriate weighting.

The DA design approach [82] is based on controlling the entropy of the association probabilities $p[j|x]$. In this case, these probabilities are part of the problem definition rather than an artificial addition to avoid nonglobal minima. It is important to note that as we approach zero temperature, the entropic constraint simply disappears and we find the solution which minimizes the cost regardless of its entropy. Annealing consists of starting at high temperature (high entropy) and gradually allowing the entropy to drop to its optimal level (where the cost is minimized).

The following results compare the DA approach with conventional design methods for NRBF and HME regression functions. The experiments were performed over some popular benchmark data sets from the regression literature. In each experiment, we compare the average squared-error obtained over the training set using the DA design method and the alternative design methods. The comparisons are repeated for different network sizes. The network size K refers to the number of local experts used in the mixture model. For the case of binary HME trees with l levels $K = 2^l$, and for the case of NRBF regression functions K is the number of Gaussian basis functions used. Following the most common implementation, the local models are constant functions in the NRBF case and linear functions in the HME case. The alternative design approaches used for comparing our HME design algorithm are:

GD: a gradient descent algorithm to simultaneously optimize all HME parameters for the squared-error cost;

ML: Jordan and Jacobs's maximum likelihood approach [53].

For the NRBF regression function, we have compared the DA design approach [82] with the GD algorithm, which is an enhanced version of the method suggested in [75] (see [82] for details). For fair comparison, we take a conservative (worst case) estimate that the complexity of the DA approach is 10× greater than that of the competing methods (in fact the complexity of DA was higher by a factor of two–ten in these experiments). To compensate for the complexity, we allow each competing method to generate results based on ten different random initializations, with the best result obtained among those runs selected for comparison with the DA result. Since the regression function obtained by DA is generally independent of initialization, a single DA run sufficed.

Let us first consider the results for the real-world examples that had been used in the piecewise regression subsection. Results for the Boston home-value prediction problem are given in Tables 9 and 10 and demonstrate that for both mixture models the DA approach achieves a significantly smaller regression error compared with the other approaches over a variety of network sizes. Results for the mortality rate example are given in Tables 11 and 12; and the meat fat-content results (measurements were obtained by the Tecator Infratec Food and Feed Analyzer) are given in Tables 13 and 14.

Finally, results are given for synthetic data. Here, $x = (x_0, x_1)$ is 2-D and the training set is generated according to a uniform distribution in the unit square. The output y is scalar. We created five different data sets based on the functions $(f_1(), f_2(), \ldots, f_5())$ specified in [17] and

Table 9 Comparison of Regression Error Obtained Using DA and GD Algorithms for NRBF Design for the Boston Home Value Problem. K Is the number of Gaussian Basis Functions Used

K	DA	GD
1	87.7	87.7
2	19.7	23.8
4	12.9	19.3
6	12.6	15.7
10	6.5	13.7

Table 10 Comparison of Regression Error Obtained Using DA, GD, and ML Algorithms for HME Function Design for the Boston Home Value Problem. K Is the Number of Leaves in teh Binary Tree

K	DA	GD	ML
4	5.7	5.9	7.5
8	3.4	3.6	5.6

Table 11 Comparison of Regression Error Obtained Using DA and GD Algorithms for NRBF Design for the Mortality Rate Prediction Problem. K Is the Number of Gaussian Basis Functions Used

K	DA	GD
1	3805.1	3805.1
2	1148.8	2154.0
4	720.8	1256.8
6	439.1	566.5
8	299.6	564.5
10	261.4	438.2

Table 12 Comparison of Regression Error Obtained Using DA, GD, and ML Algorithms for HME Design for the Mortality Rate Prediction Problem. K Is the Number of Leaves in the Binary Tree

K	DA	GD	ML
4	18.2	121.8	70.4
8	2.1	12.3	41.8

Table 13 Comparison of Regression Error Obtained Using DA and GD Algorithms for NRBF Design for the Fat Content Prediction Problem. K Is the Number of Gaussian Basis Functions Used. "TR" and "TE" Refer to Training and Test Sets, Respecively

	DA		GD	
K	TR	TE	TR	TE
1	159.9	168.2	159.9	168.2
2	52.9	58.8	131.4	159.7
4	28.6	32.9	119.8	138.0
6	27.3	40.1	74.9	83.7

[49]. Each function was used to generate both a training set and test set of size 225. We designed NRBF and HME regression estimates for each data set using both DA and the competitive design approaches. The results shown in Tables 15 and 16 show improved performance of the DA method that is consistent with the results obtained for the other benchmark sets.

IV. The Rate-Distortion Connection

Rate-distortion theory is the branch of information theory which is concerned with source coding. Its fundamental

Table 14 Comparison of Regression Error Obtained Using DA, GD, and ML Algorithms for HME Function Design for the Fat Content Prediction Problem. K Is the Number of Leaves in the Binary Tree. "TR" and "TE" Refer to Training and Test Sets, Respectively

	DA		GD		ML	
K	TR	TE	TR	TE	TR	TE
4	8.3	11.5	14.1	18.1	15.1	23.9
8	6.9	9.8	12.8	17.2	12.5	39.7

Table 15 Comparison of Regression Error Obtained by DA and GD on the Training (TR) and Test (TE) Sets, for NRBF Design to Approximate Functions, $f_1() \ldots f_5()$. K Denotes the Number of Basis Functions

Method	K	$f_1()$	$f_2()$	$f_3()$	$f_4()$	$f_5()$
DA(TR)	8	0.001	0.008	0.01	0.08	0.13
DA(TE)	8	0.001	0.009	0.01	0.09	0.13
GD(TR)	8	0.02	0.044	0.16	0.19	0.24
GD(TE)	8	0.02	0.049	0.17	0.17	0.23
DA(TR)	16	0.001	0.003	0.01	0.05	0.02
DA(TE)	16	0.001	0.005	0.01	0.05	0.03
GD(tr)	16	0.02	0.012	0.14	0.06	0.24
GD(te)	16	0.02	0.017	0.12	0.07	0.23

Table 16 Regression Error Obtained by DA, GD, and ML on the Training (TR) and Test (TE) Sets, for HME Design to Approximate Functions, $f_1() \ldots f_5()$. K Denotes the Number of Leaves in the Binary Tree

Method	K	$f_1()$	$f_2()$	$f_3()$	$f_4()$	$f_5()$
DA(TR)	4	0.0006	0.02	0.18	0.20	0.19
DA(TE)	4	0.0006	0.02	0.18	0.25	0.21
GD(TR)	4	0.0079	0.06	0.39	0.36	0.35
GD(TE)	4	0.0082	0.06	0.47	0.43	0.38
ML(TR)	4	0.026	0.08	0.86	0.36	0.43
ML(TE)	4	0.039	0.12	0.79	0.46	0.51
DA(TR)	8	0.0003	0.01	0.09	0.08	0.17
DA(TE)	8	0.0003	0.02	0.09	0.01	0.16
GD(TR)	8	0.0063	0.05	0.12	0.35	0.28
GD(TE)	8	0.0079	0.05	0.12	0.40	0.30
ML(TR)	8	0.011	0.03	0.12	0.09	0.32
ML(TE)	8	0.016	0.04	0.14	0.14	0.44

results are due to Shannon [95], [96]. These are the coding theorems which provide an (asymptotically) achievable bound on the performance of source coding methods. This bound is often expressed as an RD function $R(D)$ for a given source, whose curve separates the region of feasible operating points (R, D), from the region that cannot be attained by any coding system. Important extensions of the theory to more general classes of sources than those originally considered by Shannon have been developed since (see, e.g., [7] and [32]).

Explicit analytical evaluation of the function $R(D)$ has been generally elusive, except for very few examples of sources and distortion measures. Two main approaches were taken to address this problem. The first was to develop bounds on $R(D)$. An important example is the Shannon lower bound [96] which is useful for difference distortion measures. The second main approach was to develop a numerical algorithm, the Blahut–Arimoto (BA) algorithm [2], [11] to evaluate RD functions. The power of the second approach is in that the function can be

approximated arbitrarily closely at the cost of complexity. The disadvantage is that the complexity may become overwhelming, particularly in the case of continuous alphabets, and even more so for continuous vector alphabets where the complexity could grow exponentially with the dimensions. Another disadvantage is, of course, that no closed-form expression is obtained for the function, even if a simple one happens to exist.

We shall restrict our attention here to continuous alphabet sources. The RD curve is obtained by minimizing the mutual information subject to an average distortion constraint. Formally stated, given a continuous source alphabet $\mathcal{X}$, random variable X with a probability measure given by the density $p(x)$, and a reproduction alphabet $\mathcal{Y}$, the problem is of that optimizing the mutual information

$$I(X;Y) = H(Y) - H(Y|X) = \int dx p(x) \int dy p(y|x) \log \left[\frac{p(y|x)}{\int dx p(x) p(y|x)} \right] \tag{74}$$

over the random encoders $p(y|x)$, subject to

$$\int dx p(x) \int dy p(y|x)\, d(x, y) \leq D \tag{75}$$

where $d(\cdot, \cdot)$ is the distortion measure. By replacing the above minimization with parametric variational equations (see [7], [12], [32], or [40]), this problem can be reformulated as a problem of optimization over the reproduction density $p(y)$. The functional to be minimized is

$$F[p(y)] = -\tfrac{1}{\beta} \int dx p(x) \log \int dy p(y) e^{-\beta d(x, y)} \tag{76}$$

where β is a positive parameter that is varied to compute different points on the RD curve. But this criterion is easily recognizable as a continuous version of the free energy (25) we have developed in our (mass constrained) DA derivation! Much intuition can be obtained from this realization. In particular, the computation of the RD function is equivalent to a process of annealing; the effective reproduction alphabet is almost always discrete and grows via a sequence of phase transitions; and an efficient DA method can be used to compute the RD curve. Thus, the result is of importance to both rate-distortion theory and the basic DA approach itself. A detailed treatment of the relations between RD theory and DA is given in [87]. Here we only give a superficial outline.

To see more clearly the connection with the DA derivation we note that the objective of the optimization in (76) is to determine a probability measure on the reproduction space $\mathcal{Y}$. We may consider an alternative "mapping" approach which, instead of searching for the optimal $p(y)$ directly, searches for the optimal mapping y: $[0, 1] \rightarrow \mathcal{Y}$, where to the unit interval we assign the Lebesgue measure denoted by μ. The equivalence of the approaches is ensured by the theory of general measures in topological spaces (see for example [93, ch. 15] or [39, ch. 2 and 3]). We thus have to minimize the functional

$$F(y) = -\frac{1}{\beta} \int dx p(x) \log \int_{[0, 1]} d\mu(u) e^{-\beta\, d[x, y(u)]} \tag{77}$$

over the mapping $y(u)$. We replace direct optimization of a density defined over the reproduction space with mapping of "codevectors" with their probabilities onto this space. This is exactly what the mass-constrained DA method does.

Recall that in the basic DA derivation, at high temperature (small β), no matter how many codevectors are "thrown in" they all converge to a single point and are viewed as one effective codevector. In the RD case we have a "continuum of codevectors," yet it is easy to see that they all collapse on the centroid of the source distribution. The reproduction support (or effective alphabet) is therefore of cardinality one. Moreover, as we lower the temperature, the output remains discrete and its cardinality grows via a sequence of phase transitions exactly as we have seen in our treatment of DA for clustering. Using this approach, it was shown in [87] for the RD problem that the reproduction random variable X is continuous only if the Shannon lower bound (see e.g., [7]) is tight, which for the case of squared error distortion happens only when the source is Gaussian (or a sum of Gaussians). This is a surprising result in rate-distortion theory because the only analytically solved cases were exactly those where the Shannon lower bound is tight, which led to the implicit assumption that the optimal reproduction random variable is always continuous. (It should, however, be noted that the result was anticipated by Fix in an early paper [31] that, unfortunately, went relatively unnoticed.) From the DA viewpoint this is an obvious direct observation. It is summarized in a theorem [87].

Theorem 2: If the Shannon lower bound does not hold with equality, then the support of the optimal reproduction random variable consists of isolated singularities. Further, if this support is bounded (as is always the case in practice) then Y is discrete and finite.

For the practical problem of RD computation, we see two approaches, namely BA and DA, whose equivalence follows from the Borel Isomorphism Theorem. However, these approaches are substantially different in their computational complexity and performance if we need to discretize (as we always do). When using BA, discretization means defining a grid $\{y_i\}$ on the output space $\mathcal{Y}$. In DA we "discretize the unit interval" (i.e., replace it by a set of indexes) and induce an adaptive grid on $\mathcal{Y}$ by our mapping. Instead of a fixed grid in the output space, DA effectively optimizes a codebook of codevectors with their respective masses. This difference between the approaches is crucial because the output distributions are almost always discrete and finite. This gives DA the theoretical capability of producing exact solutions at finite model complexity, while BA can only approach exact solutions at the limit of infinite resolution. The mass-constrained DA algorithm (given in Section II) can be used to compute the RD curve [87].

It is a known result from RD theory that the parameter $\beta = 1/T$ as defined above is simply related to the slope of the (convex) RD curve

$$\beta = -\frac{dR}{dD}. \tag{78}$$

This gives a new interpretation of the DA approach to clustering and to the temperature parameter. The process of annealing is simply the process of RD computation which is started at maximum distortion and consists of "climbing" up the RD curve by optimally trading decrease in distortion for increase in rate. The position on the curve is determined by the temperature level which specifies the slope at this point. The process follows the RD curve as long as there are as many available codevectors as needed for the output cardinality. If the number of codevectors is *a priori* limited (as is the case in standard VQ design) then DA separates from, but attempts to stay as close as possible to, the RD curve after reaching the phase corresponding to the maximum allowed codebook size. Another important aspect of the annealing process, which is raised and demonstrated through the RD analysis, is the existence of two types of continuous phase transitions. One type is the cluster-splitting transition which we have analyzed and computed its critical temperature. The other kind is that of "mass growing," where a cluster is born first with zero mass and gradually gains in mass. The latter type of phase transition is more difficult to handle, and only preliminary results exist at this point. If, or when, we will be able to ensure that such phase transitions are always detected as well, we will have ensured that DA finds the global optimum. Note that, in practice, if a mass-growing phase transition is "missed" by the algorithm, this is often compensated by a corresponding splitting phase transition which occurs shortly afterwards, and optimality is regained.

V. Recent DA Extensions

In this section, a couple of recent extensions of the DA approach are briefly mentioned.

One important extension is to a method for the design of classifiers based on hidden Markov models, with obvious applications in speech recognition. Preliminary results for this work appeared in [81]. It is shown that DA can be applied to time sequences, and further, can be implemented efficiently by a forward-backward algorithm similar to the Baum–Welch reestimation algorithm [5]. The DA method allows joint optimization of the classifier components to directly minimize the classification error, rather than separate modeling of speech utterances via the maximum likelihood approach. Results so far [77], [80], [81] show substantial gains over standard methods. These preliminary results suggest that speech recognition may turn out to be the most important application of DA. Work in progress includes extensions to continuous speech, robustness of the classifier, etc.

Another advance was the application of DA to the problem of generalized vector quantization (GVQ). GVQ extends the VQ problem to handle joint quantization and estimation [35]. The GVQ observes random vector X and provides a quantized value for a statistically related, but unobservable, random vector Y. Of course, the special case $X = Y$ is the regular VQ problem. One typical application is in noisy source coding (often referred to as remote source coding in the information theory literature). Another application is concerned with the need to combine VQ with interpolation (e.g., when the vectors were down-sampled for complexity or other reasons). Preliminary results showing substantial gains due to the use of DA are given in [79].

VI. Summary

DA is a useful approach to clustering and related optimization problems. The approach is strongly motivated by analogies to statistical physics, but it is formally derived within information theory and probability theory. It enables escaping many poor local optima that plague traditional techniques without the slow schedules typically required by stochastic methods. The solutions obtained by DA are totally independent of the choice of initial configuration. The main objectives of this paper were: to derive DA from basic principles; to emphasize its generality; to illustrate its wide applicability to problems of supervised and unsupervised learning; and to demonstrate its ability to provide substantial gains over existing methods that were specifically tailored to the particular problem.

Most problems addressed were concerned with data assignment, via supervised or unsupervised learning, and the most basic of all is the problem of clustering. A probabilistic framework was constructed by randomization of the partition, which is based on the principle of maximum entropy at a given level of distortion, or equivalently, minimum expected distortion at a given level of entropy. The Lagrangian was shown to be the Helmholtz free energy in the physical analogy, and the Lagrange multiplier T is the temperature. The minimization of the free energy determines isothermal equilibrium and yields the solution for the given temperature. The resulting association probabilities are Gibbs distributions parameterized by T. Within this probabilistic framework, annealing was introduced by controlling the Lagrange multiplier T. This annealing is interpreted as gradually trading entropy of the associations for reduction in distortion. Phase transitions were identified in the process which are, in fact, cluster splits. A sequence of phase transitions produces a hierarchy of fuzzy-clustering solutions. Critical temperatures T_c for the onset of phase transitions were derived. At the limit of zero temperature, DA converges to a known descent method, the GLA or K-means, which in standard implementations is arbitrarily or heuristically initialized. Consistent substantial performance gains were obtained.

The method was first extended to a variety of related unsupervised learning problems by incorporating constraints on the clustering solutions. In particular, DA methods were derived for noisy-channel VQ, entropy constrained VQ, and structurally constrained VQ design. Additional constraints may be applied to address graph-theoretic problems.

A highly significant extension is to supervised learning problems. The DA approach was rederived while allowing the imposition of structures on the partition, and while optimizing the ultimate optimization cost. This extension enables the DA approach to optimize complicated discrete costs for a large variety of popular structures. The method's performance was demonstrated on the problem of classification with the vector quantizer, radial basis functions, and multilayer perceptron structures, and on the problem of regression with the VQ, hierarchical mixture of experts, and normalized radial basis functions. For each one of the examples, the DA approach significantly outperformed standard design methods that were developed for the specific structure.

The relations to information theory, and in particular to RD theory, were discussed. It was shown that the DA method for clustering is equivalent to the computation of the RD function. This observation led to contributions to rate-distortion theory itself, and to further insights into the workings of DA.

A couple of extensions, which are currently under investigation, were briefly introduced. One extension is to the design of hidden Markov model-based classifiers. This work extends DA to handle time sequences and is directly applicable to the important problem of speech recognition. Another extension is concerned with the problem of generalized vector quantizer design.

REFERENCES

[1] H. Akaike, "A new look at statistical model identification," *IEEE Trans. Automat. Contr.*, vol. 19, pp. 713–723, Dec. 1974.

[2] S. Arimoto, "An algorithm for calculating the capacity of an arbitrary discrete memoryless channel," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 14–20, Jan. 1972.

[3] E. Ayanoglu and R. M. Gray, "The design of joint source and channel trellis waveform coders," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 855–865, Nov. 1987.

[4] G. Ball and D. Hall, "A clustering technique for summarizing multivariate data," *Behav. Sci.*, vol. 12, pp. 153–155, Mar. 1967.

[5] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains," *Ann. Math. Statist.*, vol. 41, no. 1, pp. 164–171, 1970.

[6] G. Beni and X. Liu, "A least biased fuzzy clustering method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 954–960, Sept. 1994.

[7] T. Berger, *Rate Distortion Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[8] ——, "Minimum entropy quantizers and permutation codes," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 149–157, Mar. 1982.

[9] J. C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 1–8, Jan. 1980.

[10] G. L. Bilbro, W. E. Snyder, S. J. Garnier, and J. W. Gault, "Mean field annealing: A formalism for constructing GNC-like algorithms," *IEEE Trans. Neural Networks*, vol. 3, pp. 131–138, Jan. 1992.

[11] R. E. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 460–473, July 1972.

[12] ——, *Principles and Practice of Information Theory*. Reading, MA: Addison-Wesley, 1987.

[13] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, (Wadsworth Statistics/Probability Series). Belmont, CA: Wadsworth, 1980.

[14] J. Buhmann and H. Kuhnel, "Vector quantization with complexity costs," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1133–1145, July 1993.

[15] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based on vector quantization," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. 28, pp. 562–574, Oct. 1980.

[16] P.-C. Chang and R. M. Gray, "Gradient algorithms for designing predictive vector quantizers," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-34, pp. 679–690, Aug. 1986.

[17] V. Cherkassky, Y. Lee, and H. Lari-Najafi, "Self-organizing network for regression: Efficient implementation and comparative evaluation," in *Proc. Int. Joint Conf. Neural Networks*, 1991, vol. 1, pp. 79–84.

[18] P. A. Chou, "Optimal partitioning for classification and regression trees," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 340–354, Apr. 1991.

[19] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-37, pp. 31–42, Jan. 1989.

[20] ——, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inform. Theory*, vol. 35, pp. 299–315, Mar. 1989.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.

[22] R. D. Dony and S. Haykin, "Neural network approaches to image compression," *Proc. IEEE*, vol. 83, pp. 288–303, Feb. 1995.

[23] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1974.

[24] J. G. Dunham and R. M. Gray, "Joint source and channel trellis encoding," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 516–519, July 1981.

[25] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1974.

[26] R. Durbin, R. Szeliski, and A. Yuille, "An analysis of the elastic net approach to the travelling salesman problem," *Neural Computation*, vol. 1, no. 3, pp. 348–358, 1989.

[27] R. Durbin and D. Willshaw, "An analogue approach to the travelling salesman problem using an elastic net method," *Nature*, vol. 326, pp. 689–691, 1987.

[28] N. Farvardin, "A study of vector quantization for noisy channels," *IEEE Trans. Inform. Theory*, vol. 36, pp. 799–809, July 1990.

[29] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-Gaussian memoryless sources," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 485–497, May 1984.

[30] N. Farvardin and V. Vaishampayan, "On the performance and complexity of channel-optimized vector quantizers," *IEEE Trans. Inform. Theory*, vol. 37, pp. 155–160, Jan. 1991.

[31] S. L. Fix, "Rate distortion functions for squared error distortion measures," in *Proc. 16th Annu. Allerton Conf. Communications Control and Computers*, Oct. 1978, pp. 704–711.

[32] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[33] D. Geiger and F. Girosi, "Parallel and deterministic algorithms from MRFs: Surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 401–412, May 1991.

[34] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 721–741, Nov. 1984.

[35] A. Gersho, "Optimal nonlinear interpolative vector quantizers," *IEEE Trans. Commun.*, vol. COM-38, pp. 1285–1287, Sept. 1990.

[36] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1991.

[37] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 377–388, Apr. 1996.

[38] T. Graepel, M. Burger, and K. Obermayer, "Phase transitions in stochastic self-organizing maps," *Phys. Rev. E*, vol. 56, no. 4, pp. 3876–3890, 1997.

[39] R. M. Gray, *Probability, Random Processes, and Ergodic Properties*. New York: Springer-Verlag, 1988.

[40] ——, *Source Coding Theory*. Boston, MA: Kluwer, 1990.

[41] R. M. Gray and E. D. Karnin, "Multiple local minima in vector quantizers," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 256–261, Mar. 1982.

[42] B. Hajek, "A tutorial survey of theory and applications of simulated annealing," in *Proc. 24th IEEE Conf. Decision and Control*, 1985, pp. 755–760.
[43] D. Harrison and D. L. Rubinfeld, "Hedonic prices and the demand for clean air," *J. Environ. Economics and Management*, vol. 5, pp. 81–102, 1978.
[44] J. A. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.
[45] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Englewoods Cliffs, NJ: Prentice-Hall, 1998.
[46] G. E. Hinton and M. Revow, "Using pairs of data points to define splits for decision trees," *Neural Inform. Processing Syst.*, vol. 8, pp. 507–513, 1995.
[47] T. Hofmann and J. M. Buhmann, "Pairwise data clustering by deterministic annealing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 1–14, Jan. 1997.
[48] Y. H. Hu, S. Palreddy, and W. J. Tompkins, "Customized ECG beat classifier using mixture of experts," in *Proc. IEEE Workshop Neural Networks for Signal Processing*, 1995, pp. 459–464.
[49] J.-N. 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, pp. 342–353, May 1994.
[50] R. A. Jacobs and M. I. Jordan, "Learning piecewise control strategies in a modular neural network architecture," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 337–345, Mar./Apr. 1993.
[51] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.
[52] E. T. Jaynes, "Information theory and statistical mechanics," in *Papers on Probability, Statistics and Statistical Physics*, R. D. Rosenkrantz, Ed. Dordrecht, The Netherlands: Kluwer, 1989.
[53] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.
[54] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[55] M. Kloppenburg and P. Tavan, "Deterministic annealing for density estimation by multivariate normal mixtures," *Phys. Rev. E*, vol. 55, no. 3, pp. 2089–2092, 1997.
[56] T. Kohonen, G. Barna, and R. Chrisley, "Statistical pattern recognition with neural networks: Benchmarking studies," in *Proc. IEEE Int. Conf. Neural Networks*, 1988, vol. I, pp. 61–68.
[57] H. Kumazawa, M. Kasahara, and T. Namekawa, "A construction of vector quantizers for noisy channels," *Electron. Commun. Japan*, vol. 67, no. 4, pp. 39–47, 1984.
[58] A. Kurtenbach and P. Wintz, "Quantizating for noisy channels," *IEEE Trans. Commun.*, vol. COM-17, pp. 291–302, Apr. 1969.
[59] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.
[60] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 129–137, Mar. 1982.
[61] S. P. Luttrell, "Hierarchical vector quantization (image compression)," *Proc. Inst. Elect. Eng. I (Commun. Speech and Vision)*, vol. 136, no. 6, 1989, pp. 405–413.
[62] ——, "Derivation of a class of training algorithms," *IEEE Trans. Neural Networks*, vol. 1, pp. 229–232, June 1990.
[63] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statistics and Probability*, 1967, 281–297.
[64] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "'Neural-gas' network for vector quantization and its application to time-series prediction," *IEEE Trans. Neural Networks*, vol. 4, pp. 558–569, July 1993.
[65] J. Max, "Quantizing for minimum distortion," *IRE Trans. Inform. Theory*, vol. IT-6, pp. 7–12, Mar. 1960.
[66] G. C. McDonald and R. C. Schwing, "Instabilities of regression estimates relating air pollution to mortality," *Technometrics*, vol. 15, pp. 463–482, 1973.
[67] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Application to Clustering*. New York: Marcel Dekker, 1988.
[68] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1091, 1953.
[69] D. Miller, "An information-theoretic framework for optimization with applications in source coding and pattern recognition," Ph.D. dissertation, Univ. California, Santa Barbara, 1995.
[70] D. Miller, A. V. Rao, K. Rose, and A. Gersho, "A global optimization technique for statistical classifier design," *IEEE Trans. Signal Processing*, vol. 44, pp. 3108–3122, Dec. 1996.
[71] D. Miller and K. Rose, "An improved sequential search multistage vector quantizer," in *Proc. IEEE Data Computers Conf.*, 1993, pp. 12–21.
[72] ——, "Combined source-channel vector quantization using deterministic annealing," *IEEE Trans. Commun.*, vol. 42, pp. 347–356, Feb. 1994.
[73] ——, "Hierarchical, unsupervised learning with growing via phase transitions," *Neural Computation*, vol. 8, no. 2, pp. 425–450, 1996.
[74] D. Miller, K. Rose, and P. A. Chou, "Deterministic annealing for trellis quantizer and HMM design using Baum–Welch re-estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1994, vol. V, pp. 261–264.
[75] J. Moody and C. J. Darken, "Fast learning in locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.
[76] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels, "On the training of radial basis function classifiers," *Neural Networks*, vol. 5, no. 4, pp. 595–604, 1992.
[77] A. V. Rao, "Design of pattern recognition systems using deterministic annealing: Applications in speech recognition, regression, and data compression," Ph.D. dissertation, Univ. California, Santa Barbara, 1998.
[78] A. Rao, D. Miller, K. Rose, and A. Gersho, "A deterministic annealing approach for parsimonious design of piecewise regression models," submitted for publication.
[79] ——, "A generalized VQ method for combined compression and estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1996, vol. IV, pp. 2032–2035.
[80] A. Rao, K. Rose, and A. Gersho, "Design of robust HMM speech recognizer using deterministic annealing," in *Proc. IEEE Workshop Automatic Speech Recognition and Understanding*, Santa Barbara, CA, Dec. 1997, pp. 466–473.
[81] ——, "A deterministic annealing approach to discriminative hidden Markov model design," in *Proc. IEEE Workshop Neural Networks in Signal Processing*, Amelia Island, FL, Sept. 1997, pp. 266–275.
[82] A. Rao, D. Miller, K. Rose, and A. Gersho, "Mixture of experts regression modeling by deterministic annealing," *IEEE Trans. Signal Processing*, vol. 45, pp. 2811–2820, Nov. 1997.
[83] B. D. Ripley, "Neural networks and related methods for classification," *J. Roy. Stat. Soc., Ser. B*, vol. 56, no. 3, pp. 409–456, Nov. 1994.
[84] E. A. Riskin and R. M. Gray, "A greedy tree growing algorithm for the design of variable rate vector quantizers," *IEEE Trans. Signal Processing*, vol. 39, pp. 2500–2507, Nov. 1991.
[85] J. Rissanen, "Stochastic complexity and modeling," *Ann. Statist.*, vol. 14, pp. 1080–1100, 1986.
[86] K. Rose, "Deterministic annealing, clustering, and optimization," Ph.D. dissertation, California Inst. Technol., Pasadena, 1991.
[87] ——, "A mapping approach to rate-distortion computation and analysis," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1939–1952, Nov. 1994.
[88] K. Rose, E. Gurewitz, and G. C. Fox, "A deterministic annealing approach to clustering," *Pattern Recognition Lett.*, vol. 11, no. 9, pp. 589–594, 1990.
[89] ——, "Statistical mechanics and phase transitions in clustering," *Phys. Rev. Lett.*, vol. 65, no. 8, pp. 945–948, 1990.
[90] ——, "Vector quantization by deterministic annealing," *IEEE Trans. Inform. Theory*, vol. 38, pp. 1249–1257, July 1992.
[91] ——, "Constrained clustering as an optimization method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 785–794, Aug. 1993.
[92] K. Rose and D. Miller, "Constrained clustering for data assignment problems with examples of module placement," in *Proc. IEEE Int. Symp. Circuits and Systems*, San Diego, CA, May 1992, pp. 1937–1940.
[93] H. L. Royden, *Real Analysis*, 3rd ed. New York: Macmillan, 1988.
[94] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Parallel*

Distributed Processing. Cambridge, MA: MIT, 1986.
[95] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423, 623–656, 1948.
[96] ——, "Coding theorems for a discrete source with a fidelity criterion," *IRE Nat. Conv. Rec.*, pt. 4, pp. 142–163, 1959.
[97] J. E. Shore and R. W. Johnson, "Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy," *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 26–37, Jan. 1980.
[98] P. D. Simic, "Statistical mechanics as the underlying theory of elastic and neural optimization," *Network*, vol. 1, no. 1, pp. 89–103, 1990.
[99] ——, "Constrained nets for graph matching and other quadratic assignment problems," *Neural Computation*, vol. 3, no. 2, pp. 268–281, 1991.
[100] H. Szu and R. Hartley, "Nonconvex optimization by fast simulated annealing," *Proc. IEEE*, vol. 75, pp. 1538–1540, Nov. 1987.
[101] L. Tarassenko and S. Roberts, "Supervised and unsupervised learning in radial basis function classifiers," *Proc. Inst. Elect. Eng.-Visual Image Signal Processing*, vol. 141, no. 4, pp. 210–216, 1994.
[102] D. M. Titterington, A. F. M. Smith, and U. E. Makov, *Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.
[103] N. Ueda and R. Nakano, "Deterministic annealing variant of the EM algorithm," in *Proc. Neural Information Processing Systems*, Nov. 1994, pp. 545–552.
[104] S. R. Waterhouse and A. J. Robinson, "Non-linear prediction of acoustic vectors using hierarchical mixtures of experts," in *Proc. Neural Inform. Processing Syst.*, vol. 7, pp. 835–842, 1994.
[105] A. S. Weigend, M. Mangeas, and A. N. Srivastava, "Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting," *Int. J. Neural Syst.*, vol. 6, no. 4, pp. 373–399, 1995.
[106] Y.-F. Wong, "Clustering data by melting," *Neural Computation*, vol. 5, no. 1, pp. 89–104, 1993.
[107] E. Yair, K. Zeger, and A. Gersho, "Competitive learning and soft competition for vector quantizer design," *IEEE Trans. Signal Processing*, vol. 40, pp. 294–309, Feb. 1992.
[108] A. L. Yuille, "Generalized deformable models, statistical physics, and matching problems," *Neural Computation*, vol. 2, no. 1, pp. 1–24, 1990.
[109] K. Zeger and A. Gersho, "Vector quantizer design for memoryless noisy channels," in *Proc. IEEE Int. Conf. Communications*, Philadelphia, 1988, pp. 1693–1597.
[110] J. Zhao and J. Shawe-Taylor, "Neural network optimization for good generalization performance," in *Proc. Int. Conf. Artificial Neural Networks*, 1994, pp. 561–564.

Kenneth Rose (Member, IEEE) received the B.Sc. (summa cum laude) and M.Sc. (magna cum laude) degrees in electrical engineering from Tel-Aviv University, Israel, in 1983 and 1987, respectively, and the Ph.D. degree in electrical engineering from the California Institute of Technology, Pasadena, 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, image transmission through noisy channels, and general image processing. In January 1991 he joined the Department of Electrical and Computer Engineering, University of California at 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 co-recipient of the IEEE Communications Society's William R. Bennett Prize Paper Award (1990).