A MAXIMUM ENTROPY APPROACH FOR OPTIMAL STATISTICAL CLASSIFICATION *

David Miller, Ajit Rao, Kenneth Rose, and Allen Gersho
Center for Information Processing Research
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106
e-mail:rose@ece.ucsb.edu

Abstract

A global optimization technique is introduced for statistical classifier design to minimize the probability of classification error. The method, which is based on ideas from information theory and analogies to statistical physics, is inherently probabilistic. During the design phase, data are assigned to classes *in probability*, with the probability distributions chosen to maximize entropy subject to a constraint on the expected classification error. This entropy maximization problem is seen to be equivalent to a free energy minimization, motivating a deterministic annealing approach to minimize the misclassification cost. Our method is applicable to a variety of classifier structures, including nearest prototype, radial basis function, and multilayer perceptron-based classifiers. On standard benchmark examples, the method applied to nearest prototype classifier design achieves performance improvements over both the learning vector quantizer, as well as over multilayer perceptron classifiers designed by the standard back-propagation algorithm. Remarkably substantial performance gains over learning vector quantization are achieved for complicated mixture examples where there is significant class overlap.

*This work was supported in part by the National Science Foundation under grant no. NCR-9314335, the University of California MICRO program, Rockwell International Corporation, Hughes Aircraft Company, Echo Speech Corporation, Signal Technology Inc., Lockheed Missile and Space Company, and Qualcomm, Inc.

0-7803-2739-X/95 $4.00 © 1995 IEEE

1 Introduction

In recent years, the tremendous growth in neural networks research has stimulated renewed interest in statistical classification. Structures such as the multilayer perceptron (MLP) have the capability of implementing complex decision boundaries, and have been demonstrated to perform well in comparison with conventional classifiers, both for engineering applications such as speech recognition [8], as well as in the context of scientific inquiry [14]. However, several researchers have observed that MLPs and other structures trained to minimize the distance to output classification levels ($\{0,1\}$ for the two-class case) do not directly minimize the classification error rate. Instead, these networks approximate the Bayes-optimal discriminant function, or equivalently the *a posteriori* probabilities that observations belong to a given class, e.g. [13]. (Similar observations have been made for linear classifiers [2]). Clearly, very large networks may be able, in principle, to accurately implement the Bayes rule, and thus provide minimum classification error. However, practical classifiers have restricted size to avoid high complexity and overfitting of limited training data. Thus, in practice, approximating the optimal discriminant function may result in significantly greater classification error than alternative solutions.

Rather than choosing to approximate the discriminant function, a number of researchers have proposed alternative cost objectives and learning algorithms which better match the goal of minimizing misclassification error (or minizing risk, if errors are not weighed equally), e.g. [7],[4],[6],[11]. Typically, these methods descend on an energy surface, using either a batch or a sequential optimization technique. While these approaches optimize MLPs and other network models to effectively minimize classification error, a legitimate concern is the potential to fall into poor local minimum traps, which often riddle the energy surface. In fact, the problem of local optima in neural networks has been acknowledged in a number of papers, e.g. [14]. While some smart heuristics have been employed for initializing parameters, typically one is forced to generate solutions based on a large number of random initializations, and then choose the best result.

We propose a new deterministic learning algorithm for statistical classifier design with a demonstrated potential for avoiding local optima of the cost. Several deterministic, annealing-based techniques have been proposed for avoiding nonglobal optima in computer vision [18],[3], combinatorial optimization [1], and elsewhere. Our approach is derived based on ideas from information theory and statistical physics, and builds on the framework of the deterministic annealing (DA) approach to clustering and related problems [16][15][17]. DA's probabilistic framework for clustering was derived by applying the maximum entropy principle to determine the underlying distributions. In recent work [9], we have shown that the maximum entropy approach unifies a larger class of optimization methods than was originally conceived, and moreover, can be used to develop new, effective optimization methods for a number of challenging problems in source coding and statistics [12]. The maximum entropy formulation is useful because it precisely characterizes the annealing process in these methods as a gradual lowering of both the entropy and cost of the system with decreasing "temperature", where the temperature parameter is a Lagrange multiplier used to control the level of average cost.

Here, this DA approach is extended to minimize the cost of misclassification. We thus provide an approach for designing statistical classifiers based on training data which avoids many local minima that trap other known methods. In the next section, the method is derived and interpreted, and then we present some simulation comparisons with classifiers designed using conventional techniques. While in the sequel we assume a nearest prototype classifier structure for concreteness, we emphasize here that our approach is, in fact, generally applicable to optimize a variety of classification structures, including MLPs. A more general derivation and results for other structures can be found in [10].

2 Derivation and Algorithm

Let $\mathcal{T} = \{(\mathbf{x}, c)\}$ be a training set of N labelled vectors, where $\mathbf{x} \in \mathcal{R}^n$ is a feature vector and $c \in \mathcal{I}$ is its class label from an index set $\mathcal{I}$. A classifier is a mapping $C : \mathcal{R}^n \longrightarrow \mathcal{I}$, which assigns a class label in $\mathcal{I}$ to each vector in $\mathcal{R}^n$. A training pair $(\mathbf{x}, c) \in \mathcal{T}$ is *misclassified* if $C(\mathbf{x}) \neq c$. The performance measure of the classifier is the probability of error, i.e. the fraction of the training set that it misclassifies. Our ultimate objective is to design a classifier to minimize this cost. In this paper, we restrict C to be a nearest prototype classifier, representable by a set of vectors $\{\mathbf{x}_{jk}\} \subset \mathcal{R}^n$, where $\mathbf{x}_{jk}$ is the kth prototype associated with class $j \in \mathcal{I}$. The classifier maps a vector in $\mathcal{R}^n$ to the class associated with the nearest prototype, defining a partition of $\mathcal{R}^n$ into regions $R_j \equiv \bigcup_k C_{jk}$ where $C_{jk} \equiv \{\mathbf{x} \in \mathcal{R}^n : d(\mathbf{x}, \mathbf{x}_{jk}) \leq d(\mathbf{x}, \mathbf{x}_{lm}) \;\; \forall l, m\}$. Here, $d(\cdot, \cdot)$ is the "distance measure" used for classification.

Due to the challenging nature of the classifier design problem, we adopt a DA approach for its solution. Unlike simulated annealing, which implements a sequence of stochastic "moves" on the cost surface, DA is a deterministic learning algorithm that replaces stochastic operations with expectations over the distribution. Accordingly, we cast the problem in a probabilistic framework and consider a "random" classifier characterized by a probabilistic assignment of features to classes. We define the *probability of association* between a feature $\mathbf{x}$ and subregion C_{jk}, $P[\mathbf{x} \in C_{jk}]$, as well as the probability of association with a class region, $P[\mathbf{x} \in R_j] \equiv \sum_k P[\mathbf{x} \in C_{jk}]$. As our method will optimize over these probabilities in choosing the classifier, the distributions must be consistent with the formation of a nearest prototype classification rule. This structural restriction may be enforced via a well-chosen parametrization of the distribution. An appropriate choice is the Gibbs distribution,

$$P[\mathbf{x} \in C_{jk}] = \frac{e^{-\gamma d(\mathbf{x},\mathbf{x}_{jk})}}{\sum\limits_{l,m} e^{-\gamma d(\mathbf{x},\mathbf{x}_{lm})}}, \tag{1}$$

which depends on the prototype vectors and on a scale parameter γ which controls the fuzziness of the distribution. As $\gamma \longrightarrow \infty$, the association probabilities revert to hard classifications equivalent to application of the nearest prototype rule. Note that this choice can be directly obtained using the principle of maximum entropy, which provides stronger justification for the resulting optimization method [10]. However, for conciseness we omit this derivation.

In our approach, we simultaneously control the probability of error and the randomness of the classifier. We start with a highly random classifier with a high expected classification error rate and then gradually reduce both the randomness and the expected probability of error. A natural measure of randomness is Shannon's entropy. In fact, Jaynes [5] showed that while there may be infinitely many distributions which satisfy expected value constraints, the least biased distribution is that which maximizes entropy. For the classification problem, the expected value of interest is the average classification error $< P_e >$. Thus, the maximum entropy distribution $\{P[\mathbf{x} \in C_{jk}]\}$ associated with the classification problem is obtained by solving

$$\max_{\{\mathbf{x}_{jk}\},\gamma} H \equiv \max_{\{\mathbf{x}_{jk}\},\gamma} \{-\frac{1}{N} \sum_{(\mathbf{x},c)\in\mathcal{T}} \sum_{j} P[\mathbf{x} \in R_j] \log P[\mathbf{x} \in R_j]\} \tag{2}$$

subject to

$$< P_e >= \frac{1}{N} \sum_{(\mathbf{x},c)\in\mathcal{T}} \sum_{j} P[\mathbf{x} \in R_j]\rho(c,j).$$

Here the cost of misclassification is $\rho(c,j) = 1$ if $c \neq j$ and 0 otherwise. Effectively, entropy maximization over the distribution is achieved through optimization over its parameter set. Solving this problem is equivalent to solving the unconstrained minimization of the Lagrangian:

$$\min_{\{\mathbf{x}_{jk}\},\gamma} L \equiv \min_{\{\mathbf{x}_{jk}\},\gamma} \beta < P_e > -H, \tag{3}$$

where β is the Lagrange multiplier used to enforce a constraint on $< P_e >$. For $\beta = 0$, the sole objective is entropy maximization, which is achieved by the uniform distribution, choosing the prototype vectors to be non-distinct. For $\beta \longrightarrow \infty$, minimizing L is equivalent to minimizing the probability of error P_e, leading to a non-random (i.e. $H \rightarrow 0$) classifier. This solution can be obtained within our probabilistic framework by choosing all prototype vectors to be distinct and sending $\gamma \rightarrow \infty$. Thus, we observe that an annealing approach is naturally obtained by minimizing the Lagrangian starting from $\beta = 0$ and tracking the solution while increasing β towards infinity. In this way, we obtain a sequence of solutions of decreasing entropy and cost, leading to a "hard" classifier at the limit. The annealing process can avoid local optima of the cost, and is motivated by the physical interpretation of the Lagrangian as a Helmholtz free energy [9]. We can rewrite the Lagrangian explicitly as:

$$L = \frac{1}{N} \sum_{(\mathbf{x},c)\in\mathcal{T}} \sum_{j} P[\mathbf{x} \in R_j] \left(\beta\rho(c,j) + \log P[\mathbf{x} \in R_j]\right) \qquad (4)$$

$$\equiv \frac{1}{N} \sum_{(\mathbf{x},c)\in\mathcal{T}} (\sum_{j} P[\mathbf{x} \in R_j] L_{xj}) \equiv \frac{1}{N} \sum_{(\mathbf{x},c)\in\mathcal{T}} L_x .$$

Here, parentheses identify L_{xj}, the contribution to the cost when the feature $\mathbf{x}$ is assigned to class j, and L_x, the average contribution for this feature. The necessary conditions for minimizing L at any β are :

$$\frac{\partial L}{\partial \mathbf{x}_{jk}} = \frac{2\gamma}{N} \sum_{(\mathbf{x},c)\in\mathcal{T}} (L_{xj} - L_x) P[\mathbf{x} \in C_{jk}] \frac{\partial d(\mathbf{x}, \mathbf{x}_{jk})}{\partial \mathbf{x}_{jk}} = 0, \quad \forall j,k \qquad (5)$$

and

$$\frac{\partial L}{\partial \gamma} = \frac{1}{N} \sum_{(\mathbf{x},c)\in\mathcal{T}} \sum_{j} L_{xj} (P[\mathbf{x} \in R_j] v_x - v_{xj}) = 0. \qquad (6)$$

Here v_x is the average distance from $\mathbf{x}$ to a prototype, i.e. $v_x = \sum_j \sum_k P[\mathbf{x} \in C_{jk}] d(\mathbf{x}, \mathbf{x}_{jk})$, and v_{xj} is the contribution to this average from the prototypes of R_j, i.e. $v_{xj} = \sum_k P[\mathbf{x} \in C_{jk}] d(\mathbf{x}, \mathbf{x}_{jk})$.

These conditions can be interpreted, appropriately, within the context of supervised learning. The condition for a prototype vector suggests moving it away from (towards) vectors that it "owns" probabilistically through $P[\mathbf{x} \in C_{jk}]$ and for which the cost L_{xj} incurred by classifying to region R_j is greater than (less than) the average cost. The optimality condition for the scale parameter γ leads to a similar interpretation. Essentially, γ is either increased to solidify ownership of a point by a region if the cost is small, or is decreased to weaken ownership of a point if the cost is large. The optimization at each β can be implemented by gradient descent or any other function minimization technique. For $\beta \rightarrow \infty$, $H \rightarrow 0$ and $< P_e > \rightarrow P_e$. At this limit, our method terminates satisfying the necessary optimality conditions.

3 Results

We have performed experimental comparisons of our nearest-prototype method with the learning vector quantizer (LVQ) [7]. As an example, consider the two-class data of Figure 4. Each class consists of a Gaussian mixture with three components. We designed prototype-based classifiers with three prototypes per class, using both the LVQ and DA optimization methods. LVQ solutions were generated using the public domain LVQ-pak software, running both an optimized LVQ (OLVQ) learning phase, as well as a fine-tuning phase with 500,000 iterations. The learning parameter μ was set to 0.03. Ten LVQ solutions were generated based on random initialization and in all cases the method was unable to discriminate the class 0 "minority" component in the upper right of Figure 4a (which retains only 5 % of the training set mass). Apparently, the initialization did not select a prototype from the class 0 minority component, and LVQ is unable to move class 0 prototypes through the "wall" of class 1 data which separates them from this component. The best LVQ solution, which is shown in Figure 4a, achieved $P_e = 7.7\%$. Increasing the number of prototypes, we found that LVQ was only able to discriminate the minority component when 21 prototypes per class were introduced, and in this case the method achieved $P_e = 3.4\%$. The extremity of this sub-optimality does suggest that the LVQ-pak initialization could be improved. For example, if an initialization of prototypes based on Isodata clustering followed by allocation of prototypes to the majority class of the cluster were used, much fewer than 42 prototypes (but greater than six) would suffice to find good solutions. However, this example does demonstrate LVQ's susceptibility to finding poor solutions. In fact, we also performed gradient descent on $< P_e >$ and found that poor solutions were obtained in this case as well - excepting omnisicient initialization in the vicinity of the optimal solution, the best performance obtained for six prototypes was $P_e = 7.0\%$. It thus appears that strict descent methods will fail on this example unless given an excellent initialization. By contrast, the DA method using only five prototypes achieved the solution shown in Figure 1b, with $P_e = 2.7\%$. Note that the DA method is independent of the initialization, placing all prototypes together at the global data centroid (marked by X) at $\beta = 0$ so as to maximize entropy. (Such an initialization is in fact "fatal" for a strict descent-based approach, as the associated learning rule will not permit a class 0 prototype to pass through the "wall" of class 1 data.) Then, as β is increased, the prototypes separate, reducing the entropy as well as $< P_e >$. This example demonstrates the ability of the method to avoid local optima, since the DA optimization does succeed in moving a class 0 prototype from X directly *through* the class 1 data "wall" to correctly classify the minority class 0 component and achieve what appears to be the optimal piece-wise linear result. (Here, two of the class 0 prototypes are non-distinct, so the solution effectively uses five prototypes.)

In addition to this example, we have tested our approach on the "synthetic" example from [14], as well as on some other complicated synthetically generated mixture examples. On the example from [14], our approach achieved $P_e = 8.9\%$ on the test set using eight prototypes and $P_e = 8.6\%$ using twelve prototypes, in comparison to LVQ's $P_e = 9.5\%$ based on twelve prototypes. For general reference, an MLP with six hidden units achieved $P_e = 9.4\%$. For complicated mixture examples, with possibly twenty or more overlapping mixture components and multiple classes, we have found our method to consistently achieve substantial peformance gains over LVQ. As an example, we generated training data for a four-class problem involving twenty-four overlapping, non-isotropic mixture components in two dimensions. We designed nearest prototype classifiers with 16 prototypes (four per class) using both LVQ and DA. The best LVQ solution based on ten random initializations achieved $P_e = 31\%$. By contrast the single DA solution achieved $P_e = 23\%$. This comparison is typical of what we have seen through extensive experimentation. Similar performance gains are achieved for higher-dimensional data sets, but we have restriced these examples to two dimensions for visual illustration. While for certain problems other structures such as MLPs or RBFs may be inherently superior to the prototype-based structure discussed here, our results demonstrate the potential of the optimization technique. Moreover, as we describe in [10], our method achieves similar performance gains in optimizing the RBF and MLP classifier structures.

References

[1] G. L. Bilbro, W. E. Snyder, and R. C. Mann. Mean-field approximation minimizes relative entropy. *Journal of the Opt. Soc. of Amer.*, 8:290–294, 1991.

[2] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, New York, NY, 1974.

[3] D. Geiger and F. Girosi. Parallel and deterministic algorithms from MRFs: Surface reconstruction. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 13:401–412, 1991.

[4] J. B. Hampshire and A. H. Waibel. A novel objective function for improved phoneme recognition using time-delay neural networks. *IEEE Trans. on Neural Net.*, 1:216–228, 1990.

[5] E. T. Jaynes. Information theory and statistical mechanics. In R. D. Rosenkrantz, editor, *Papers on probability, statistics and statistical physics.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1989. (Reprint of the original 1957 papers in *Physical Review*).

[6] B.-H. Juang and S. Katagiri. Discriminative learning for minimum error classification. *IEEE Trans. on Sig. Proc.*, 40:3043–3054, 1992.

[7] T. Kohonen, G. Barna, and R. Chrisley. Statistical pattern recognition with neural networks: Benchmarking studies. In *IEEE Proc. ICNN*, volume 1, pages 61–68, 1988.

[8] R. Lippmann. Review of neural networks for speech recognition. *Neural Comp.*, 1:1–39, 1989.

[9] D. Miller, A. Rao, K. Rose, and A. Gersho. A maximum entropy framework for optimization with application to supervised learning. (Submitted for publication.), 1994.

[10] D. Miller, A. Rao, K. Rose, and A. Gersho. An information-theoretic framework for optimal statistical classification. (Submitted for publication.), 1995.

[11] V. Nedeljkovic. A novel multilayer neural networks training algorithm that minimizes the probability of classification error. *IEEE Trans. on Neural Net.*, 4:650–659, 1993.

[12] A. Rao, D. Miller, K. Rose, and A. Gersho. Generalized vector quantization. (In preparation.), 1994.

[13] M. D. Richard and R. P. Lippmann. Neural network classifiers estimate Bayesian a posteriori probabilities. *Neural Comp.*, 3:461–483, 1991.

[14] B. D. Ripley. Neural networks and related methods for classification. *Journal of the Royal Stat. Soc., Ser. B*, 56:409–456, 1994.

[15] K. Rose, E. Gurewitz, and G. C. Fox. Statistical mechanics and phase transitions in clustering. *Phys. Rev. Lett.*, 65:945–948, 1990.

[16] K. Rose, E. Gurewitz, and G. C. Fox. Vector quantization by deterministic annealing. *IEEE Trans. on Inform. Theory*, 38:1249–1258, 1992.

[17] K. Rose, E. Gurewitz, and G. C. Fox. Constrained clustering as an optimization method. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 15:785–794, 1993.

[18] A. L. Yuille. Generalized deformable models, statistical physics, and matching problems. *Neural Comp.*, 2:1–24, 1990.

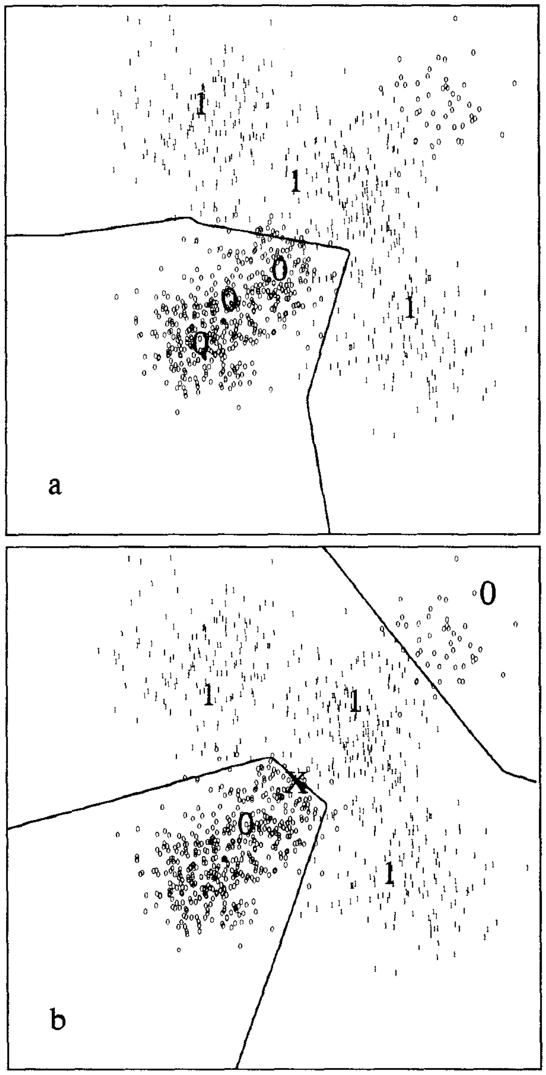

Figure 1: A two-class example, with a 3-component Gaussian mixture in each class: a) The LVQ solution, using three prototypes per class, with $P_e = 7.7\%$. b) The DA solution, using three prototypes per class, with $P_e = 2.7\%$. Note that since the solution at $\beta = 0$ placed all prototypes at the global centroid (X), the DA optimization has allowed a prototype for class 0 to "pass through a wall" of class 1 data in order to correctly classify the minority "0" mixture component.