NEAREST-PROTOTYPE CLASSIFIER DESIGN BY DETERMINISTIC ANNEALING WITH RANDOM CLASS LABELS*

Ertem Tuncel and Kenneth Rose
Signal Compression Laboratory
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106, USA
E-mail: ertem@laurel.ece.ucsb.edu, rose@ece.ucsb.edu

Abstract.

The design of nearest-prototype (NP) classifiers is a challenging problem because of the prevalence of poor local minima, and the piecewise constant nature of the cost function which is incompatible with gradient-based techniques. This paper extends the deterministic annealing (DA) method for NP-classifier design in two ways. First, the association between prototypes and class labels is also randomized, and the corresponding association probabilities are added to the set of parameters to be optimized. Second, the multiplicity (or the mass) of prototypes are optimized. During the design, all parameters are optimized so as to minimize the expected misclassification rate for a given level of randomness. The *joint entropy*, which measures the level of randomness, is gradually reduced while optimizing the cost Lagrangian. As the entropy approaches zero, the method seeks a deterministic classifier that minimizes the rate of misclassification.

INTRODUCTION

The nearest-prototype (NP) classifier is an efficient tool for classifying data, that has the capability of implementing *nonconvex* or even *disconnected* partitions which are determined by a small number of parameters, namely, the prototype locations. Moreover, although the NP-classifier decision regions are restricted to be unions of Voronoi cells, any classifier can be approximated by NP-classifiers, with a sufficient number of prototypes.

*THIS WORK WAS SUPPORTED IN PART BY THE NATIONAL SCIENCE FOUNDATION UNDER GRANT NO. NCR-9314335, THE UNIVERSITY OF CALIFORNIA MICRO PROGRAM, CISCO SYSTEMS, INC., DIALOGIC CORP., FUJITSU LABORATORIES OF AMERICA, INC., GENERAL ELECTRIC CO., HUGHES NETWORK SYSTEMS, INTEL CORP., LERNOUT & HAUSPIE SPEECH PRODUCTS, LOCKHEED MARTIN, LUCENT TECHNOLOGIES, INC., QUALCOMM, INC., ROCKWELL INTERNATIONAL CORP., PANASONIC TECHNOLOGIES, INC., AND TEXAS INSTRUMENTS, INC.

A known difficulty with the design of NP classifiers is that the optimal labeling, or allocation of prototypes to classes, is not known a priori. In practice, the allocation is made in a heuristic fashion. For example, prototype labels may be distributed among classes uniformly, or according to the class label distribution in the training set. Such heuristic initialization often results in suboptimal distribution of prototypes and compromises the ultimate performance. Known algorithms, such as the learning vector quantizer (LVQ) design [1], as well as the existing deterministic annealing (DA) approach [2], suffer from this weakness.

In this paper, this problem is tackled by a significant extension of the DA algorithm of [2], which itself builds on the DA approach to clustering [3] (see also [4] for a tutorial). In the DA algorithm proposed here, the classifier randomization is extended to include the assignment of labels to prototypes. In other words, each prototype is assigned to a class label *in probability.* The randomness is controlled by monitoring the *joint entropy* of the system.

NP CLASSIFIER FORMULATION AND DESIGN

In general, a classifier is defined [5] as a mapping $C : \mathcal{R}^m \rightarrow \mathcal{K}$, where $\mathcal{R}^m$ is the feature space and $\mathcal{K}$ is any index set of labels. It induces a partitioning of the feature space into regions $R_k \equiv \{x \in \mathcal{R}^m : C(x) = k\}$, where $\bigcup_k R_k \equiv \mathcal{R}^m$ and $\bigcap_k R_k \equiv \emptyset$.

NP Classification

A *nearest-prototype* classifier is a composite mapping $C = Q \circ V$ where $V : \mathcal{R}^m \rightarrow \mathcal{J}$ is the partitioning function and $Q : \mathcal{J} \rightarrow \mathcal{K}$ is the labeling function, and where $\mathcal{J}$ is an intermediate (prototype) index set. The partition function is

$$V(x) = \arg\min_{j \in \mathcal{J}} \{d(x, s_j)\}\,, \tag{1}$$

where, $S = \{s_j : j \in \mathcal{J}\}$ is the set of *prototypes* and $d(\cdot, \cdot)$ is any appropriately defined *distance or distortion measure* on $\mathcal{R}^m$. Each vector $x \in \mathcal{R}^m$ is first mapped to the nearest prototype $s_{V(x)}$, and then classified to the corresponding label by $Q(\cdot)$. Accordingly, each classifier region R_k is the union of Voronoi cells:

$$\begin{aligned} R_k &\equiv \bigcup_{j : Q(j) = k} V_j \text{ with} \\ V_j &\equiv \{x \in \mathcal{R}^m : V(x) = j\}\,. \end{aligned} \tag{2}$$

Let $\mathcal{T} = \{(x_n, c_n)\}$ be a training set of N pairs of vectors consisting of $x_n \in \mathcal{R}^m$ and corresponding class labels $c_n \in \mathcal{K}$. The performance of a classifier C is measured by the empirical error rate

$$P_e(C) = \frac{1}{N} \sum_{n=1}^{N} \rho(c_n, C(x_n))\,, \tag{3}$$

where $\rho(c_n, k) = 1$ if $c_n \neq k$ and 0 otherwise. For reasons that will soon become obvious, it is more convenient to rewrite $P_e(C)$ as

$$P_e(C) = \frac{1}{N} \sum_{n} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} v_{nj} q_{jk} \rho(c_n, k)\,, \tag{4}$$

where

$$\begin{aligned} v_{nj} &= \begin{cases} 1 & \text{if } V(x_n) = j\,, \\ 0 & \text{otherwise}\,. \end{cases} \\ q_{jk} &= \begin{cases} 1 & \text{if } Q(j) = k\,, \\ 0 & \text{otherwise}\,. \end{cases} \end{aligned} \tag{5}$$

Design with Deterministic Annealing

In the DA approach for NP-classifier design [2], $Q(\cdot)$ was fixed as a deterministic mapping, and the mapping $V(\cdot)$ was randomized. In other words, points in the feature space $\mathcal{R}^m$ were assigned to prototypes *in probability* during the design phase. More specifically,

$$P[x_n \in V_j] = \frac{e^{-\gamma d(x_n, s_j)}}{\sum_i e^{-\gamma d(x_n, s_i)}}\,, \tag{6}$$

where γ is a parameter governing the "peakiness" of the distribution. The classifier output was declared as the fixed-label of the chosen prototype. A "randomness level" of $V(\cdot)$, measured by the *Shannon entropy*

$$H = -\frac{1}{N} \sum_{n} \sum_{j \in \mathcal{J}} P[x_n \in V_j] \log P[x_n \in V_j]\,, \tag{7}$$

was imposed on the solution, and this level was gradually decreased. At the limit of zero entropy, a "hard" classifier is obtained, in which case $\gamma \to \infty$, and hence $V(\cdot)$ goes to (1). At each intermediate level of randomness, the objective was to minimize the misclassification rate over the training set, which can still be expressed as in (4) with the modification $v_{nj} = P[x_n \in V_j]$, subject to the entropy constraint. From a different perspective, the same objective could be stated as "to maximize the entropy while gradually lowering the level of misclassification rate."

It was shown [2] with several synthetic and "real-world" examples that the fixed-label DA algorithm outperforms LVQ-based classification [1]. However, the open question of how $Q(\cdot)$ should be fixed, i.e., how many prototypes should be associated with each class, remained unresolved. In fact, there is no analytical way to determine the optimum $Q(\cdot)$ prior to the actual classifier design. There are several known *ad hoc* ways, such as distributing the prototype labels among classes uniformly, or according to the frequency of the class labels in the training set. Each of these heuristics has its limitations. We next propose a remedy for this shortcoming.

Random-label DA

We proceed to randomize $Q(\cdot)$ as well. We redefine q_{jk} as the *probability* that prototype j belongs to class k, so that the misclassification rate is still given by (4). However, if we reinterpret the meaning of $\mathcal{J}$ as the index set for *distinct* prototypes, and denote the number of *identical* prototypes located at a distinct position by λ_j, the overall misclassification rate becomes

$$P_e(C) = \frac{1}{N} \sum_n \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \lambda_j v_{nj} q_{jk} \rho(c_n, k)\,. \tag{8}$$

We also optimize λ_j as done for clustering in [6] and [4], to obtain the *correct* multiplicity at each distinct prototype. We do not require λ_j to be an integer, so it may be viewed as the *mass* of prototype s_j, where the total mass to be distributed among prototypes is M.

The randomized composite mapping $C = Q(V(\cdot))$ is in fact a two-stage Markov chain, whose overall entropy is

$$\begin{aligned} H &= -\frac{1}{N} \sum_n \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \lambda_j v_{nj} q_{jk} \log\left(v_{nj} q_{jk}\right) \\ &= -\frac{1}{N} \sum_n \sum_{j \in \mathcal{J}} \lambda_j v_{nj} \log v_{nj} - \frac{1}{N} \sum_n \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \lambda_j v_{nj} q_{jk} \log q_{jk}\,. \end{aligned} \tag{9}$$

With this modification we follow a derivation similar to that of [2], and consider the expected *structural cost* induced by this random partitioning of the feature space:

$$D = \frac{1}{N} \sum_n \sum_{j \in \mathcal{J}} \lambda_j v_{nj} d(x_n, s_j)\,. \tag{10}$$

Our objective is to minimize the misclassification rate (8) where $H = H^*$ is given as a constraint. So an equivalent objective is the following:

$$\begin{array}{rl} \text{Minimize} & L = \beta P_e - H \\ \text{with respect to} & \gamma, s_j, \lambda_j, \text{and } q_{jk} \\ \text{subject to} & \displaystyle\sum_{k \in \mathcal{K}} q_{jk} = 1\ , \forall j \in \mathcal{J} \\ & \displaystyle\sum_{j \in \mathcal{J}} \lambda_j = M \\ & v_{nj} = \dfrac{e^{-\gamma d(x_n, s_j)}}{\sum_{i \in \mathcal{J}} \lambda_j e^{-\gamma d(x_n, s_i)}}\ , \end{array} \tag{11}$$

where β is the "reciprocal temperature" controlling the joint entropy level. Initially $\beta = 0$, in which case the sole objective becomes the maximization of the joint entropy, which is achieved by $\gamma = 0$ and an arbitrary set of identical prototypes. When $\beta \to \infty$, the objective becomes the minimization of the misclassification rate, which is, of course, our ultimate objective. This minimum is achieved only by $H = 0$, i.e., $\gamma \to \infty$ and all prototypes are distinct. It is easy to see from (11), that in this case the classifier is a hard NP-classifier. When β is gradually increased from 0 to ∞, the prototypes undergo a sequence of prototype splits, or *bifurcations*, which correspond to *phase transitions* in the statistical physics analogy. One significant problem with the above formulation as it stands is that it allows too much freedom in the location of prototypes until β reaches the critical value for the second phase transition. To be more specific,

- When $0 \leq \beta < \beta_1$, that is, before the first phase transition, any set of identical prototypes will minimize the cost function.
- When $\beta_1 \leq \beta < \beta_2$, that is, between the first and second phase transitions, the line on which the two sets of prototypes should be is unique, but the distance between the prototypes and the value of γ still maintain a degree of freedom. It is possible to increase one and decrease the other without changing the value of the Lagrangian cost function.

The excessive freedom is very undesirable and increases the likelihood of trapping the algorithm in a poor local minimum. Instead, we propose a regularized objective, namely

$$\begin{aligned}
\text{Minimize} \quad & L = \beta P_e + \gamma D - H \\
\text{with respect to} \quad & \gamma, s_j, \lambda_j, \text{and } q_{jk} \\
\text{subject to} \quad & \sum_{k \in \mathcal{K}} q_{jk} = 1 \ , \forall j \in \mathcal{J} \\
& \sum_{j \in \mathcal{J}} \lambda_j = M \\
& v_{nj} = \frac{e^{-\gamma d(x_n, s_j)}}{\sum_{i \in \mathcal{J}} \lambda_j e^{-\gamma d(x_n, s_i)}} \, .
\end{aligned} \tag{12}$$

It can be shown that as $\beta \to \infty$, not only $\gamma \to \infty$, but also $\frac{\gamma}{\beta} \to 0$, which is consistent with the original aim of minimizing the misclassification rate P_e. The excessive degrees of freedom are eliminated by the addition of the expected structural cost term. For example, when $\beta = 0$, the set of identical prototypes must be at the global centroid of the training set $\mathcal{T}$ (assuming the squared-distance measure $d(x, y) = ||x - y||^2$.)

The objective (12) is a convex optimization problem in terms of q_{jk}. So the solution for q_{jk} is *analytically* found as

$$q_{jk} = \frac{e^{-\beta D_{jk}}}{\sum_m e^{-\beta D_{jm}}} \, , \tag{13}$$

where

$$D_{jk} = \frac{\sum_n v_{nj} \rho(c_n, k)}{\sum_n v_{nj}} \,, \tag{14}$$

which is easily interpreted as the expected misclassification rate given that prototype s_j is associated with class label k. Substituting q_{jk} back in the Lagrangian of (12), we obtain

$$L = -\sum_n \log\,(\sum_{j \in \mathcal{J}} \lambda_j e^{-\gamma d(x_n, s_j)}) - \sum_n \sum_{j \in \mathcal{J}} \lambda_j v_{nj} \log\,(\sum_{k \in \mathcal{K}} e^{-\beta D_{jk}})\,, \tag{15}$$

which is a natural generalization of the *free energy* (or strictly speaking the *potential*) derived in [3] (and [6] with the mass constraint on codevectors) for the case of clustering or vector quantization.

Necessary Conditions for Optimality

At each value of β, the solution of (15), i.e., the optimal locations of the prototypes $\{s_j\}$, their masses $\{\lambda_j\}$, and γ should satisfy

$$\nabla_{s_j} L = 0 = \gamma \sum_n \lambda_j v_{nj} \nabla_{s_j} d(x_n, s_j)(1 + L_n - L_{nj})\,, \tag{16}$$

and

$$\frac{\partial L}{\partial \gamma} = 0 = \sum_n \sum_{j \in \mathcal{J}} \lambda_j v_{nj} d(x_n, s_j)(1 + L_n - L_{nj})\,, \tag{17}$$

and

$$\frac{N}{M} = \sum_n v_{nj}(1 + L_n - L_{nj})\,, \tag{18}$$

where

$$L_{nj} = \beta \sum_{k \in \mathcal{K}} q_{jk} \{\rho(c_n, k) - D_{jk}\} - \log\,(\sum_{k \in \mathcal{K}} e^{-\beta D_{jk}}) - \log\,(\sum_{i \in \mathcal{J}} \lambda_i e^{-\gamma d(x_n, s_i)}) \tag{19}$$

and

$$L_n = \sum_{i \in \mathcal{J}} \lambda_i v_{ni} L_{ni}\,. \tag{20}$$

Note also that

$$L = \sum_n \sum_{j \in \mathcal{J}} \lambda_j v_{nj} L_{nj} = \sum_n L_n\,. \tag{21}$$

EXPERIMENTAL RESULTS

We performed experiments to compare the performances of the LVQ method [1], the fixed-label DA method [2], and the proposed random-label DA method. As a training set, we used the Finnish phoneme data set that accompanies the standard LVQ software package. The training set consists of 1962 vectors which represent 20-dimensional cepstral coefficients of the corresponding phoneme uttered by the speaker. There are 20 classes (phonemes) in the training set. The experiments comparing LVQ and the fixed-label DA method have already appeared in [2]. In both LVQ and fixed-label DA approaches, the number of prototypes that are associated with a particular class is proportional to the relative frequency of occurrence of that class in the training set. The random-label DA algorithm, circumvents the need for such an *ad hoc* decision.

M (# of prototypes)	20	30	40	50	80	100
P_e (LVQ)	13.25	12.44	11.47	10.96	10.09	8.17
P_e (fixed-label DA)	11.67	9.99	8.36	5.55	4.83	4.23
P_e (random-label DA)	7.65	7.18	6.88	5.86	4.63	4.17

Table 1: A comparison of P_e values (in percent) for the NP-classifiers designed by LVQ, fixed-label DA, and random-label DA methods. The training set is the 20-dimensional, 20 class Finnish phoneme data set.

The experiments are performed using various values of total number of prototypes. The results are presented in Table 1. As seen from the table, in most cases, the random-label DA algorithm outperforms both the fixed-label DA and the LVQ algorithms. The achieved performance gain is more significant when the total number of prototypes is small. For example, the 20-prototype classifier designed by the random-label DA method is even better than the 40-prototype classifier designed by the fixed-label DA, and 100-prototype classifier designed by LVQ. So, one can achieve either substantially improved performance at the same classifier complexity, or similar performance at substantially reduced complexity. Although it is known that deterministic annealing methods avoid many poor local minima, there is no guarantee of finding the global minimum. This is indeed the case of 50 prototypes, where the fixed-label DA performs slightly better than the random-label DA.

CONCLUSIONS

This paper presents an extension of fixed-label DA, where all mappings, including the mapping from prototypes to class labels, are randomized, and the level of randomness is measured by the *joint entropy* which is gradually reduced to zero.

The results of the experiments on the Finnish phoneme set indicate that further significant gains, in terms of better classification performance or reduced classifier complexity, are achieved by randomization and optimization of the mapping $Q(\cdot)$ between the prototypes and the class labels.

REFERENCES

[1] T. Kohonen, G. Barna and R. Chrisley, “Statistical pattern recognition with neural networks: Benchmarking studies”, IEEE Proc. ICNN, vol. 1, pp 61-68, 1988.

[2] D. Miller, A. V. Rao, K. Rose and A. Gersho, “A Global Optimization Technique for Statistical Classifier Design”, IEEE Trans. on Signal Processing, vol. 44, pp 3108-3122, 1996.

[3] K. Rose, E. Gurewitz and G. C. Fox, “Vector Quantization by Deterministic Annealing”, IEEE Trans. on Information Theory, vol. 38, pp 1249-1257, 1992.

[4] K. Rose, “Deterministic annealing for clustering, compression, classification, regression, and related optimization problems”, Proceedings of IEEE, vol. 86, no. 11, pp 2210-2239, 1998.

[5] R. O. Duda and P. E. Hart, “Pattern Classification and Scene Analysis”, John Wiley & Sons, 1973.

[6] K. Rose, E. Gurewitz and G. C. Fox, “Constrained Clustering as an Optimization Method”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 15, no. 8, pp 785-794, 1993.