LETTER Communicated by Carsten Peterson

Natural Discriminant Analysis Using Interactive Potts Models

Jiann-Ming Wu
jmwu@server.am.ndhu.edu.tw
Department of Applied Mathematics, National Donghwa University, Shoufeng, Hualien 941,Taiwan, Republic of China

Natural discriminant analysis based on interactive Potts models is developed in this work. A generative model composed of piece-wise multivariate gaussian distributions is used to characterize the input space, exploring the embedded clustering and mixing structures and developing proper internal representations of input parameters. The maximization of a log-likelihood function measuring the fitness of all input parameters to the generative model, and the minimization of a design cost summing up square errors between posterior outputs and desired outputs constitutes a mathematical framework for discriminant analysis. We apply a hybrid of the mean-field annealing and the gradient-descent methods to the optimization of this framework and obtain multiple sets of interactive dynamics, which realize coupled Potts models for discriminant analysis. The new learning process is a whole process of component analysis, clustering analysis, and labeling analysis. Its major improvement compared to the radial basis function and the support vector machine is described by using some artificial examples and a real-world application to breast cancer diagnosis.

1 Introduction

The task of discriminant analysis (Hastie & Simard 1998; Hastie & Tibshirani 1996; Hastie, Tibshirani, & Buja, 1994; Hastie, Buja, & Tibshirani, 1995) aims to achieve a mapping function from a parameter space R^d to a set of discrete and nonordered output labels S subject to interpolating conditions proposed by training samples $\{(x_i, q_i), 1 \leqq i \leqq N, x_i \in R^d, q_i \in S\}$. S is represented by $\{\mathbf{e}_1^M, \mathbf{e}_2^M, \ldots, \mathbf{e}_M^M\}$ for the discrete and nonordered property of output labels, where M denotes the number of categories and $\mathbf{e}_m^M$ is a unitary vector of M elements with only the mth bit one. The category of an item is predicted by d measurements of features $x \in R^d$. The following design cost quantitatively measures the fitness of all training samples to a mapping function g: $R^d \to S$,

$$D = \sum_{1 \leqq i \leqq N} L(g(x_i), q_i), \tag{1.1}$$

where $L(\cdot)$ denotes an arbitrary distance between two unitary vectors. A mapping function absolutely minimizing the design cost automatically satisfies all interpolating conditions proposed by training samples; however, such a mapping function can be obtained by simply recording all samples in a look-up table. It is not the purpose of supervised learning, since without any other objective, a learning process subject only to the design cost is an ill-posed problem. Coming from a future validation by testing samples, the generalization cost proposes another essential criterion. Both the training set and the testing set are assumed to have the same underlying input-output relation. An effective supervised learning process is expected to minimize the design cost as well as the generalization cost. The minimization of this generalization cost has been considered as the reduction of model size in the field of statistics (Hastie & Simard, 1998; Hastie & Tibshirani, 1996; Hastie et al., 1994, 1995), which is used for maximal generalization in this work.

The derivation of our new learning process starts with a generative model responsible for characterizing the parameter space on the basis of a nonoverlapping partition. Assume that there exist K internal regions $\Omega_k, 1 \leqq k \leqq K$, in the partition; each region Ω_k is centered at y_k and is defined by $\Omega_k = \{x | \arg\min_j \|x - y_j\|_{A_k} = k, x \in R^d\}$, where $\|x\|_A$ denotes the Mahalanobis distance of $\sqrt{x' Ax}$. The local distribution of the input parameter in each region Ω_k is modeled by a multivariate gaussian distribution with a mean vector at the center y_k and a covariance matrix A_k. As a special case, all local generative models are assumed to have the same covariance A in this work to facilitate our presentations. The kernels $\{y_k\}$ partition the space R^d into K nonoverlapping subspaces with the property of $\bigcup_k \Omega_k = R^d$ and $\Omega_{k_1} \bigcap \Omega_{k_2} = \emptyset$ for all $k_1 \neq k_2$. The fitness of all parameters in each internal region Ω_k to the corresponding local generative model is quantitatively measured by a log-likelihood function. In this work, the supervised learning process for discriminant analysis is a process of maximizing the sum of all log-likelihood functions and minimizing the design cost.

This work insists on solving the task of discriminant analysis by collective decisions performed by the architecture of neural networks. The formulation of the above two objectives involves two kinds of variables: discrete combinatorial variables and continuous geometrical variables. The resulting optimization framework is a mixed integer and linear programming, of which the optimization is difficult for the gradient-descent method due to numerous shallow local minimum within the corresponding energy function. The Potts encoding, which possesses flexibility in internal representations and reliability in collective decisions, is employed to deal with this computational difficulty. The Potts encoding is suitable for the design of neural networks and has been applied to fundamental complex tasks, including combinatorial optimizations (Peterson & Söderberg, 1989), self-organization (Liou & Wu, 1996; Rose, Gurewitz, & Fox, 1990, 1993), classification, and regression (Rao, Miller, Rose, & Gersho, 1999). The multistate

Potts neuron, generalized from the two-state spin neuron, is used to reduce the search space of feasible configurations and realize the problem modeling for effective internal representations. In this work, the combinatorial internal representations include the assignment of each input parameter x_i to one and only one internal region, denoted by the membership vector $\delta_i \in \{e_1^K, \ldots, e_K^K\}$, and the dynamical assignment of each region Ω_k to one output label, denoted by the category response $\xi_k \in \{e_1^M, \ldots, e_M^M\}$. Each δ_i or ξ_k is considered as the discrete Potts neural variable. The continuous geometrical variables include the kernels $\{y_k\}$ and the common covariance matrix A. By these representations, the maximization of the sum of all log-likelihood functions, each measuring the finesse of the corresponding local generative model, and the minimization of the design cost together form a mixed integer and linear programming and lead to a novel energy function for discriminant analysis. All these variables—$\{\delta_i\}$, $\{\xi_k\}$, $\{y_k\}$, and A—are collectively optimized by a hybrid of the mean-field annealing and the gradient-descent method toward the minimization of the energy function. The resulting learning process consists of four sets of interactive dynamics characterizing the coupled Potts models of discriminant analysis.

The evolution of the four sets of interactive dynamics is well controlled by an annealing process for the minimization of the energy function. The annealing process is analogous to physical annealing, which is a process of gradually and carefully scaling the temperature from a sufficiently large scale to a small one. At each temperature, the mean configuration of the whole system is a balancing result of trading off minimizing the mean energy against maximizing the entropy. When this process is used, mean activations of a Potts neuron, indicating probabilities of active states, are increasingly influenced by injected mean fields. At the beginning, mean activations are independent of injected mean fields; the system is ruled over by the principle of maximal entropy; a Potts neuron has almost the same probability of activating each of its states. As the process progresses, the symmetry is broken; each Potts neuron has a decreasing degree of freedom; the mean configuration of the system is increasingly dominated by the tendency toward the minimal mean energy and decreasingly by the criterion of maximal entropy. Toward the end of the process, the mean configuration is totally controlled by the force of minimal mean energy; mean activations of a Potts neuron behave winner-take-all. The Potts encoding has been applied to model collective decisions (Liou & Wu, 1996; Peterson & Söderberg, 1989). Its applicability to the task of discriminant analysis is explored in this work.

1.1 The Learning Network. The learning process is composed of four sets of interactive dynamics, which constitute the coupled Potts model in architecture. The coupled Potts model is a modular recurrent neural network

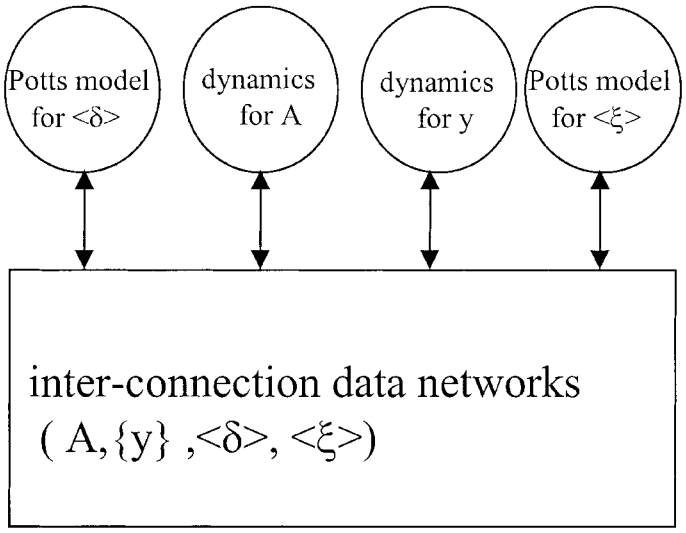

Figure 1: The learning network is composed of four interactive dynamics and the interconnection network.

for supervised learning. As shown in Figure 1, the coupled Potts model consists of four interactive modules, of which two are Potts neural networks calculating the mean $\langle\delta\rangle$ and $\langle\xi\rangle$ of combinatorial variables $\{\delta_i\}$ and $\{\xi_k\}$, respectively, and the others are linear networks for updating kernels $\{y_k\}$ and the covariance matrix A. Four interactive modules communicate with each other through interconnection networks. The same learning network has appeared in implementing the recurrent backpropagation (Pineda, 1987, 1989) with two modules.

1.2 The Discriminant Network. The discriminant network derived by the new learning process is closely related to a network of multilayer perceptrons (MLP) (Rumelhart & McClelland, 1986; Pineda, 1987, 1989) and radial basis functions (RBF) (Benaim, 1994; Freeman & Saad, 1995; Girosi, Jones, & Poggio, 1995; Girosi, 1998; Moody & Darken, 1989). It is a network of normalized radial basis functions (RBFs) with generalized hidden units. The hidden units of a normalized RBF network (Moody & Darken, 1989) use a normalized gaussian activation function,

$$G_k^I(x) = \frac{\exp(\|x - y_k\|^2/2\sigma^2)}{\sum_j \exp(\|x - y_j\|^2/2\sigma^2)}, \tag{1.2}$$

with a scalar variance σ^2. The hidden units of the current recognizing network have a normalized multivariate gaussian activation function,

$$G_k^A(x) = \frac{\exp(-\beta(x-y_k)'A(x-y_k))}{\sum_j \exp(-\beta(x-y_j)'A(x-y_j))}, \tag{1.3}$$

where A is a covariance matrix and β is the inverse of an artificial temperature used in the annealing process. By normalization, we mean that $\sum_j G_j^A(x) = 1$ for any x. $G_k^I(x)$ is a special case of $G_k^A(x)$, whereas the function $G_k^A(x)$ can be translated to the form of $G_k^I(z)$ if one rewrites the term $(x-y_k)'A(x-y_k)$ as $(z-z_k)'(z-z_k)$ with $z_k = By_k$ and $z = Bx$, where $B'B = A$. By this translation, the function $G_k^A(x)$ is decomposed as the combination of $z = Bx$ and $G_k^I(z)$. The current recognizing network is exactly the composition of a linear transformation and a normalized RBF network. The learning process derived in this work can be directly applied to a normalized RBF network by fixing the covariance matrix as I, and it is also applicable to an MLP network based on the connection (Girosi et al., 1995; Girosi, 1998) between a normalized RBF network and an MLP network. Practical experiments (Miller & Uyar, 1998) have shown the gradient-descent-based learning algorithms, including the backpropagation algorithm for an MLP network and the learning algorithm (Moody & Darken, 1988, 1989) for an RBF network, suffering at the trap of tremendous local minima in optimizing their internal representations. Based on a hybrid of the mean-field annealing and the gradient-descent method, the new learning process proposed in this work is essential for developing effective nonlinear boundaries, well optimizing the internal representation for the parameter space of a real application.

The normalized multivariate gaussian activation function in equation 1.3 defines an overlapping partition into the parameter space, where the overlapping degree is modulated by the β parameter, indirectly by the annealing process. We consider the function $G_k^A(x)$ as the projection probability of assigning an input parameter x to an internal region Ω_k. For an input parameter x, all its projection probabilities $\{G_k^A(x), 1 \leqq k \leqq K\}$ characterize different partition phases; at a sufficiently low β, they are nearly identical to $\frac{1}{K}$, denoting a complete overlapping partition. As the β value increases, they become asymmetric for some degree of overlapping partition. To a sufficiently large β, they behave winner-take-all, such that the only winner $G_{k^*}^A(x)$ is one and the others are zero, where $k^* = \arg\min_k \|x - y_k\|_A$. If each region Ω_k is equipped with an optimal category response ξ_k, based on the above projection mechanism, the mapping function of the current recognizing network is

$$g(x) = \sum_k G_k^A(x)\,\xi_k. \tag{1.4}$$

This discriminant function at a sufficiently large β is similar to the nearest prototype classifier, but with a generalized distance measure $\|\cdot\|_A$,

$$g(x) = \xi_{k^*}, \tag{1.5}$$
$$k^* = \arg\min_k \left\|x - y_k\right\|_A,$$

and can be further translated to a composition of

$$g(z) = \xi_{k^*} \tag{1.6}$$
$$k^* = \arg\min_k \|z - z_k\|$$
$$\text{and} \quad z = Bx, \tag{1.7}$$

where $z_k = By_k$ and $A = B'B$. If $A = I$, the latter form exactly defines the nearest prototype classifier (Rao et al., 1999), of which the measure is the Euclidean distance. The partition of $\{z_k\}$ into the parameter space is nonoverlapping, and each internal region is attached with its own category response. The nearest prototype classifier is known to be suitable for the case with statistically independent components, but for most real applications, this assumption is not valid, so a preprocessor for feature extraction like the above linear transformation, $z = Bx$, is usually additionally employed. But the development of a preprocessor, such as using independent component analysis (Lin, Grier, & Cowan, 1997; Makeig, Jung, & Bell, 1997) or principal component analysis, is independent of the formation of a classifier; the combined discriminant function may suffer from the inconsistency between the extracted feature and the classifier. Alternatively, based on the Mahalanobis distance, the new learning process in this work focuses on the mapping function in equation 1.5 and directly explores the whole discriminant process of component analysis, clustering analysis, and labeling analysis.

In the next section, we introduce the generative model for characterizing the parameter space and derive a mathematical framework for discriminant analysis. Four sets of interactive dynamics and the coupled Potts model are developed in section 3. Another issue in this work is incremental learning, a procedure for determining the optimal number of internal regions or the minimal model size for maximal generalization. The incremental learning scheme is introduced in section 4. In the final section, we test the new method in comparisons with RBF (Muller et al., 1999; Ratsch, Onoda, & Muller, 2001) and the support vector machine (Vapnik, 1995; Platt, 1999; Cawley, 2000) using artificial examples and a real-world application to breast cancer diagnosis (Wolberg & Mangasarian, 1990; Malini Lamego, 2001), and discussions about simulation results are described.

2 Supervised Learning for Discriminant Analysis

2.1 The Generative Model. The generative model for a parameter space R^d is composed of K piece-wise multivariate normal distributions. Each is of

$$p_k(x) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{|A^{-1}|}} \exp\left(-\frac{1}{2}(x - y_k)'A(x - y_k)\right) \tag{2.1}$$

centered at the vector y_k with a nonsingular covariance matrix A. The K distributions have been assumed to have the same covariance matrix. These kernels $\{y_k\}$ form K internal regions $\{\Omega_k\}$ in the parameter space, which are nonoverlapping, such as $\bigcup_k \Omega_k = R^d$ and $\Omega_{k_1} \bigcap \Omega_{k_2} = \emptyset$ for all $k_1 \neq k_2$. For each internal region Ω_k, the fitness of $p_k(x)$ to all input parameters $x_i \in \Omega_k$ is measured proportional to the following log-likelihood function:

$$l_k = \log \prod_{x_i \in \Omega_k} p_k(x_i). \tag{2.2}$$

A summation of all l_k leads to the following function:

$$\begin{aligned} l &= \sum_k l_k \\ &= \sum_k \log \prod_{x_i \in \Omega_k} p_k(x_i) \\ &= \sum_k \sum_{x_i \in \Omega_k} \log p_k(x_i). \end{aligned} \tag{2.3}$$

Recall that the assignment of each input x_i to one of K internal regions has been represented by a membership vector δ_i, of which each element δ_{ik} is either one or zero and $\sum_k \delta_{ik} = 1$. The function l can be rewritten as

$$\begin{aligned} l &= \sum_i \sum_k \delta_{ik} \log p_k(x_i) \\ &= -\frac{1}{2} \sum_i \sum_k \delta_{ik}(x_i - y_k)'A(x_i - y_k) \\ &\quad -\frac{N}{2} \log \det(A^{-1}) - \frac{Nd}{2} \log(2\pi), \end{aligned} \tag{2.4}$$

where $\det(\cdot)$ denotes the determinant of a matrix. By the fact $\det(A^{-1}) = -\det(A)$ and neglecting the last constant term, we obtain the following objective:

$$E_1 = \frac{1}{2} \sum_i \sum_k \delta_{ik}(x_i - y_k)'A(x_i - y_k) - \frac{N}{2} \log \det(A). \tag{2.5}$$

Maximizing the function l is equivalent to minimizing the function E_1.

2.2 A Mathematical Framework. Recall that each region Ω_k has been attached with a category response $\xi_k \in \{e_1^M, \ldots, e_M^M\}$ for classification. The design cost in equation 1.1 can be expressed as

$$\begin{aligned} E_2 &= \frac{1}{2}\sum_i ||q_i - \sum_k \delta_{ik}\xi_k||^2 \\ &= \frac{1}{2}\sum_i ||q_i - \Lambda\delta_i||^2, \end{aligned} \tag{2.6}$$

where the matrix $\Lambda = [\xi_1, \ldots, \xi_k, \ldots, \xi_K]$. By combining the two objectives of equations 2.5 and 2.6 and injecting all constraints, we have the following mathematical framework for discriminant analysis.

Minimize

$$E(\delta, \xi, y, A) = E_1 + cE_2 \tag{2.7}$$

$$= \frac{1}{2}\sum_i \sum_k \delta_{ik}(x_i - y_k)'A(x_i - y_k) - \frac{N}{2}\log\det(A) + \frac{c}{2}\sum_i ||q_i - \Lambda\delta_i||^2, \tag{2.8}$$

subject to

$$\begin{aligned} \delta_{ik} &\in \{0, 1\}, \text{ for all } i, k \\ \sum_k \delta_{ik} &= 1, \text{ for all } i \\ \xi_{km} &\in \{0, 1\}, \text{ for all } k, m \\ \sum_m \delta_{km} &= 1, \text{ for all } k, \end{aligned} \tag{2.9}$$

where δ, ξ, and y denote collections of $\{\delta_i\}$, $\{\xi_k\}$, and $\{y_k\}$ respectively, and c is a weighting constant. The learning process for discriminant analysis turns to search for a set of δ, ξ, y, and A, which minimize the weighted sum of the negative log-likelihood function and the design cost subject to a set of constraints as in equation 2.9. We consider the mathematical framework in equations 2.8 and 2.9 as a mixed integer and linear programming, of which $\{y_k\}$ and A are continuous geometrical variables and $\{\delta_i\}$ and $\{\xi_k\}$ are discrete combinatorial variables. In the following section, we employ a hybrid of the mean-field annealing and gradient-descent methods to the optimization of all these variables simultaneously.

3 Interactive Dynamics and Coupled Potts Models

A hybrid of the mean-field annealing and the gradient-descent methods is applied to the above mixed integer and linear programming. As a result,

four sets of interactive dynamics are developed for the variables A, $\{y_k\}$, $\{\delta_i\}$, and $\{\xi_k\}$, respectively. These dynamics interact following an analogous process of the physical annealing and perform a parallel and distributed learning process for discriminant analysis.

When we relate each vector δ_i or ξ_k to a Potts neuron, the two unitary constraints in equation 2.9 are subsequently taken over by the normalization of the Potts activation function. The above mathematical framework is reduced to the minimization of the energy function E. By fixing the matrix A and $\{y_k\}$, the mean-field annealing traces the mean configuration $< \delta >$ and $< \xi >$, emulating thermal equilibrium at each temperature. It follows that the probability of the system configuration is proportional to the Boltzmann distribution:

$$\Pr(\delta, \xi) \propto \exp(-\beta E(\delta, \xi)). \tag{3.1}$$

Following the annealing process to a sufficiently large β value, the Boltzmann distribution is ultimately dominated by the optimal configuration,

$$\lim_{\beta \to \infty} \Pr(\delta^*, \xi^*) = 1,$$

where

$$E(\delta^*, \xi^*) = \min_{\delta, \xi} E(\delta, \xi).$$

The annealing process gradually increases the parameter β from a sufficiently low value to a large one. At each β value, the process iteratively executes the mean-field equations to a stationary point, which represents the mean configuration for thermal equilibrium. The obtained mean configuration at each β value is used as the initial configuration for the process at its subsequent β value. The mean-field equation can be derived from the following free energy function, which is similar to that proposed by Peterson and Söderberg (1989),

$$\begin{aligned} &\Psi(y, A, \langle \delta \rangle, \langle \xi \rangle, v, u) \\ &= E(y, A, \langle \delta \rangle, \langle \xi \rangle) + \sum_i \sum_k \langle \delta_{ik} \rangle v_{ik} + \sum_k \sum_m \langle \xi_{km} \rangle u_{km} \qquad (3.2) \\ &-\frac{1}{\beta} \sum_i \ln \left(\sum_k \exp(\beta v_{ik}) \right) - \frac{1}{\beta} \sum_k \ln \left(\sum_m \exp(\beta u_{km}) \right), \qquad (3.3) \end{aligned}$$

where $\langle \delta \rangle$, $\langle \xi \rangle$, u, and v denote $\{\delta_i\}$, $\{\xi_k\}$, $\{u_{km}\}$, and $\{v_{ik}\}$, respectively, and u_i and v_k are auxiliary vectors. When fixing y, A and β, a saddle point of the

free energy satisfies the following condition:

$$\frac{\partial \Psi}{\partial \langle \delta_i \rangle} = 0, \quad \frac{\partial \Psi}{\partial v_i} = 0, \text{ for all } i$$
$$\frac{\partial \Psi}{\partial \langle \xi_k \rangle} = 0, \quad \frac{\partial \Psi}{\partial u_k} = 0, \text{ for all } k.$$

These lead to two sets of mean-field equations in the following vector form:

$$\begin{aligned} v_i &= -\frac{\partial E(y, A, \langle \delta \rangle, \langle \xi \rangle)}{\partial \langle \delta_i \rangle} \\ &= -\frac{1}{2}(x_i - y_k)' A (x_i - y_k) - c\Lambda'(q_i - \Lambda \delta_i) \end{aligned} \tag{3.4}$$

$$\langle \delta_i \rangle = \left[\frac{\exp(\beta v_{i1})}{\sum_h \exp(\beta v_{ih})}, \ldots, \frac{\exp(\beta v_{iK})}{\sum_h \exp(\beta v_{ih})} \right]' \tag{3.5}$$

$$\begin{aligned} u_k &= -\frac{\partial E(y, A, \langle \delta \rangle, \langle \xi \rangle)}{\partial \langle \xi_k \rangle} \\ &= c \sum_i \langle \delta_{ik} \rangle (q_i - \Lambda \langle \delta_i \rangle) \end{aligned} \tag{3.6}$$

$$\langle \xi_k \rangle = \left[\frac{\exp(\beta u_{k1})}{\sum_m \exp(\beta u_{km})}, \ldots, \frac{\exp(\beta u_{kM})}{\sum_m \exp(\beta u_{km})} \right]'. \tag{3.7}$$

During the stage of evaluating the mean configuration at each β value, the matrix A and the kernels $\{y_k\}$ are considered constants. Once determined, the mean configuration feeds back to the adaptation of the covariance matrix and the kernels. By applying the gradient-descent method to the free energy, we have the following updating rule for each element A_{mn} in the matrix A:

$$\begin{aligned} \triangle A_{mn} &\propto -\frac{\partial \Psi}{\partial A_{mn}} \\ &= -\frac{\partial E}{\partial A_{mn}} \\ &= -\frac{1}{2} \sum_i \sum_k \langle \delta_{ik} \rangle (x_{im} - y_{km})(x_{in} - y_{kn}) + \frac{N}{2}[(A')^{-1}]_{mn}. \end{aligned} \tag{3.8}$$

When all $\triangle A_{mn} = 0$, we have

$$A = (W^{-1})', \tag{3.9}$$

where

$$W_{mn} = \frac{1}{N} \sum_i \sum_k \langle \delta_{ik} \rangle \, (x_{im} - y_{km})(x_{in} - y_{kn}). \tag{3.10}$$

The adaption of the kernels $\{y_k\}$ is also derived by the gradient-descent method:

$$\begin{aligned} \triangle y_k &\propto -\frac{\partial \Psi}{\partial y_k} \\ &= \frac{1}{2} \sum_i \langle \delta_{ik} \rangle \, (A + A')(x_i - y_k). \end{aligned} \tag{3.11}$$

Again, when $\triangle y_k = 0$, we have

$$y_k = \frac{\sum_i \langle \delta_{ik} \rangle \, x_i}{\sum_i \langle \delta_{ik} \rangle}. \tag{3.12}$$

Now we conclude the new learning process for discriminant analysis as follows:

1. Set a sufficiently low β value, each kernel y_k near the mean of all predictors and each $\langle \delta_{ik} \rangle$ near $\frac{1}{K}$ and $\langle \xi_{km} \rangle$ near $\frac{1}{M}$.
2. Iteratively update all $\langle \delta_{ik} \rangle$ and v_{ik} by equations 3.4 and 3.5, respectively, to a stationary point.
3. Iteratively update each $\langle \xi_{km} \rangle$ and u_{km} by equations 3.6 and 3.7, respectively, to a stationary point.
4. Update each $\mathbf{y}_i$ by equation 3.12.
5. Update $\mathbf{A}$ by equations 3.9 and 3.10.
6. If $\sum_{ik} \langle \delta_{ik} \rangle^2$ and $\sum_{km} \langle \xi_{km} \rangle^2$ are larger than a prior threshold, then halt; otherwise increase β by an annealing schedule and go to step 2.

The convergence of the algorithm is well guaranteed. For steps 2 and 3, two sets of mean-field equations define a stationary point of the free energy, equation 3.2. For steps 4 and 5, since all $\{\langle \delta_{ik} \rangle\}$ and $\{\langle \xi_{km} \rangle\}$ are fixed, the change $\Delta \psi$ of the free energy, equation 3.2, due to the change $\Delta \mathbf{y}_k$ and ΔA_{mn} has the nonincreasing property supported by the gradient-descent method. A mathematical treatment to the convergence property is given in the appendix. The two sets of mean-field equations, 3.4–3.5 and 3.6–3.7, constitute two interactive Potts models. The learning network for the whole learning process is shown in Figure 1.

4 Incremental Learning for Optimal Model Size

An incremental learning procedure is developed for the interactive Potts models to optimize the model size subject to the criterion of minimal design cost. The model size indicates the number of kernels. According to the learning process in the last section, when the annealing process halts with a sufficiently large β, each individual mean $\langle\delta_{ik}\rangle$ or $\langle\xi_{km}\rangle$ is close to either one or zero and the two square sums of mean activations are larger than the predetermined threshold. The variables $\{\langle\delta_{ik}\rangle\}$ and $\{\langle\xi_{km}\rangle\}$ are first recorded by temporal variables $\{\delta^*_{ik}\}$, $\{\xi^*_{km}\}$, respectively, to establish an intermediate recognizing network with kernels $\{y_k\}$ and the matrix A. Whether the model size of this intermediate recognizing network, which is the size of $\{y_k\}$, is sufficient depends on the quantity of the design cost $\sum_i ||q_i - \sum_k \xi^*_k \delta^*_{ik}||^2$, which denotes the number of errors of predicting all training samples. Define the hit ratio r as $1 - \frac{1}{N}\sum_i \|q_i - \sum_k \xi^*_k \delta^*_{ik}\|^2$. If the hit ratio is not acceptable, such as $r < \theta$ and $\theta = 0.98$, it is conjectured that the underlying boundary structure for well-discriminating training samples overloads the partition formed by the kernels $\{y_k\}$ and the covariance matrix A. The incremental learning procedure aims to improve this hit ratio r by properly increasing the model size and adapting the boundary structure. Define the local hit ratio r_k as $1 - \frac{1}{N_k}\sum_i \delta^*_{ik}\|q_i - \xi^*_k\|$ for each internal region $\Omega_k = \{x | k = \arg\min_j \|x - y_j\|_A\}$, where N_k denotes the size of the set $\{x_i \in \Omega_k\}$. A set of underestimated internal regions, each having an unacceptable local hit ratio, such as $r_k < \theta$, is then picked out and their kernels are duplicated. The duplication involves the variation of the original coupled Potts model in organization.

The idea of divide-and-conquer tends to invoke a subtask for each underestimated region and then deal with each subtask independently. This is not the best choice, since it loses the point of global optimization of the boundary structure for discriminant analysis. Alternatively, the incremental learning procedure makes use of collective decisions of interactive Potts models. The kernel y_k of an underestimated internal region, such as $r_k < \theta$, is duplicated with small perturbation to produce its twin kernel denoted by y^*_k. A set of new kernels $\{y^{new}_k\}$ is created to be the union of $\{y_k\}$ and $\{y^*_k | r_k < \theta\}$, having the model size of $K + K^*$, where K and K^*, respectively, denote the number of the original kernels and that of the underestimated internal regions. In the new set, let the index of y_k be still k and that of y^*_k be denoted by k'. For each input parameter x_i, assuming $x_i \in \Omega(y_k)$, a new membership vector δ^{new}_i, now with $K + K^*$ element, is created as follows:

$$\begin{aligned}\delta^{new}_i &= e^{K+K^*}_k \text{ if } r_k \geq \theta \\ &= \frac{1}{2}\left(e^{K+K^*}_k + e^{K+K^*}_{k'}\right) \text{ otherwise.}\end{aligned}$$

In the second line of the above equation, the new vector δ^{new}_i means that x_i has the same probability $\frac{1}{2}$ of being mapped to each of the twins, y^{new}_k and

y_k^{new}. Create new category responses $\{\xi_k^{new}\}$ for the new kernels $\{y_k^{new}\}$, and set each element of ξ_k^{new} near $\frac{1}{M}$. After duplication and replacing all means with new vectors $\{\delta_i^{new}\}$, $\{\xi_k^{new}, 1 \leqq k \leqq K+K^*\}$ and kernels with $\{y_k^{new}\}$, the system variables include $\{y_k, 1 \leqq k \leqq K + K^*\}$, $\{\langle\delta_i\rangle\}$ and $\{\langle\xi_k\rangle, 1 \leqq k \leqq K + K^*\}$. Note that now each $\langle\delta_i\rangle$ contains $K + K^*$ elements. Return to the annealing process. The β value has been increased to a sufficiently large one, where the two sums $\sum(\delta_{ik}^*)^2$ and $\sum(\xi_{km}^*)^2$ are larger than the predetermined halting threshold, but these system variables after duplication no more satisfy the halting condition. The β value can be further increased to continue the annealing process. We conclude the incremental learning process as follows for the interactive Potts models:

1. Set a sufficiently low β value, a threshold θ, and an initial model size K. Set each kernel y_k near the mean of all predictors, each $\langle\delta_{ik}\rangle$ near $\frac{1}{K}$, and $\langle\xi_{km}\rangle$ near $\frac{1}{M}$.
2. Iteratively update all $\langle\delta_{ik}\rangle$ and v_{ik} by equations 3.4 and 3.5, respectively, to a stationary point.
3. Iteratively update each $\langle\xi_{km}\rangle$ and u_{km} by equations 3.6 and 3.7, respectively to a stationary point.
4. Update each y_i by equation 3.12.
5. Update A by equations 3.9 and 3.10.
6. If $\sum_{ik}\langle\delta_{ik}\rangle^2$ and $\sum_{km}\langle\xi_{km}\rangle^2$ are larger than a prior threshold, such as 0.98, then go to step 7. Otherwise, increase β by an annealing schedule, and then go to step 2.
7. Record $\{\langle\delta_i\rangle\}$, $\{\langle\xi_k\rangle\}$ by temporal variables $\{\delta_i^*\}$, $\{\xi_k^*\}$, respectively.
8. Determine r and all r_k using A, $\{y_k\}$, $\{\delta_i^*\}$, $\{\xi_k^*\}$.
9. If $r > \theta$, halt. Otherwise, duplicate the kernels of K^* underestimated regions with small perturbation and create new variables $\{y_k^{nw}\}$, $\{\delta_i^{new}\}$, $\{\xi_k^{new}\}$ using $\{y_k\}$, $\{y_k^*|r_k < \theta\}$, $\{\delta_i^*\}$, $\{\xi_k^*\}$, as described in the text.
10. $K \leftarrow K + K^*$. Decrease β with a small constant, and replace $\{y_k\}$, $\{\langle\delta_i\rangle\}$, and $\{\langle\xi_k\rangle\}$ with $\{y_k^{new}\}$, $\{\delta_i^{new}\}$, and $\{\xi_k^{new}\}$ respectively, and go to step 2.

5 Numerical Simulations and Discussion

The incremental learning process in section 4 has been implemented in Matlab codes and is referred to as PottsDA in the following context.

5.1 Examples. We first test the new method (PottsDA) in comparisons with the RBF and support vector machine (SVM) methods (Vapnik, 1995) using some artificial examples. In our simulations, the β parameter of the PottsDA is initialized as $\frac{1}{3.8}$, and each annealing process increases it to a

value of $\frac{\beta}{0.98}$; the weighting constant is $c = 4.5$. Theoretical derivations of the weighting constant c and the initial β parameter can further refer to Aiyer, Niranjan, and Fallside (1990) and Peterson and Söderberg (1989), respectively. The Matlab package used for the RBF (Muller et al., 1999; Ratsch et al., 2001) is provided in Ratsch et al. (2001), where the centers are initialized with k-means clustering. For the SVM (Vapnik, 1995), the Matlab package is provided in Cawley (2000), and the corresponding learning method is of sequential minimal optimization (Platt, 1999). In this section, the three methods are executed five times for each example. Their average error rates for training and testing are reported.

The input parameters in the first example are generated by a linear mixture, $x(t) = Hs(t)$, where $s(t) = [s^1(t)\ s^2(t)]'$ denotes time-varying samples from two independently uniform distributions within $[-.05, 0.5]$, and

$$H = \begin{bmatrix} 0.4384 & -0.8988 \\ -0.8493 & 0.5279 \end{bmatrix}$$

is a randomly generated mixing matrix. The desired output of each input parameter is determined by the rule $q(t) = sign(s^1(t)) * sign(s^2(t))$. We use the same process to generate 1600 samples and split them into two equal sets—one for training and the other for testing. In Figure 2, the position of each input parameter in the training set is marked with a black or gray dot, which, respectively, denote two distinct output labels. Since the input parameter in this example is a linear mixture of independent sources, a discriminant rule depending on only the input parameter, such as $sign(x^1(t)) * sign(x^2(t))$, does not describe an optimal prediction for correct output labels. The primary challenge to the learning process is the recovery of the original independent sources such that an effective discriminant rule can be encoded by a minimal set of kernels to achieve maximal generalization. Our simulations show that the PottsDA method outperforms the RBF and the SVM methods for this example. The PottsDA has an initial model of two kernels and halts with an optimal hit ratio of $r = 100\%$. As shown in Table 1, the error rates of the PottsDA for both training and testing are zero. This is a result carried out by a discriminant network composed of a covariance matrix, four kernels and their category responses. In Figure 2, the position of each of four kernels is marked with a circle or cross symbol representing the category denoted by black or gray dots, respectively. By the relation $B'B = A$, one can obtain a demixing matrix,

$$B = \begin{bmatrix} 12.0370 & 5.8743 \\ 5.8743 & 10.2847 \end{bmatrix}$$

from the covariance matrix A. The two columns of the inverse B^{-1} are shown by the two lines in Figure 2, which exactly coincide with the mixing structure in direction for this example. The obtained covariance matrix provides

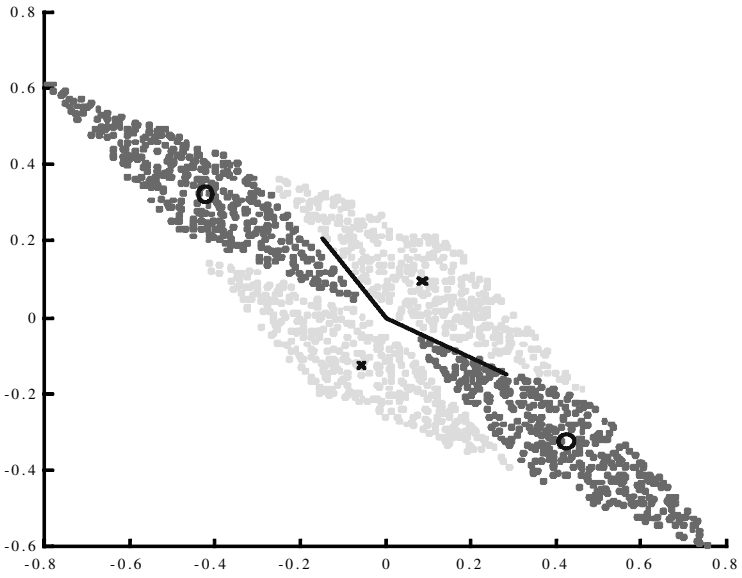

Figure 2: The training patterns of the first example and the result of the learning process, including the two columns of the inverse of the demixing matrix, the four kernels, and their category responses.

Table 1: Performance of the Three Methods, First Example.

	RBF(4)	RBF(8)	RBF(12)	RBF(24)	SVM	PottsDA(4)
Training	14.1%	12.0%	8.6%	3.9%	13.2%	0%
Testing	13.0	12.1	8.3	4.5	14.3	0

Note: Numbers in parentheses refer to number of kernels.

a suitable distance measure between input parameters and kernels such that the four kernels faithfully partition the parameter space into four internal regions and the resulting discriminant network successfully classifies samples in the training set and the testing set. In contrast, since the RBF network is based on the Euclidean distance, its kernels result in nonfaithful representations for input parameters. To illustrate this point, the RBF method was tested separately with 4, 8, 12, and 24 kernels. Our simulations show that the average error rate of the RBF method with 24 kernels is 3.9% for training and 4.5% for testing and that of the SVM method is 13.2% for training and 14.3% for testing. The average execution time of the PottsDA for this example is 13.4 seconds.

In the second example, the input parameter contains three elements. Each of the input parameters, $x(t) = [x^1(t)\ x^2(t)\ x^3(t)]'$, is a result of the linear

mixture, $x(t) = Hs(t)$, of three independent sources, $s(t) = [s^1(t)\ s^2(t)\ s^3(t)]'$, where H is a randomly generated mixing matrix with entries as follows:

$$H = \begin{bmatrix} 0.9288 & 0.2803 & 0.3770 \\ 0.3122 & 0.9366 & 0.2572 \\ 0.1994 & 0.2098 & 0.8897 \end{bmatrix}.$$

The first two sources, $s^1(t)$ and $s^2(t)$, are of uniform distributions within $[-0.5, 0.5]$, and $s^3(t)$ is a gaussian noise of $N(0, \sqrt{2})$. The discriminant rule is the same as in the first example, $sign(s^1(t)) * sign(s^2(t))$, treating the third source as a noise for prediction. To retrieve this discriminant rule from the mixture with the minimal model size, the learning process has to deal with interference caused by the mixing structure and the noise source. As in the first example, both the training set and the testing set contain 800 samples each generated by the same linear mixture. For the testing set, all samples of three independent sources are shown in the first three rows in Figure 3, and the three mixed signals, $x^1(t)$, $x^2(t)$, and $x^3(t)$, are shown by the next three rows. The seventh row in Figure 3 shows the desired output of each

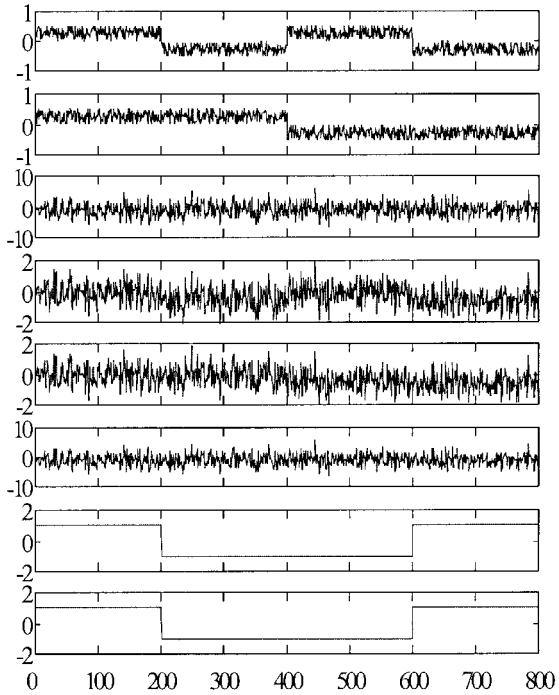

Figure 3: The time sequence of the three independent sources, the three input parameter, the desired output, and the predicted output of the second example.

Table 2: Performance of the Three Methods, Second Example.

	RBF(4)	RBF(8)	RBF(12)	RBF(24)	SVM	PottsDA(4)
Training	45.3%	31.2%	22.6%	10.9%	3.2%	0.2%
Test	44	31.1	24.6	13.9	5.9	0

Note: Numbers in parentheses refer to number of kernels.

sample. In all of five executions, the PottsDA derives a discriminant network composed of a distance measure, four kernels and their category responses, by which the resulting output for 800 testing samples as shown in the eighth row exactly coincides with the desired output. To facilitate presentations, according to the combination of the sign of $s_1(t)$ and $s_2(t)$, we have sorted the 800 testing samples in Figure 3 into four segments, such that each segment contains the same category response. As shown in Table 2, the PottsDA method is better than the RBF and the SVM methods in handling input parameters of linear mixtures with noises. The average execution time of the PottsDA for this example is 30.07 seconds.

The third example tests the three methods for the spiral data as shown in Figure 4, where two distinct categories are denoted by stars and dots, respectively. In this example, both the training set and testing set contain 40 spiral-distributed interleave clusters, and each cluster contains 20 input parameters. The primary challenge to a learning method is to find the centers of 40 clusters. For this example, the PottsDA method has an initial model of 10 kernels. The quantity $\sum \langle \delta_{ik} \rangle^2$ measured at step 7 in the PottsDA learning process along updating iterations is shown in Figure 5, which also displays the change of the model size. The final model size has been further reduced to 40 by considering possible combinations of any two neighboring kernels. The obtained 40 kernels in one of five executions with their category responses, denoted by circle or cross symbols, are plotted in Figure 4. The two lines in Figure 4 denote the two columns of the inverse of the obtained demixing matrix in direction. The average training and testing error rates of the three methods are shown in Table 3. Obviously, the PottsDA method also outperforms the RBF and the SVM methods for this example.

5.2 Breast Cancer Data. We use the Wisconsin Breast Cancer Database (as of July 1992) to test the PottsDA method for actual applications. This database contains 699 instances, each containing 9 features for predicting one of benign and malignant categories. There are 458 instances in the benign category and 241 instances in the malignant category in this database. The input parameters are clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses, each represented by integers ranging from 1 to 10. The original work (Wolberg & Mangasarian, 1990) ap-

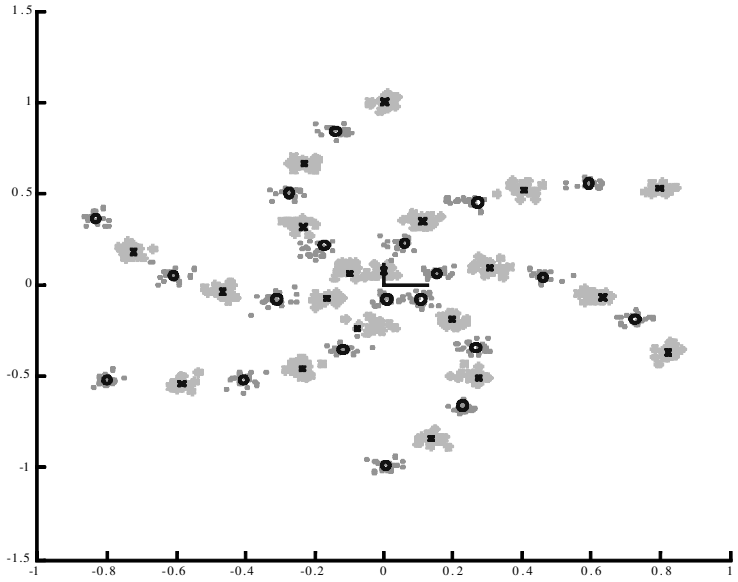

Figure 4: The training patterns of the third example and the result of the learning process, including the two columns of the inverse of the demixing matrix, the 40 kernels, and their category responses.

plied the multisurface method to a 369-case subset of the database, resulting in error testing rates more than 6% (Malini Lamego, 2001). Recently, Malini Lamego (2001) used the neural network with algebraic loops to deal with the first 683 instances of the database. In his experiment, the last 200 instances of the 683 instances form the testing set, and the others form the learning set; the resulting error rates are 2.3% for learning and 4.5% for testing. It has been claimed (Malini Lamego, 2001) that the classification is more difficult than the previous one (Wolberg & Mangasarian, 1990), and the result is better than those of all previous works for this database. For comparison, we use the same training set and testing set as in Malini Lamego (2001) to evaluate the performance of the PottsDA method. The PottsDA method obtains a discriminant network with 42 kernels and has error rates of 1.4% for training and 1% for testing, as shown in Table 4. Only two instances in the testing set are incorrectly classified by the discriminant network. If the testing set also includes the last 19 instances in the database, one additional instance is missed by the same discriminant network, and the test error rate is 1.39%. For the 219-case test set, the RBF method with 80 kernels and the SVM method result in error rates of 4.17% and 4.63% for testing, respectively. The PottsDA method is significantly better than the other two methods for the Wisconsin Breast Cancer Database.

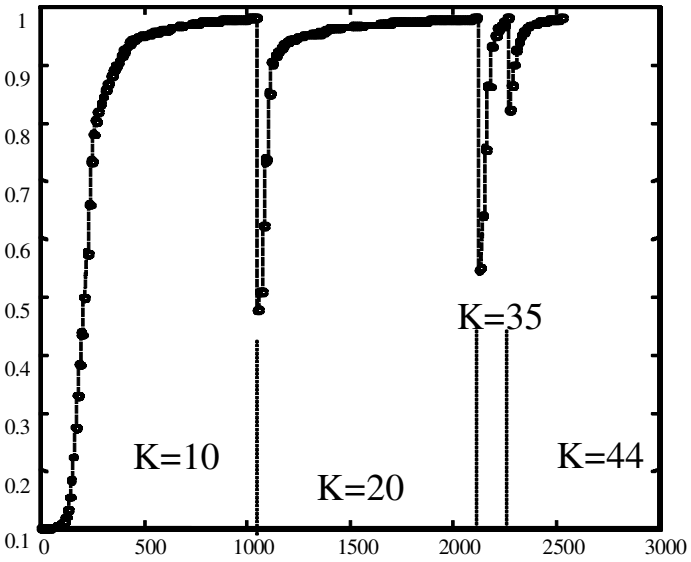

Figure 5: The convergence of the learning network for the third example and the change of the model size. The vertical coordinate denotes the ratio of the square sum of the mean activations of membership vectors to the number of training patterns. The horizontal coordinate is the time index, each denoting a change of the beta value.

Table 3: Performance of the Three Methods, Third Example.

	RBF(40)	RBF(50)	RBF(60)	RBF(80)	SVM	PottsDA(40)
Training	14.6%	10.4%	7.8%	3.3%	45.5%	0.8%
Test	15.7	12.3	9.5	4.1	45.6	0.4

Note: Numbers in parentheses refer to number of kernels.

Table 4: Performance of the PottsDA Method and the Neural Network with Algebraic Loops for the 683-Case Subset of the Wisconsin Breast Cancer Database.

	PottsDA(42)	Neural Net with Algebraic Loops
Training (483)	1.4%	2.3%
Testing (200)	1	4.5

5.3 Discussion. The major improvement of the PottsDA method compared to the other methods is illustrated from four perspectives: the flexibility of the discriminant network, effective collective decisions of the annealed recurrent learning network, the advantages of the generative model, and the capability of the incremental learning process.

5.3.1 Discriminant Network. The discriminant network of the PottsDA method composed of normalized multivariate gaussian activation functions $\{G_k^A\}$ of equation 1.3 is a general version of a normalized RBF network. To an extreme large β, the discriminant function, equation 1.5, is a piecewise function composed of a set of local functions, each defined within an internal region of a faithful nonoverlapping partition into the parameter space based on the Mahalanobis distance $\|\cdot\|_A$. This discriminant function is indeed a composition of a linear transformation and the nearest prototype classifier as in equation 1.6, and it possesses more flexibility for a desired mapping function. Consider the first two artificial examples, where the training parameters are results of linear mixtures of independent sources, and their targets are exactly encoded with source instances instead of mixtures. With an adaptive distance measure A, the PottsDA method succeeds in locating four kernels $\{y_k\}$ for the optimal discriminant function. In contrast, because of using the Euclidean distance and lacking circumspect efforts to recover independent instances from mixtures, the RBF method results in nonfaithful internal representations, a relatively large number of local functions based on a Voronoi partition. As shown in Tables 2 and 3, the testing error rate of the RBF method of 24 kernels is higher than that of the PottsDA method of four kernels. This reflects the weakness of the discriminant function of a normalized RBF network as being a special case of the PottsDA method.

5.3.2 Annealed Recurrent Learning Network. The annealed recurrent learning network of the PottsDA method containing four sets of interactive dynamics is effective for optimizing highly coupled parameters of the discriminant network under an annealing process. The supervised learning process is formulated into the mathematical framework, equation 2.7, composed of a mixed integer and linear programming, and a hybrid of the mean-field annealing and the gradient-descent method is employed to derive linear and nonlinear interactive dynamics.

The primary advantage of the annealed recurrent learning network over a cascaded learning process or simply a gradient-descent-based learning process is the effect of collective decisions for four sets of continuous and discrete variables and the capability of escaping from the trap of tremendous local minima within the energy function to approach a global minimum. Consider the second artificial example, where input parameters are a result of linear mixtures and one independent source is treated as noise to its discriminant rule. Collective decisions realized by the annealed recurrent learning network succeed in dealing with component analysis, clustering

analysis, and labeling determination as a whole process and can thus achieve an optimal discriminant function for this problem. Numerical simulations show that the power of the discriminant network is strongly supported by the annealed recurrent learning network. Although the RBF method has employed the k-means algorithm to set up its initial kernels, it cannot well handle the interference caused by linear mixtures and noises due to the limitation from its discriminant function and learning process. Further evaluations show that the testing error rate of the RBF method of 100 kernels is still above 10% for this example.

5.3.3 Generative Model. Both the discriminant network and the annealed recurrent learning network are rooted from the generative model of multiple disjoint multivariate gaussian distributions. The overall distribution corresponding to the generative model is general enough to characterize an arbitrary parameter space, and the involved parameter estimation gains potential advantages from maximal likelihood principle, which provides solid theoretical fundamentals to develop the mathematical framework. To realize natural discriminant analysis, the PottsDA method initiates a generative model to all predictors, using the annealed recurrent learning network to adapt the kernels and the covariance matrix subject to interpolating conditions proposed by paired training samples, and using the discriminant network to classify instances. A simplified version of the generative model is simulated with a unified covariance matrix. For the Wisconsin Breast Cancer Database, its performance is encouraging. The obtained parameters, including the 42 kernels and the covariance matrix, could provide feedback for understanding relations among components of predictors.

5.3.4 Incremental Learning Process. The incremental learning process is capable of determining minimal model size subject to interpolating conditions. Consider the third artificial example composed of 40 interleaving clusters in two different classes. Without interferences caused by linear mixtures and noises, this problem tests the capability for clustering analysis. For this example, the RBF method behaves better than the SVM method, but it still takes the RBF method 80 kernels to produce a testing error rate of 4.1%. The incremental learning process of the PottsDA method is more effective. It obtains a discriminant network of 40 kernels with testing error rates near zero.

6 Conclusions

We have proposed a new learning process for discriminant analysis based on the four sets of interactive dynamics, and its encouraging performance has been shown by numerical simulations for some artificial and real examples. To develop the interactive dynamics, we have proposed multiple disjoint multivariate gaussian distributions to serve as a generative model for

the parameter space. By combining the maximization of the log-likelihood functions, the fitness of the generative model to all input parameters, and the minimization of the design cost, we have a mathematical framework for discriminant analysis. By relating the discrete variables to Potts neural variables, we can further apply a hybrid of the mean-field annealing and gradient-descent methods to the optimization of the mixed integer and linear programing and obtain the four sets of interactive dynamics performing the annealed recurrent learning network for discriminant analysis. The new learning process is of a parallel and distributed process, and its evolution is well controlled by an annealing process in an analog with the physical annealing. An effective incremental learning procedure is also developed for optimizing the model size. The adaptive covariance matrix of the discriminant network plays a central role of retrieving the unknown mixing structure within the input parameters and extracting output-dependent features for discriminant analysis. The new learning process is effective for developing faithful internal representations of the input parameters and constructing essential boundary structures for classification.

Appendix

That steps 2–5 in the learning process converge can be proved. Rewrite the mean-field equations in the context as the following continuous form,

$$\frac{d\mathbf{u}_{ik}}{dt} = -\frac{\partial \psi}{\partial \langle \delta_{ik} \rangle} = \frac{-\partial E(y, A, \langle \delta \rangle, \langle \xi \rangle)}{\partial \langle \delta_{ik} \rangle}$$

$$\langle \delta_i \rangle = \left[\frac{\exp(\beta u_{i1})}{\sum_l \exp(\beta u_{il})} \cdots \frac{\exp(\beta u_{iK})}{\sum_l \exp(\beta u_{il})} \right]'$$

$$= \sum_k \frac{\exp(\beta u_{ik})}{\sum_l \exp(\beta u_{il})} e_k$$

and

$$\frac{dv_{km}}{dt} = -\frac{\partial \psi}{\partial \langle \xi_{km} \rangle} = \frac{-\partial E(y, A, \langle \delta \rangle, \langle \xi \rangle)}{\partial \langle \xi_{km} \rangle}$$

$$\langle \xi_k \rangle = \left[\frac{\exp(\beta v_{k1})}{\sum_l \exp(\beta v_{kl})} \cdots \frac{\exp(\beta v_{kM})}{\sum_l \exp(\beta v_{kl})} \right]'$$

$$= \sum_h \frac{\exp(\beta v_{kh})}{\sum_l \exp(\beta v_{kl})} e_h,$$

where vector e_k is a standard unit vector of which the kth element is one. Then rewrite the updating rule as the following dynamics:

$$\frac{dA_{mn}}{dt} \equiv -\eta_1 \frac{\partial \psi}{\partial A_{mn}} = -\eta_1 \frac{\partial E(y, A, \langle \delta \rangle, \langle \xi \rangle)}{\partial A_{mn}}$$

and

$$\frac{dy_k}{dt} \equiv -\eta_2 \frac{\partial \psi}{\partial y_k} = -\eta_2 \frac{\partial E(y, A, \langle \delta \rangle, \langle \xi \rangle)}{\partial y_k}.$$

Then the convergence of the free energy ψ along the trace of these four sets of dynamics can be shown:

$$\begin{aligned}
\frac{d\psi}{dt} &= \sum_i \left(\frac{\partial \psi}{\partial \langle \delta_i \rangle} \right)' \frac{d \langle \delta_k \rangle}{dt} + \sum_k \left(\frac{\partial \psi}{\partial \langle \xi_k \rangle} \right)' \frac{d \langle \xi_k \rangle}{dt} \\
&\quad + \sum_{mn} \left(\frac{\partial \psi}{\partial A_{mn}} \right) \frac{dA_{mn}}{dt} + \sum_k \left(\frac{\partial \psi}{\partial y_k} \right)' \frac{dy_k}{dt} \\
&= -\sum_i \left(\frac{du_i}{dt} \right)' \left(\Gamma_1 \frac{du_i}{dt} \right) - \sum_k \left(\frac{dv_k}{dt} \right)' \left(\Gamma_2 \frac{dv_k}{dt} \right) \\
&\quad - \eta_1 \sum_{mn} \left(\frac{dA_{mn}}{dt} \right) \left(\frac{dA_{mn}}{dt} \right) - \eta_2 \sum_k \left(\frac{dy_k}{dt} \right)' \left(\frac{dy_k}{dt} \right), \\
&\leqq 0
\end{aligned}$$

where Γ_1 is the Hessian of $\ln z(u_k, \beta)$,

$$\Gamma_1 = \frac{\sum_{[\sigma_l]} \exp(\beta \langle \delta_k \rangle' \sigma_l)[\sigma_l - \langle \delta_k \rangle][\sigma_l - \langle \delta_k \rangle]'}{\sum_{[\sigma_l]} \exp(\beta \langle \delta_k \rangle' \sigma_l)}.$$

$[\sigma_k]$ runs over $\{e_1, \ldots, e_K\}$. Since Γ_1 is positive definite,

$$\left(\frac{du_k}{dt} \right)' \left(\Gamma_1 \frac{du_k}{dt} \right) > 0.$$

For the same reason,

$$\left(\frac{dv_m}{dt} \right)' \left(\Gamma_2 \frac{dv_m}{dt} \right) > 0.$$

$\frac{d\psi}{dt} \leqq 0$ is shown.

References

Aiyer, S. V. B., Niranjan, M., & Fallside, F. (1990). A theoretical investigation into the performance of the Hopfield model. *IEEE Trans. Neural Networks, 1*, 204–215.

Benaim, M. (1994). On functional approximation with normalized gaussian units. *Neural Computation, 6*, 319–333.

Cawley, G. C. (2000). MATLAB Support Vector Machine Toolbox. Available online at: http://theoval.sys.uea.ac.uk/svm/toolbox.

Freeman, J. A. S., & Saad, D. (1995). Learning and generalization in radial basis function networks. *Neural Computation, 7*, 1000–1020.

Girosi, F. (1998). An equivalence between sparse approximation and support vector machine. *Neural Computation, 10*, 1455–1480.

Girosi, F., Jones, M., & Poggio, T. (1995). Regularization theory and neural networks architectures. *Neural Computation, 7*, 219–269.

Hastie, T., Buja, A., & Tibshirani, R. (1995). Penalized discriminant-analysis . *Ann. Stat., 23*, 73–102.

Hastie, T., & Simard, P. Y. (1998). Metrics and models for handwritten character recognition. *Stat. Sci., 13*, 54–65.

Hastie, T., & Tibshirani, R. (1996). Discriminant analysis by gaussian mixtures. *J. Roy. Stat. Soc. B Met., 58*, 155–176.

Hastie, T., Tibshirani, R., & Buja, A. (1994). Flexible discriminant-analysis by optimal scoring. *J. Am. Stat. Assoc., 89*, 1255–1270.

Lin, J. K., Grier, D. G., & Cowan, J. D. (1997). Faithful representation of separable distribution. *Neural Computation, 9*, 1305–1320.

Liou, C. Y., & Wu, J.-M. (1996). Self-organization using Potts models. *Neural Networks, 9*, 671–684.

Makeig, S., Jung, T. P., & Bell, A. J. (1997). Blind separation of auditory event–related brain responses into independent components. *P. Natl. Acad. Sci. USA, 94*, 10979–10984.

Malini Lamego, M. (2001). Adaptive structures with algebraic loops. *IEEE Trans. Neural Networks, 12*, 33–42.

Miller, D. J., & Uyar, H. S. (1998). Combined learning and use for a mixture model equivalent to the RBF classifier. *Neural Computation, 10*, 281–293.

Moody, J., & Darken, C. (1988). Learning with localized receptive fields. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), *Proceedings of the 1988 Connectionist Models Summer School* (pp. 133–143). San Mateo, CA: Morgan Kaufmann.

Moody, J., & Darken, C. (1989). Fast learning in networks of locally-tuned processing units. *Neural Computation, 1*, 281–294.

Muller, K.-R., Smola, A. J., Ratsch, G., Scholkopf, B., Kohlmorgen, J., & Vapnik, V. (1999). Using support vector machines for time series prediction. In B. Scholkopf, C. J. C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods—Support vector learning* (pp. 243–254). Cambridge, MA: MIT Press.

Peterson, C., & Söderberg, B. (1989). A new method for mapping optimization problems onto neural network. *Int. J. Neural Syst., 1*, 3–22.

Pineda, F. J. (1987).Generalization of back-propagation to recurrent neural networks. *Physical Review Letters, 59*, 2229–2232.

Pineda, F. J. (1989). Recurrent back-propagation and the dynamical approach to adaptive neural computation. *Neural Computation, 1*, 161–172.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Scholkopf, C. Burges, & A. J. Smola (Eds.), *Advances in kernel methods—Support vector learning* (pp. 185–208). Cambridge, MA: MIT Press.

Rao, A. V., Miller, D. J., Rose, K., & Gersho, A. (1999). A deterministic annealing approach for parsimonious design of piecewise regression models. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 21*, 159–173.

Ratsch, G., Onoda, T., & Muller, K. R. (2001). Soft margins for AdaBoost. *Machine Learning, 42*, 287–320.

Rose, K., Gurewitz, E., & Fox, G. C. (1990). Statistical mechanics and phase transitions in clustering. *Phys. Rev. Lett., 65*, 945–948.

Rose, K., Gurewitz, E., & Fox, G. C. (1993). Constrained clustering as an optimization method. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 15*, 785–794.

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel and distributed processing* (Vol. 1). Cambridge, MA: MIT Press.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

Wolberg, W. H., & Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Science, 87*, 9193–9196.

Received September 8, 2000; accepted June 18, 2001.