Available online at www.sciencedirect.com

SCIENCE DIRECT®

Neural Networks 18 (2005) 719–726

Neural Networks

www.elsevier.com/locate/neunet

2005 Special issue

A new classifier based on information theoretic learning with unlabeled data

Kyu-Hwa Jeong[a,*], Jian-Wu Xu[a], Deniz Erdogmus[b], Jose C. Principe[a]

[a]Computational NeuroEngineering Laboratory, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA
[b]Oregon Graduate Institute, OHSU, Portland, OR 97006, USA

Abstract

Supervised learning is conventionally performed with pairwise input–output labeled data. After the training procedure, the adaptive system's weights are fixed while the testing procedure with unlabeled data is performed. Recently, in an attempt to improve classification performance unlabeled data has been exploited in the machine learning community. In this paper, we present an information theoretic learning (ITL) approach based on density divergence minimization to obtain an extended training algorithm using unlabeled data during the testing. The method uses a boosting-like algorithm with an ITL based cost function. Preliminary simulations suggest that the method has the potential to improve the performance of classifiers in the application phase.
© 2005 Elsevier Ltd. All rights reserved.

Keywords: Adaptive system; Unlabeled data; Classification; Information Theoretic learning; (ITL)

1. Introduction

Supervised learning, including system identification, regression and classification, is performed with input–output labeled data[1] using linear and nonlinear system topologies and optimal criteria based on statistics of the error between the desired samples and the system output. In the classification problem, the purpose of learning is to extract as much information as possible from the labeled training data to obtain optimal system weights so that they generalize to unlabeled data (typically by splitting the data set into the training and testing sets). Once the system is trained, there is no further optimization carried out over the unlabeled data during the actual application (testing) phase[2] (Haykin, 1999). This approach is considered natural to all since labels are not available for the testing data to further train the adaptive system. However, there is partial new information contained in the input test data that is simply discarded. In particular, if the test data is not drawn uniformly from the class distributions as frequently happens in biomedical applications (the inter subject variability is normally very large with each subject data lying in a subspace of the within class manifold), seeking ways to adapt the weights during testing seem reasonable.

Researchers in machine learning have been proposing learning methods from mixing labeled and unlabeled data due to the fact that labeled samples are much more expensive to collect when compared with unlabeled samples. The challenge is to design classifiers that can utilize the information present both in the labeled as well as in the unlabeled data. One method is transductive inference using support vector machines proposed by Vapnik et al. (Gammerman, Vapnik, & Vowk, 1998; Saunders, Gammerman, & Vowk, 1999). Transduction goes from particular (past) samples to particular (future) samples without any attempt to generalize. Another prominent approach is active learning where the learner can 'ask' the expert for a label of a sample in contrast to normal (passive) machine learning where the learner is presented with a static set of examples used to construct a model (Novak, 2004). There are also some other methods in the literature such as the EM algorithm in a maximum

* Corresponding author.
E-mail addresses: khjeong@cnel.ufl.edu (K.-H. Jeong), jianwu@cnel.ufl.edu (J.-W. Xu), derdogmus@ieee.org (D. Erdogmus), principe@cnel.ufl.edu (J.C. Principe).

[1] In this paper, labeled data pairs are those that are available in the form (x, y) where x denote the input sample and y is the corresponding desired sample.

[2] The testing phase means a real application phase and the testing data means the unlabeled data set throughout this paper.

0893-6080/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2005.06.018

likelihood framework (Blum & Mitchell, 1998; Nigram, McCallum, Thrum, & Mitchell, 2000); Smoother function approximation of unlabeled data using the representer theorem in the context of regularization (Belkin, Niyogi, & Sindhwani, 2004). Recently, Erdogmus et al. and Jeong et al. introduced an information theoretic framework based on Kullback–Leibler divergence minimization for training adaptive systems in supervised learning settings using both labeled and unlabeled data (Erdogmus, Rao, & Principe, 2005; Jeong, Xu, & Principe, 2005). This information theoretic learning framework can exploit the information of the labeled and unlabeled data to improve the performance of classifiers in the application phase.

This information theoretic learning framework has the potential to exploit the information contained in the labeled and unlabeled data to improve the performance of classifiers in the application phase by exploiting the distance between the system outputs during training and testing. In this paper, we propose the Euclidean distance based probability density function (pdf) matching algorithm to extend the adaptive system training even after supervised learning is completed. We use the aggregate information of the system output from the training data and the output to unlabeled novel testing data to adjust the classifier weights. One big advantage of this new information theoretic approach is the simplicity of computation. We also elucidate the circumstances under which our method can improve the performance of classification. Additionally, we propose a new boosting-like algorithm to make up for the absence of the a priori probability knowledge for the test data set. The paper is organized as follows. First, we give the description of the problem. Then the Euclidean distance based probability density function matching algorithm and a new boosting-like procedure are presented in Section 3. Next, we discuss the conditions under which our method might improve the performance of classification. In order to test our algorithm, we apply our method to the classification problem with an artificial data set and a real biomedical data set in Section 5.

2. Problem description

Consider a general function approximation problem. Suppose we have input–output data $\{(u_1,y_1),\ldots,(u_N,y_N)\}$ available from an unknown nonlinear function as follow:

$$d = f(\mathbf{u}) + n \tag{1}$$

The observed output (desired response) d is called the label of the input signal $\mathbf{u}$ and n is additive noise that translates our imprecise knowledge of the label. In function approximation, the labels are continuous-valued while in classification problem, the labels are discrete. The goal of supervised learning is to construct an adaptive system with input signal $\mathbf{u}$, output y, and weights $\mathbf{w}$ to approximate the function f or classify the output into different categories.

$$y = g(\mathbf{u}, \mathbf{w}) \tag{2}$$

The adaptive system could be a linear filter ($y = \mathbf{w}^{\mathrm{T}}\mathbf{u}$), a neural network, or any other topology. The process of supervised learning seeks to optimize the weights to extract relevant information from input–output sample pairs for a specific task. The optimization is carried out by minimization or maximization of an optimality criterion. Typically the mean square error (MSE) (Haykin, 1999; Widrow & Stearns, 1985) is used, however, there are also alternative selections such as minimum error entropy (Principe, Xu, & Fisher, 2000) or the $\in$-insensitive loss function (Cristianini & Taylor, 2000). The error signal is defined as the difference between the available desired output and the output produced by the adaptive system for a particular input ($e = d - y$). In the training phase, the adaptive system weights are adjusted to obtain an approximation to f or classify the input data into different categories by minimizing the error. In the application (testing) phase, the weights are fixed and the trained system is tested on novel unlabeled input data $\{u_{N+1},\ldots,u_K\}$. The probability distributions of the input signal for the training $\{u_1,\ldots,u_N\}$ and for the testing $\{u_{N+1},\ldots,u_K\}$ can be the same or different. The following question is to be answered in this paper: How can we continue updating the weights of the adaptive system in the application phase and under what conditions can we improve performance?

3. Information theoretic learning with unlabeled data

3.1. Euclidean distance-matching algorithm

3.1.1. General idea

Our initial idea was to combine the desired response in the training with the classifier output to unlabeled input data in the test set in order to continue adjusting the system weights (Jeong et al., 2005). When the a priori probability of each class during the test is the same as for the training, the system weights during the application phase continue to be adapted by the Euclidean distance pdf matching algorithm given by

$$\min_{\mathbf{w}} \int (f_{\mathrm{dtrn}}(x) - f_{\mathrm{ytst}}(x))^2 \, dx \tag{3}$$

where $f_{dtrn}(x)$ is the pdf of desired signal during the training phase, $f_{ytst}(x)$ is the pdf of system output signal during the testing phase and $\mathbf{w}$ is the weight vector of the adaptive system. If the two distributions are close to each other, the Euclidean distance pdf matching cost function minimizes the divergence between the training desired signal and the testing output signal. In other words, we create a desired signal for the input data during the testing by utilizing the pdf information of the desired signal encountered during the training. This is the reason why we must require the same a priori probabilities between the training and testing. Instead of setting the matching problem in the input data space directly, the Euclidean distance pdf matching criterion utilizes the system output space, which avoids the curse of dimensionality.

The Euclidean distance pdf matching cost function can be computed directly from data samples (i.e. nonparametrically). This requires a smooth (i.e. continuous and differentiable) estimator for the two probability density functions $f_{\mathrm{dtrn}}(x)$ and $f_{\mathrm{ytst}}(x)$. One nonparametric method to estimate the densities is the Parzen window method (Parzen, 1962). Given N independent and identically distributed (iid) samples $\{x_1,\ldots,x_N\}$, the pdf can be approximated by

$$\hat{f}_x(\xi) = \frac{1}{N}\sum_{i=1}^{N}\kappa(\xi - x_i, \sigma^2) \tag{4}$$

where $\kappa(\cdot, \sigma^2)$ is typically a zero-mean Gaussian kernel with standard deviation σ. One of the advantages of Parzen window with Gaussian kernel is that we can avoid the integral computation directly, since the integral of the product of two Gaussian kernels generates another Gaussian kernel with a different double standard deviation, that is, the convolution of two Gaussian kernels centered at a_i and a_j is a Gaussian centered at $a_i - a_j$ with covariance equal to the sum of the original covariance (Principe et al., 2000).

3.1.2. Continuous-valued pdf estimation

A Parzen pdf estimate is in general biased. The bias can be asymptotically reduced to zero by selecting an unimodal symmetric kernel function (such as the Gaussian) and reducing the kernel size monotonically with increasing number of samples, so that the kernel asymptotically approaches a Dirac-delta function. In the finite sample case, the kernel size must be selected according to a trade-off between estimation bias and variance: decreasing the kernel size increases the variance, whereas increasing the kernel size increases the bias. Therefore, selection of optimal kernel size is one of the important steps in the Parzen windowing method. In the classification problem of (3), the desired signal is the discrete-valued label, such as $+1$ and -1. The labels essentially just contain the information about the a priori probability of each class, and no pdf shape information. Therefore, the methodology of (3) is too much dependent upon the assumption of a priori probability and does not exploit the shape information of the pdfs during training and testing.

In this paper we propose an alternate criterion that uses the output of the classifier instead of the desired response because it contains much more information for pdf matching than the discrete value labels. The cost function becomes

$$\min_{\mathbf{w}} \int (f_{ytrn}(x) - f_{ytst}(x))^2\,\mathrm{d}x \tag{5}$$

where $f_{\mathrm{ytrn}}(x)$ is the pdf of system output during the training phase, $f_{\mathrm{ytst}}(x)$ is the pdf of the system output signal during the testing phase and $\mathbf{w}$ is the weight vector of the adaptive system. The system output is still low dimensional and preserves better the information of the training data set.

3.1.3. Algorithm details

Next, we derive a gradient descent algorithm for the cost function in (5). For convenience, we derive the algorithm in one dimension, but this can be easily extended to multidimensional cases.

$$\begin{aligned} J(\mathbf{w}) &= \int(\hat{f}_{ytrn}(x,y) - \hat{f}_{ytst}(x,y))^2\,\mathrm{d}x \\ &= \int\hat{f}^2_{ytrn}(x,y)\,\mathrm{d}x - 2\int\hat{f}_{ytrn}(x,y)\hat{f}_{ytst}(x,y)\,\mathrm{d}x \\ &\quad + \int\hat{f}^2_{ytst}(x,y)\,\mathrm{d}x \\ &= J_0 + J_1(\mathbf{w}) + J_2(\mathbf{w}) \end{aligned} \tag{6}$$

where

$$\begin{aligned} J_0 &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\kappa(\tilde{y}_i - \tilde{y}_j, 2\sigma^2), \\ J_1(\mathbf{w}) &= -\frac{2}{\mathrm{NM}}\sum_{i=1}^{N}\sum_{j=1}^{M}\kappa(\tilde{y}_i - y_j(\mathbf{w}), 2\sigma^2), \\ J_2(\mathbf{w}) &= -\frac{1}{M^2}\sum_{i=1}^{N}\sum_{j=1}^{M}\kappa(y_i(\mathbf{w}) - y_j(\mathbf{w}), 2\sigma^2) \end{aligned} \tag{7}$$

J_0 is not a function of $\mathbf{w}$, $\tilde{y}$ is the system output signal during the training, N is the number of the training samples and y is the system output during the testing phase with M samples. Only J_1 and J_2 are functions of $\mathbf{w}$ through $y = g(\mathbf{u}, \mathbf{w})$. In order to adjust the weights in the testing phase, we take the derivative of J with respect to $\mathbf{w}$ to obtain the gradient descent update

$$\mathbf{w}_{\mathrm{new}} = \mathbf{w}_{\mathrm{old}} - \eta\nabla J(\mathbf{w}) \tag{8}$$

where the gradient is evaluated from

$$\begin{aligned} \nabla J(\mathbf{w}) &= \nabla J_1(\mathbf{w}) + \nabla J_2(\mathbf{w}), \\ \nabla J_1(\mathbf{w}) &= \frac{2}{\mathrm{NM}}\sum_{i=1}^{N}\sum_{j=1}^{M}\kappa'(\tilde{y}_i - y_j, 2\sigma^2)\frac{\mathrm{d}y_j}{\mathrm{d}\mathbf{w}}, \\ \nabla J_2(\mathbf{w}) &= \frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}\kappa'(y_i - y_j, 2\sigma^2)\frac{d(y_i - y_j)}{\mathrm{d}\mathbf{w}} \end{aligned} \tag{9}$$

A batch method is used here to compute the weights update. An online approach is also possible with the introduction of stochastic information gradient (Erdogmus, Principe, & Hild, 2003).

3.2. Boosting-like algorithm

The hypothesis of our Euclidean distance pdf matching algorithm for the classification application is still based on the assumption that the a priori probability of the testing class is unchanged from the training class. However, in practical applications, we do not know the true mix of classes in the testing data. Unfortunately, a change of a priori probabilities between training and testing has in general an important impact on the decision boundary obtained by the proposed method. In order to minimize the effect of the a priori probability, we propose a boosting-like iterative procedure with the criterion in (5). Similar to the general boosting algorithm, our method consists of two steps: (1) generate several decision boundaries with the procedure outlined above with different random subsets of samples from the testing data. The test data subsets are matched for the same a priori probability according to the classifier obtained in the training set. (2) combine the collection of individual decision boundaries. We describe our algorithm for the two-classification problem in Algorithm 1.

Summary of the boosting-like procedure

1	Train the adaptive system with the labeled data
2	After the training phase, bypass all the unlabeled data through the classifier and count the numbers of each classes (called pseudo classes), i.e. estimate the a priori probabilities of pseudo classes based on the decision boundary obtained from the training data
3	Pick new testing samples randomly so that a priori probabilities are the same as those for the training data
4	Update the adaptive system weights with our ITL criterion and generate a new decision boundary
5	Bypass all the unlabeled data through the new classifier and estimate the a priori probabilities of pseudo classes based on the decision boundary obtained from previous step
6	Iterate (3) to (5) until there is not much change in the decision boundary
7	Combine a collection of decision boundaries

Theoretical analysis of the ensemble classifier was provided by the bias/variance decomposition in (Naftaly, Intrator, & Horn, 1997). Let $F(\mathbf{u})$ denote the input–output function realized by the network and $F_I(\mathbf{u})$ denote the average of the input–output function of the expert network over different initial conditions. Then the following results hold

$$E_D[F(\mathbf{u})^2] \geq E_{D'}[F_I(\mathbf{u})^2] \tag{10}$$

$$V_{D'}(F_I(\mathbf{u}) \leq V_D(F(\mathbf{u})) \tag{11}$$

where D is the space encompassing the distribution of all data set and the distribution of all initial conditions and D' is the remnant space. Eq. (10) and (11) imply that the variance is reduced by ensemble averaging the experts over the initial conditions, leaving the bias unchanged.

In our proposed iterative boosting-like method, the classifier is trained on the testing data with different data subsets, just like the Boosting method.

4. Discussion

Our methodology requires that the a priori probability of each class remains the same during the training and application phase, but it allows for slight differences between the pdf of the training and testing. To examine further, we employ the Bayesian method to find out under what conditions performance can be improved. Consider a classification problem with two classes, C_1 and C_2 with the a priori probability $P(C_1)$ and $P(C_1)$ respectively. The optimal discriminant function separating the two classes is given by

$$P(y = C_i|\mathbf{u}) = \frac{P(\mathbf{u}|y = C_i)P(y = C_i)}{P(\mathbf{u})}, \quad i = 1,2 \tag{12}$$

Since we use the probability of output data for training as a pseudo desired signal in the testing phase in order to continue updating the classifier weights, we have to assume that priors do not change from the training to the testing set, i.e. $P(C_i;\ \mathit{training}) = P(C_i;\ \mathit{testing})$ for $i = 1, 2$. In the case that the likelihood functions $P(\mathbf{u}|y = C_i)$ change from the training to the testing set, obviously the optimal decision boundary obtained from the training set will not be the optimal for the testing set. Under this condition, our algorithm will adjust the decision boundary to a better position such that the correct classification probability increases. In the case that the likelihood functions do not change from the training to the testing set, then according to the Bayesian Eq. (12) the optimal decision boundary obtained from the training phase will remain optimal, since the value of the a posterior probability remains the same. Then there is no need to apply our algorithm. We will illustrate these two cases in the next simulation section.

5. Simulation results

In this section, we will give the simulation results of the proposed boosting-like algorithm with the Euclidean distance pdf matching criterion in a simple pattern classification problem as well as a real biomedical classification problem. In the simulation, we used the same simple neural network topology for the training and testing. The objective of the experiment is to distinguish (classify) two classes of overlapping two-dimensional patterns labeled class 1 and class 2. Let C_1 and C_2 denote the set of events for which a random vector $\mathbf{u}$ belong to Class 1 and 2, respectively. In the first simulation, we artificially generated two classes with conditional pdf for class 1, $P(\mathbf{u}|C_1)$, being a Gaussian distributed with zero mean and unit variance; the conditional pdf for class 2, $P(\mathbf{u}|C_2)$, has mean vector [2,2] and unit variance respectively. The two classes have equal a priori probabilities.($P(C_1;\ \mathit{training}) = P(C_2;\ \mathit{training}) = 0.5$)

In order to illustrate the effectiveness of our algorithm, we simulated with four different cases. In the first case, the conditional distributions of the testing pattern 1 and 2 are the same as those of the training patterns and a prior probabilities of the testing classes are equal to the training, i.e. $P(C_1;\ training)=P(C_2;\ training)=0.5$. Ideally, the decision boundary will not change after we apply our algorithm since the decision boundary is optimal both for the training and testing data sets. But due to the finite sample data effects, the likelihood functions will differ slightly for both simulations and the decision boundary will not be optimal during the testing. In Fig. 1, the dashed line is the decision boundary for the training data set and the solid line is the one after we applied the algorithm for the unlabeled data set (this assignment will be kept for all other figures). The figure shows that the decision boundary resulting from the training data moves slightly in the testing data after we applied our algorithm. The correct classification probability increases from 0.915 to 0.927. In this situation, our algorithm is able to improve generalization by counteracting overfitting. In the second case, we generated another testing data set of slightly different likelihood functions compared with the training data to mimic the normal variability between experimental conditions. For the testing data class 1 is a gaussian with mean vector [0.2,0.2] and its variance is 1.2, while class 2 is a gaussian with mean vector [2.1,2.1] and variance is 0.8. $P(C_1;\ testing)=P(C_2;\ testing)=0.5$. The results are given in Fig. 2 and 3. Fig. 2 plots the decision boundaries with and without continuous training in the testing data set. The dashdot line is the boundary at the last iteration stage of our boosting-like algorithm and the solid line is the combined decision boundary by averaging a collection (20) of boundaries obtained during iteration. We can see that the decision boundary shifts to a new position where the correct classification probability increases. The correct classification probability increases from 0.792 to 0.883 and 0.85(combined) by applying our algorithm. The third testing data shares the same distribution with the training data, only a priori probabilities of each testing class are different. $P(C_1;\ testing)=0.3$ and $P(C_2;\ testing)=0.7$. The result in Fig. 4, based on Bayes theory, the optimal boundary should slightly move toward class 1 by the a prior probability ratio. In Fig. 4, our method indeed changes the decision boundary to balance the effect of the a priori probability. The correct classification probability is increased slightly from 0.908 to 0.911. In the case boosting is not used, the proposed criterion may worsen performance since the a priori probability hypothesis for our Euclidean distance pdf matching algorithm is not satisfied. Therefore, our boosting-like procedure seems to deal with the effect of different a priori probabilities. The fourth case has a different distribution and a different a priori probability. The distribution is the same as the second simulation, but the a priori is 0.3 for class 1 and 0.7 for class 2. In this case,

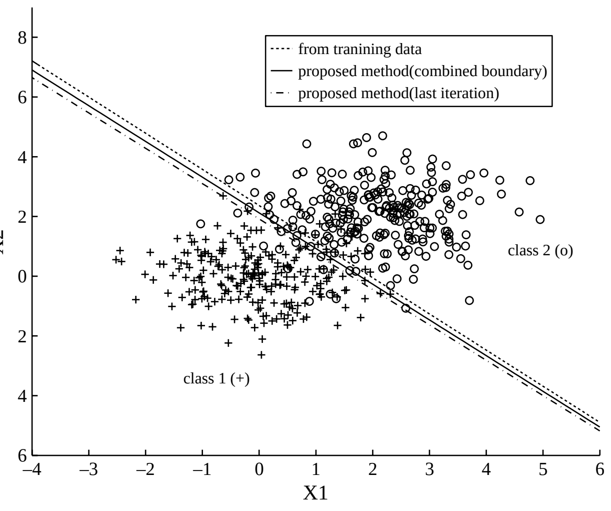

Fig. 2. Case2: Decision boundaries with and without continuous training in testing data set in the case $P(\mathbf{u}|C_i;\ training)\neq P(\mathbf{u}|C_i;\ testing)$ $P(C_1;\ testing)=P(C_2;\ testing)=0.5$ for simulation 1.

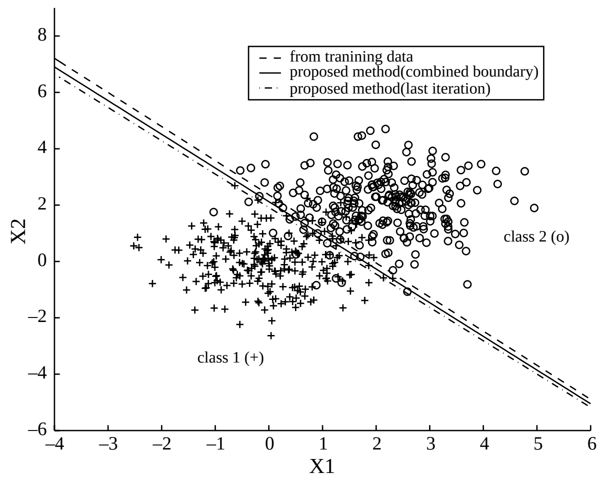

Fig. 1. Case1: Decision boundaries with and without continuous training in testing data set in the case $P(\mathbf{u}|C_i;\ training)=P(\mathbf{u}|C_i;\ testing)$, $P(C_1;\ testing)=P(C_2;\ testing)=0.5$ for simulation 1.

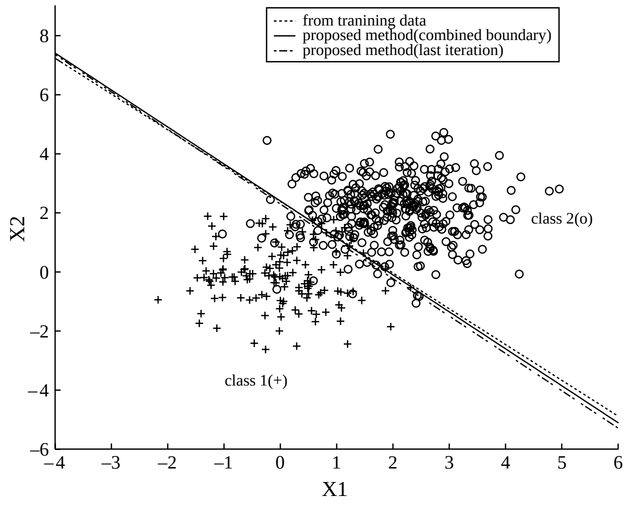

Fig. 3. Case3: Decision boundaries with and without continuous training in testing data set in the case $P(\mathbf{u}|C_i;\ training)=P(\mathbf{u}|C_i;\ testing)$ $P(C_1;\ testing)=0.3, P(C_2;\ testing)=0.7$ for simulation 1.

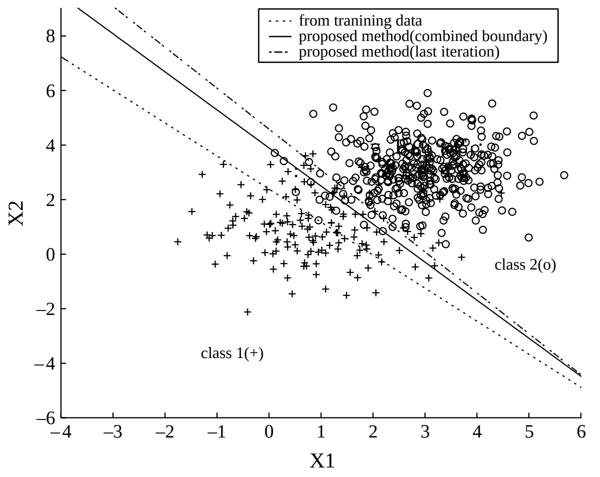

Fig. 4. Case4: Decision boundaries with and without continuous training in testing data set in the case $P(\mathbf{u}|C_i;\ training) \neq P(\mathbf{u}|C_i;\ testing)$ $P(C_1;\ testing)=0.3$,$P(C_2;\ testing)=0.7$ for simulation 1.

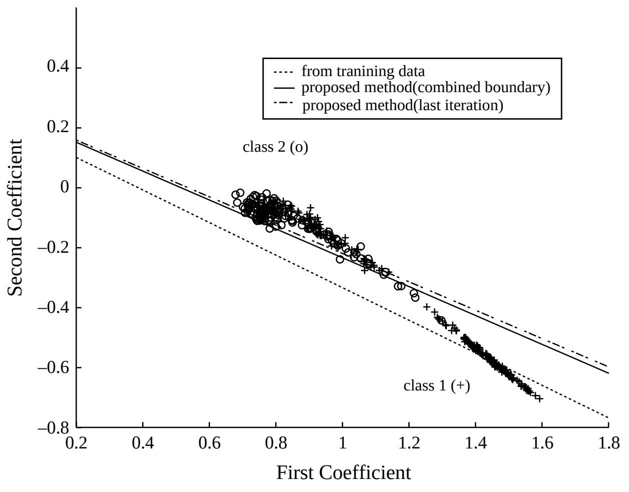

Fig. 6. Test patient1: Decision boundaries with and without continuous training in testing data set for patient 1 in simulation 2.

the correct classification probability is increased from 0.878 to 0.933 and 0.929(combined).

In the second simulation, we apply the proposed our algorithm to the classification of a biomedical data set. This data set was obtained from neural recordings in the surgical treatment of Parkinson's disease. Spike trains are collected from thalamic (Thal) and subthalamic nucleus (STN) cellular activity using deep brain stimulation. The objective of classification is to distinguish the two classes Thal and STN. A second order autoregressive(AR) model is applied to segments of the neural activity as a feature extractor (Sanchez, Pukala, Principe, Bova, & Okun, 2005). The classifier uses the weights of the AR model. Since Thal and STN signals are different across patients, the conditional probability functions for the training and testing will differ. The simulation results are presented in Figs. 5–8 with class 1 for STN and class 2 for Thal. Fig. 5 shows the scatter plot of the two classes and the decision boundary obtained from a perceptron in the training set comprising the three patients. The total number of the training set is 12198 samples and the a priori probabilities of each classes are $P(C_1;\ training)=0.6002$, $P(C_2;\ training)=0.4008$. The correct classification probability after the conventional training is 0.7537. Then we tested our algorithm in each of the different patients and the results are given in Figs. 6–8, respectively. The number of the testing set in each test patients are 1976, 2148 and 4086 samples respectively. The a priori probabilities of each testing classes are $P(C_1;\ patient1)=0.622$, $P(C_2;\ patient1)=0.378$, $P(C_1;\ patient2)=0.7072$, $P(C_2;\ patient2)=0.2928$, $P(C_1;\ patient3)=0.665$, $P(C_2;\ patient3)=0.335$, respectively.

Since the individual patient distributions as well as the a priori probabilities are different from those of the training set, the performance of the conventional

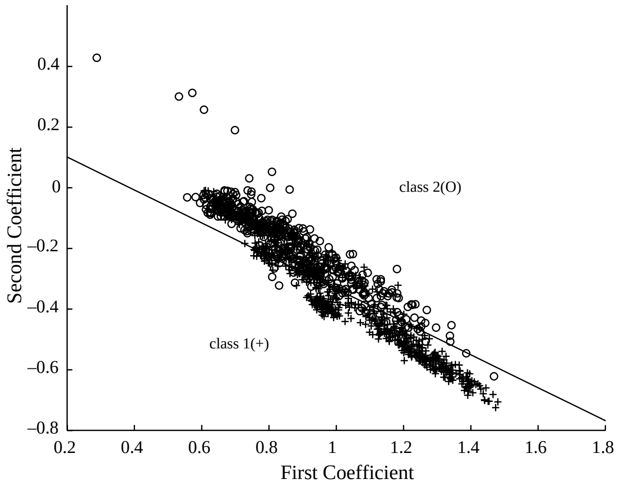

Fig. 5. Training data set and decision boundary in simulation 2.

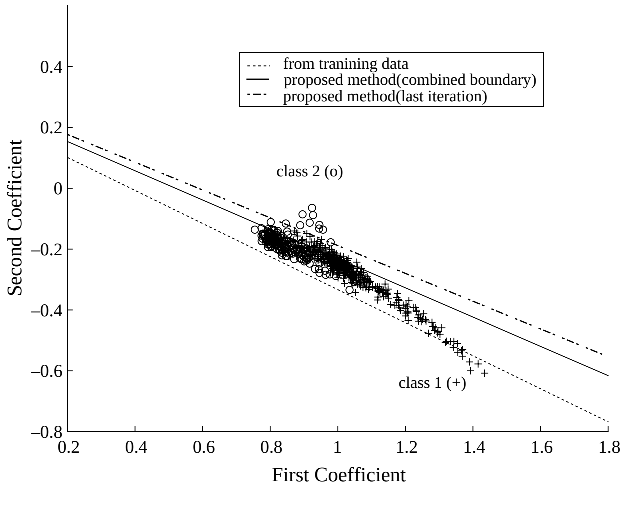

Fig. 7. Test patient 2: Decision boundaries with and without continuous training in testing data set for patient 2 in simulation 2.

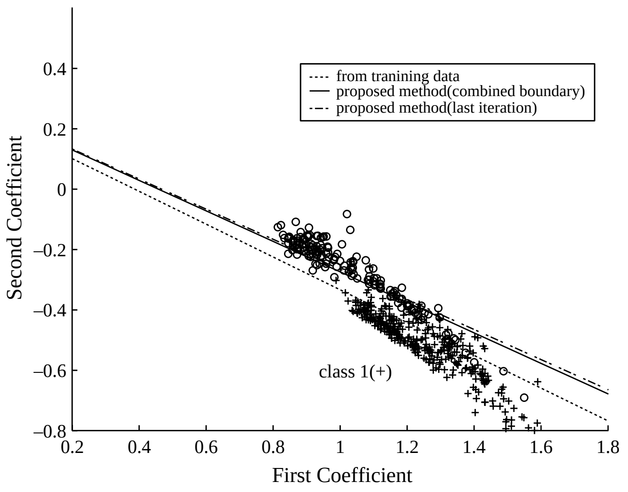

Fig. 8. Test patient 3: Decision boundaries with and without continuous training in testing data set for patient 3 in simulation 2.

Table 1
Comparison of correct classification probabilities

Testing data	Only test	Our algorithm with boosting-like (combined)	Our algorithm without boosting-like
Test patient 1	0.583	0.7646 (0.7758)	0.7844
Test patient 2	0.3073	0.7211 (0.6550)	0.5601
Test patient 3	0.8823	0.8872 (0.9033)	0.8921

An abbreviated version of some portions of this article appeared in (Jeong et al., 2005), published under the IEEE copyright.

classifier in the testing phase is not as good as the overall training result. Then we apply our algorithm with and without boosting (20 times, data sub sets of 20, respectively) and the results are shown in Table 1. From Table 1, we see that the classification performance can be improved by applying our method either with boosting-like algorithm or without it in this biomedical data application. The simulation in the real biomedical data illustrates the decision boundaries shift so that the overall correct classification probabilities increase after applying the Euclidean distance pdf matching algorithm.

6. Conclusions

This paper proposed a new information theoretic approach for training an adaptive system using unlabeled data even after supervised learning is completed. This so called the Euclidean distance pdf matching algorithm utilizes the information of the training system output and outputs of the unlabeled input during the testing phase, which provides a straightforward method to adjust system weights for better performance. For classification, the method requires preservation of a priori probability for each class during both the training and testing, which may be unrealistic in many applications. The simulations on the artificial data set and the real biomedical data suggest that the Euclidean distance pdf matching algorithm can improve classification performance. For future research, we should provide a theoretical treatment of the method, and systematic ways to combine a collection of decision boundaries and stop criterion for the boosting-like algorithm.

Acknowledgements

This work was supported in part by the National Science Foundation under grant ECS-0300340, Graduate Alumni Fellowship from University of Florida and Korea Science and Engineering Foundation under grant M06-2003-000-10146-0. We thank Dr Michael Okun for letting us use the data set collected in the University of Florida Movement Clinic.

References

Belkin, M., Niyogi, P., Sindhwani, V. (2004). Manifold regularization: A geometric framework for learning from examples, technical report tr-2004-06, Department of computer university of Chicago. *Technical Report* TR-2004-06.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In: *Proceedings of the conference on computational learning theory*, pp. 92–100.

Cristianini, N., & Taylor, J. S. (2000). *An introduction to support vector machines*. Cambridge: Cambridge University Press.

Erdogmus, D., Principe, J. C., & Hild, K. E. (2003). On-line entropy manipulation: Stochastic information gradient. *Signal Processing Letters*, *10*(8), 242–245.

Erdogmus, D., Rao, Y., & Principe, J.C. (2005). Supervised training of adaptive systems with partially labeled data. In *Proceedings of the international conference on accoustic, speech and signal processing*.

Gammerman, A., Vapnik, V., & Vowk, V. (1998). Learning by transduction. In: *Proceedings of the conference on uncertainty in artificial intelligence*. pp. 148–155.

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation* (2nd ed). New Jersey: Prentice Hall.

Jeong, K. H., Xu, J. W., & Principe, J. C. (2005). An information theoretic approach to adaptive system training using unlabeled data. In: *Proceedings of the international joint conference on neural networks*, Montreal, Canada.

Naftaly, U., Intrator, N., & Horn, D. (1997). Optimal ensemble averaging of neural networks. *Neural Network*, *8*, 283–296.

Nigram, K., McCallum, A., Thrum, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *In Machine Learning Theory, volume*, *39*, 103–134.

Novak, B. (2004). Use of unlabeled data in supervised machine learning. In: *Proceedings of the conference on data mining and warehousers*pp. 834–846.

Parzen, E. (1962). On the estimation of probability density function and mode. *The Annals of Mathematical Statistics.*.

Principe, J. C., Xu, D., & Fisher, J. (2000). Information theoretic learning. In S. Haykin (Ed.), *Unsupervised Adaptive Filtering* (pp. 265–319). New York: Wiley, 265–319.

Sanchez, J., Pukala, J., Principe, J.C., Bova, F., & Okun, M. (2005). Linear predictive analysis for targeting the basal ganglia in deep brain stimulation surgeries. In: *Proceedings of the conference on 2nd int ieee workshop on neural engineering*.

Saunders, C., Gammerman, A., Vowk, V. (1999). Transduction with confidence and credibility. In: *Proceedings of the conference on uncertainty in artificial intelligence.* pp. 722–726.

Widrow, B., & Stearns, S. D. (1985). *Adaptive Signal Processing*. New Jersey: Prentice Hall.