Multiclass Cancer Classification Using Semisupervised Ellipsoid ARTMAP and Particle Swarm Optimization with Gene Expression Data

Rui Xu, Georgios C. Anagnostopoulos, and Donald C. Wunsch II

Abstract—It is crucial for cancer diagnosis and treatment to accurately identify the site of origin of a tumor. With the emergence and rapid advancement of DNA microarray technologies, constructing gene expression profiles for different cancer types has already become a promising means for cancer classification. In addition to research on binary classification such as normal versus tumor samples, which attracts numerous efforts from a variety of disciplines, the discrimination of multiple tumor types is also important. Meanwhile, the selection of genes which are relevant to a certain cancer type not only improves the performance of the classifiers, but also provides molecular insights for treatment and drug development. Here, we use Semisupervised Ellipsoid ARTMAP (ssEAM) for multiclass cancer discrimination and particle swarm optimization for informative gene selection. ssEAM is a neural network architecture rooted in Adaptive Resonance Theory and suitable for classification tasks. ssEAM features fast, stable, and finite learning and creates hyperellipsoidal clusters, inducing complex nonlinear decision boundaries. PSO is an evolutionary algorithm-based technique for global optimization. A discrete binary version of PSO is employed to indicate whether genes are chosen or not. The effectiveness of ssEAM/PSO for multiclass cancer diagnosis is demonstrated by testing it on three publicly available multiple-class cancer data sets. ssEAM/PSO achieves competitive performance on all these data sets, with results comparable to or better than those obtained by other classifiers.

Index Terms—Cancer classification, gene expression profile, semisupervised ellipsoid ARTMAP, particle swarm optimization.

1 Introduction

With the emergence and rapid advancement of DNA microarray technologies, including cDNA and high-density oligonucleotide microarray [23], [37], cancer classification through identification of the corresponding gene expression profiles has already attracted numerous efforts from a wide variety of research communities. Cancer classification is important for subsequent diagnosis and treatment. Without the correct identification of cancer types, it is rarely possible to provide useful therapies and achieve expected treatment effects. Traditional classification methods are largely dependent on the morphological appearance of tumors, parameters derived from clinical observations, and other biochemical techniques. Their applications are limited by the existing uncertainties and their prediction accuracy is very low [1], [25]. Tumors with a similar appearance may have quite different origins and, therefore, respond differently to the same treatment therapy. For example, for diffuse large B-cell lymphoma (DLBCL), almost half of the clinical cases fail the pharmaceutical treatment due to the existence of unknown subtypes that cannot be discriminated by their morphologic parameters [1]. DNA microarray technologies offer cancer researchers a new method to investigate the pathologies of cancer, from a molecular angle, under a systematic framework, and further, to make more accurate predictions in prognosis and treatment.

Along with the opportunity brought by the microarray technologies, new challenges have also appeared, such as high dimensionality, small samples, and inherent noise. These factors require the proposed computational analysis methods to have corresponding solving mechanisms. Research on binary cancer classification through gene expression profiles has already been reported, with promising results. Golub et al. described cancer classification as two challenges, class discovery and class prediction, and used several strategies, including weighted voting, neighborhood analysis, and self-organizing feature maps (SOFMs), to discriminate two types of human acute leukemias (ALL versus AML) [25]. Alizadeh et al. distinguished two molecularly distinct subtypes of diffuse large B-cell lymphoma with centroid average hierarchical clustering [1]. Other explorations include colon cancer [2], cutaneous melanoma [11], ovarian cancer [47], breast cancer [43], [54], and lung cancer [53], to name a few.

In practice, it is common to discriminate more than two types of cancers [20], [32], [35], [40], [41], [44], [45], [51]. Ramaswamy et al. divided the multiclass problem into a series of binary classification subproblems through the

- *R. Xu and D.C. Wunsch II are with the Applied Computational Intelligence Laboratory, Department of Electrical and Computer Engineering, University of Missouri-Rolla, Rolla, MO 65409-0249. E-mail: {rxu, dwunsch}@umr.edu.*
- *G.C. Anagnostopoulos is with the Department of Electrical and Computer Engineering, Florida Institute of Technology, Melbourne, FL 32901-6975. E-mail: georgio@fit.edu.*

Manuscript received 14 Apr. 2005; revised 16 Sept. 2005; accepted 3 Nov. 2005; published online 9 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0028-0405.
Digital Object Indentifier No. 10.1109/TCBB.2007.1009.

one-versus-all or all-pairs approach and employed support vector machines, weighted voting, and k-nearest-neighbors methods to distinguish 14 different tumor types [44]. Khan et al. trained multilayer perceptrons to categorize small round blue-cell tumors (SRBCTs) with four subclasses [32]. A nearest shrunken centroid method was proposed by Tibshirani et al. and was tested on the SRBCT data set with 100 percent accuracy [51]. Furthermore, Scherf et al. constructed a gene expression database to study the relationship between genes and drugs for 60 human cancer cell lines originating from 10 different tumors, which provides an important criterion for therapy selection and drug discovery [45]. Although these methods manifest interesting performance for some cancer data sets, their classification accuracy deteriorates dramatically with the increasing number of classes in the data sets. For example, in a comparative study by Li et al., who investigated the performance of decision trees, naive Bayes, support vector machines, and k-nearest-neighbor, together with six different feature selection methods, the classification accuracy for the NCI60 data set is only below 70 percent [36].

One of the major challenges of microarray data analysis is the overwhelming number of measures of gene expression levels compared with the small number of samples. This is known as the curse of dimensionality in machine learning, which is introduced to indicate the exponential growth in computational complexity and the demand for more samples as a result of high dimensionality in the feature space [22]. Not all of these genes (features) are relevant to the discrimination of tumors and, often, only a small part of them is enough for effective classification [19], [25], [41], [50]. The existence of numerous genes that do not contribute to the distinction in data sets not only increases the computational complexity, but impairs the analysis of the relevant ones. Furthermore, cancer research requires identifying the relation of tumors and their causes at the molecular level, which is imperative in determining the appropriate therapy. Therefore, feature selection or extraction, also known as *informative gene selection*, is critically important. Principal component analysis (PCA) is a widely used tool for dimensionality reduction, which attempts to seek the projection that best interprets the variation of the data [22]. PCA has already been used in some applications on gene expression data [24]. However, according to the empirical results by Yeung and Ruzzo, PCA cannot always find the correct structure with just the first few principal components [60]. Other methods are generally based on ranking genes according to their expression differentiation under two different classes, examples including signal-to-noise ratio [25], Fisher discriminant score [29], t-statistics score [40], and nonparametric test statistics like the TNoM score [9] and Park score [42]. Considering the possibility that the feature-rank-based methods may pick out many highly correlated genes that will affect the classification accuracy, clustering techniques are utilized to group genes with similar profiles in order to decrease redundancy [29], [38]. These criteria can usually achieve some meaningful insights for binary classification; however, they do not work well for the multiclass discrimination problem due to the increasing complexity.

In order to address the insufficiency of the existing methods and provide a more effective method to analyze complex data sets, like the NCI60 data set, here, we use a combination of semisupervised Ellipsoid ARTMAP (ssEAM), formerly known as Boosted Ellipsoid ARTMAP [6], with particle swarm optimization (PSO) for multiclass cancer discrimination and gene selection. ssEAM is based on Adaptive Resonance Theory (ART) [26], which was inspired by neural modeling research and was developed as a solution to the *plasticity-stability dilemma*: How adaptable (plastic) should a learning system be so that it does not suffer from catastrophic forgetting of previously learned rules (stability). Coming as an enhancement and generalization of Ellipsoid ART (EA) and Ellipsoid ARTMAP (EAM) [4], [5], which, in turn, follow the same learning and functional principles of Fuzzy ART (FA) [15] and Fuzzy ARTMAP (FAM) [14], ssEAM is capable of learning associative maps between clusters of an input and an output space. As a special case, when the output space is a set of class labels, ssEAM can be used as a classifier. ssEAM features fast, stable, and finite learning and creates hyperellipsoidal clusters that induce complex nonlinear decision boundaries. On the other hand, PSO is an evolutionary computation technique for global optimization which is based on the simulation of complex social behavior [31]. A random velocity is associated with each potential solution, which is considered to "be flown through the problem space" [30]. Also, PSO is implemented with a memory mechanism, which can retain the information of previous best solutions that may be lost with the population evolution in other evolutionary techniques. Herein, we demonstrate the potential of ssEAM/PSO in addressing massive, multidimensional gene expression data through analyzing three publicly available cancer data sets: the NCI60 data set [45], the acute leukemia data set [25], and the acute lymphoblastic leukemia (ALL) data set [59]. ssEAM/PSO achieves competitive performance on all three data sets and the results are comparable to or better than those obtained by other classifiers.

The paper is organized as follows: Section 2 describes the ssEAM/PSO system for multiclass cancer discrimination. The results of experiments are presented and discussed in Section 3 and Section 4 concludes the paper.

2 Methods and Systems

2.1 EAM and Semisupervised EAM

The Ellipsoid ARTMAP classifier (EAM), a member of the ART family of neural architectures, accomplishes classification tasks by clustering data that are attributed with the same class label. The geometric representations of these clusters, which are called *categories*, are hyperellipsoids embedded in the feature space. A typical example of such a category representation, when the input space is two-dimensional, is provided in Fig. 1, where it is shown that each category j is described by its center location $\mathbf{m}_j$, its orientation $\mathbf{d}_j$, and a Mahalanobis radius M_j. The collection of the three aforementioned quantities is typically represented as the template vector $\mathbf{w}_j = [\mathbf{m}_j, \mathbf{d}_j, M_j]$ of category j. If we define the distance between an input pattern $\mathbf{x}$ and a category j as

$$\begin{aligned} dis(\mathbf{x}, \mathbf{w}_j) &= \max\{\| \mathbf{x} - \mathbf{m}_j \|_{\mathbf{c}_j}, M_j\} - M_j, \\ \| \mathbf{x} - \mathbf{m}_j \|_{\mathbf{c}_j} &= \sqrt{(\mathbf{x} - \mathbf{m}_j)^T \mathbf{C}_j (\mathbf{x} - \mathbf{m}_j)}, \end{aligned} \tag{1}$$

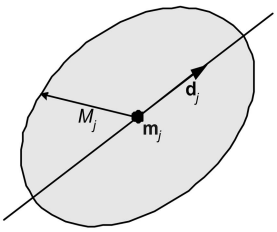


Fig. 1. Example of the geometric representation of an EAM category j when the feature space is two-dimensional. $\mathbf{m}_j$ is the center of the ellipsoid, $\mathbf{d}_j$ is the orientation vector, coinciding with the direction of the major axis of the ellipsoid, and M_j is the Mahalanobis radius of the category.

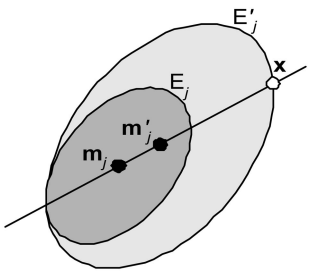


Fig. 2. Update of an EAM category j due to a training pattern $\mathbf{x}$ when the feature space is two-dimensional. The representation region expands to contain the original region and the new pattern.

where $\mathbf{C}_j$ is the category's shape matrix, defined as $\mathbf{C}_j = 1/\mu^2(\mathbf{I} - (1-\mu^2)\mathbf{d}_j\mathbf{d}_j^T)$, and μ is the ratio between the length of the hyperellipsoid's minor axes (with equal length) and major axis, the *representation region* of category j, which is the shaded area in the figure, can be defined as a set of points in the input space satisfying the condition

$$dis(\mathbf{x}, \mathbf{w}_j) = 0 \Rightarrow \| \mathbf{x} - \mathbf{m}_j \|_{\mathbf{c}_j} \leq M_j. \quad (2)$$

A category encodes whatever information the EAM classifier has learned about the presence of data and their associated class labels in the locality of its geometric representation. This information is encoded into the location and size of the hyperellipsoid. The latter feature is primarily controlled via the baseline vigilance $\overline{\rho} \in [0, 1]$ and indirectly via two additional network parameters, namely, the choice parameter $a > 0$ and a parameter $\omega \geq 0.5$ [5]. Typically, small values of $\overline{\rho}$ produce categories of larger size, while values close to 1 produce the opposite effect. As a special case, when $\overline{\rho} = 1$, EAM will create solely point categories (one for each training pattern) after completion of training and it implements the ordinary, Euclidian 1-Nearest Neighbor classification rule. A category's particular shape (eccentricity of its hyperellipsoid) is controlled via the network parameter $\mu \in (0, 1]$; for $\mu = 1$, the geometric representations become hyperspheres, in which case the network is called *Hypersphere ARTMAP* [3].

Learning in EAM occurs by creating new categories or updating already existing ones. If a training pattern $\mathbf{x}$ initiates the creation of a new category J, then J receives the class label $L(\mathbf{x})$ of $\mathbf{x}$ by setting the class label of J to $I(J) = L(\mathbf{x})$. The recently created category J is initially a *point category*, meaning that $\mathbf{m}_J = \mathbf{x}$ and $M_J = 0$. While training progresses, point categories are being updated due to the presentation of other training patterns and their representation regions may grow. Specifically, when it has been decided that a category j must be updated by a training pattern $\mathbf{x}$, its representation region expands so that it becomes the minimum-volume hyperellipsoid that contains the entire, original representation region and the new pattern. An example of this process for a two-dimensional feature space is depicted in Fig. 2, where the original representation region E_j expands to become E'_j. Learning eventually ceases in EAM when no additional categories are being created and the existing categories have expanded enough to capture all training data. Notice that, if $\mathbf{x}$ falls inside the representation region of j, no update occurs since j has already taken into account the presence of $\mathbf{x}$.

The procedure of deciding which category, j, is going to be updated, with a training pattern, $\mathbf{x}$, involves competition among preexisting categories. Let us define this set of categories as N as well as the set $S \subseteq N$ of all categories that are candidates in the competition; initially, $S = N$. Before the competition commences, for each category j, two quantities are calculated: the *category match function* (CMF) value

$$\rho(\mathbf{w}_j|\mathbf{x}) = \frac{D - 2M_j - dis(\mathbf{x}, \mathbf{w}_j)}{D}, \quad (3)$$

where D is a parameter greater than 0, and the *category choice function* (CCF) value

$$T(\mathbf{w}_j|\mathbf{x}) = \frac{D - 2M_j - dis(\mathbf{x}, \mathbf{w}_j)}{D - 2M_j + a}. \quad (4)$$

Next, EAM employs two category-filtering mechanisms: the *vigilance test* (VT) and the *commitment test* (CT). Both tests decide if the match between the pattern and the category's representation region is sufficient to assign that pattern to the cluster represented by the category in question. These tests can function as a novelty detection mechanism as well: If no category in S passes both tests, then $\mathbf{x}$ is not a typical pattern in comparison to the data experienced by the classifier in the past. Categories that do not pass these tests can subsequently be removed from the candidate set S. Next, the competition for $\mathbf{x}$ is won by the category J that features the maximum CCF value with respect to the pattern; in case of a tie, the category with the minimum index is chosen. The final verdict on whether to allow J to be updated with $\mathbf{x}$ or not is delivered by the *prediction test* (PT): J is allowed to be updated with $\mathbf{x}$ only if both J and $\mathbf{x}$ feature the same class label, that is, if $I(J) = L(\mathbf{x})$. If J fails the PT, a *match tracking* process is invoked by utilizing a stricter VT in the hope that another suitable EAM category will be found that passes all three tests. If the search eventually fails, a new point category will be created as described before. The reader is referred to [4] and [5] for a more detailed description of EAM's operation.

EAM does not allow categories to learn training patterns of dissimilar class labels. This property is ideal when the individual class distributions of the problem are relatively well separated. However, in the case of high class overlap, or when dealing with increased amount of noise in the feature domain, EAM will be forced to create many small-sized categories, a phenomenon called the *category proliferation problem*. Moreover, when EAM is trained in offline mode to perfection, its posttraining error will be zero, which can be viewed as a form of data overfitting.

The Semisupervised EAM classifier (ssEAM) extends the generalization capabilities of EAM by allowing the clustering into a single category of training patterns not necessarily belonging to the same class [6]. This is accomplished by augmenting EAM's PT in the following manner: A winning category J may be updated by a training pattern $\mathbf{x}$, even if $I(J) \neq L(\mathbf{x})$, as long as the following inequality holds:

$$w_{J,I(J)}/1 + \sum_{c=1}^{C} w_{J,c} \geq 1 - \varepsilon. \quad (5)$$

In (5), C denotes the number of distinct classes related to the classification problem at hand and the quantities $w_{j,c}$ contain the count of how many times category j was updated by a training pattern belonging to the cth class. In other words, (5) ensures that the percentage of training patterns that are allowed to update category J and carry a class label different from the class label $I(J)$ (the label that was initially assigned to J, when it was created) cannot exceed $100\varepsilon\%$, where $\varepsilon \in [0, 1]$ is a new network parameter, the *category prediction error tolerance*, which is specific only to ssEAM. For $\varepsilon = 1$, the modified PT will allow categories to be formed by clustering training patterns, regardless of their class labels, in an unsupervised manner. In contrast, with $\varepsilon = 0$, the modified PT will allow clustering (into a single category) only of training patterns belonging to the same class, which makes the category formation process fully supervised. Under these circumstances, ssEAM becomes equivalent to EAM. For intermediate values of ε, the category formation process is performed in a semisupervised fashion.

EAM and ssEAM feature a common performance phase, which is almost identical to their training phases. However, during the presentation of test patterns, no categories are created or updated. The predicted label for a test pattern $\mathbf{x}$ is determined by the *dominant class label* $D(J)$ of the winning category J, defined as:

$$\hat{L}(\mathbf{x}) = D(J) = \arg \max_{c=1..C} w_{J,c} = 1. \quad (6)$$

When $\varepsilon < 0.5$, ssEAM's PT guarantees that, throughout the training phase, $D(j) = I(j)$ for any category j. Fig. 3 provides pseudocode describing a single iteration of ssEAM's training and performance phase.

For $\varepsilon > 0$, ssEAM will, in general, display a nonzero posttraining error, which implies a departure from EAM's overfitting and category proliferation issues. For classification problems with noticeable class distribution overlap or noisy features, ssEAM with $\varepsilon > 0$ will control the generation of categories representing localized data distribution exceptions, thus improving the generalization capabilities of the resulting classifier. Most importantly, the latter quality is achieved by ssEAM without sacrificing any of the other valuable properties of EAM, that is, stable and finite learning, model transparency, and detection of atypical patterns.

ssEAM has many attractive properties for classification or clustering. First, ssEAM is capable of both online (incremental) and offline (batch) learning. Using *fast learning* [6] in offline mode, the network's training phase completes in a small number of epochs. The computational cost is relatively low and it can cope with large amounts of multidimensional data, maintaining efficiency. Moreover, ssEAM is an *exemplar-based model*, that is, during its training, the architecture summarizes data via the use of exemplars in order to accomplish its learning objective. Due to its exemplar-based nature, responses of an ssEAM architecture to specific test data are easily explainable, which makes ssEAM a *transparent learning model*. This fact contrasts with other, *opaque* neural network architectures for which it is difficult, in general, to explain why an input $\mathbf{x}$ produced a particular output $\mathbf{y}$. Another important feature of ssEAM is the capability of detecting atypical patters during either its training or performance phase. The detection of such patterns is accomplished via the employment of a match-based criterion that decides to which degree a particular pattern matches the characteristics of an already formed category in ssEAM. Additionally, via the utilization of hyperellipsoidal categories, ssEAM can learn complex decision boundaries that frequently arise in gene expression classification problems. Finally, ssEAM is far simpler to implement, for example, than Backpropagation for feed-forward neural networks and the training algorithm of Support Vector Machines. Many of these advantages are inherited, general properties of the ART family of neural networks: fast, exemplar-based, match-based, learning [13], transparent learning [27], capability of handling massive data sets [16], [28], and implementability in software and hardware [48], [55], [56]. Also, ART neural networks dynamically generate clusters without specifying the number of clusters in advance as the classical k-means algorithm asks for [58].

2.2 Particle Swarm Optimization

PSO is motivated by the behavior of bird flocking or fish schooling and originally intended to explore optimal or near-optimal solutions in sophisticated continuous spaces [31]. A random velocity is associated with each potential solution, called a particle in the swarm. These particles change their positions in the search space until a stop condition is satisfied. The basic idea of PSO is to accelerate each particle toward its corresponding $pbest$ and $gbest$ locations at each time step, where $pbest$ is the previous best solution for the particle, based on the calculated fitness value, and $gbest$ is the best overall value in the whole swarm. This concept is depicted in Fig. 4, in which $L(t)$ and $L(t+1)$ represent the locations at current and next time point, $V(t)$ and $V(t+1)$ represent the velocities at current and next time point, $W_I V(t)$ is the momentum part, $V_{pbest}(t)$ is the velocity according to $pbest$, and $V_{gbest}(t)$ is the velocity according to $gbest$. Compared to other evolutionary computational algorithms, PSO has many desirable characteristics. PSO is easy to implement, fast to achieve high-quality solutions, and has the flexibility in balancing global and local exploration. More important, the memory mechanism of PSO can keep track of previous best solutions and, therefore, avoid the possible loss of previously learned knowledge.

Since our goal is to choose important genes (features) from a large gene pool, we employ a discrete binary version of PSO [30]. The major change of the binary PSO comes from the reexplanation of the meaning of the particle velocity. Given a set of particles $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$, where N is the number of particles in the swarm, the velocity for the ith particle $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, where D is the number of dimensions in a particle, is represented as $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The possible values for each bit $x_{id} (1 \leq i \leq N, 1 \leq d \leq D)$ is either 1 or 0, which indicates whether the corresponding genes are

```
Set S:=N and ρ := ρ̄
If S=∅, set J:=none; otherwise
    Compute CMF values ρ(j|x)  ∀j∈S
    Perform Vigilance Test: S:=S-{j∈S| ρ(j|x) <ρ}
If S=∅, set J:=none; otherwise
    Compute CCF values T(j|x)  ∀j∈S
    Select winning category J := min arg max_{j∈S} T(j|x)
    Perform Commitment Test: If T(J|x) < T_u , set J:=none

If x is a test pattern (ssEAM is in its testing phase)
    Set L̂(x) := D(J) ≙ arg max_{c=1..C} w_{J,c}
   and proceed with the next test pattern.

If x is a training pattern (ssEAM is in its training phase)
   While J?none do
       Perform Prediction Test:
       If I(J) = L(x) or w_{J,I(J)} / (1 + Σ_{c=1}^{C} w_{J,c}) ≥ 1 − ε , update category J with pattern x,
          set w_{J,L(x)} := w_{J,L(x)} + 1 and exit the while-loop.
       Otherwise,
           Perform Match Tracking: Set ρ := ρ(J|x)
           Perform Vigilance Test: S:=S-{j∈S| ρ(j|x) ≤ρ}
           If S=∅, set J:=none;
           otherwise
               Select winning category J := min arg max_{j∈S} T(j|x)
               Perform Commitment Test:
               If T(J|x) < T_u , set J:=none
   If J =none
       Create a new point category K
       Set I(K):=L(x), w_{K,L(x)} := 1 and N:=N ∪{K}
    Proceed with next training pattern.
```

Fig. 3. Pseudocode describing a single iteration of the training and performance phase for the Semisupervised Ellipsoid ARTMAP classifier. When $\varepsilon = 0$, the pseudocode describes the operation of the Ellipsoid ARTMAP classifier.

selected or not for cancer discrimination (1 for selected and 0 for not selected). Its corresponding velocity v_{id} is explained as the probability that x_{id} takes the value of 1 and is squashed into the interval [0, 1] through a logistic function $S(v_{id}) = 1/(1 + \exp(-v_{id}))$. The basic procedure of binary PSO for gene selection is as follows:

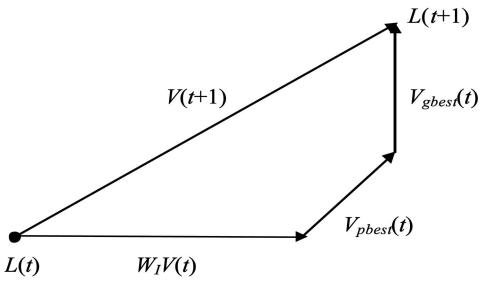


Fig. 4. Basic concept of the position change of a particle in PSO.

1. Initialize a population of N particles with random positions and velocities. The dimensionality D of the problem space is dependent on the number of genes in the microarray data.
2. Evaluate the classification performance of ssEAM and calculate the optimization fitness function for each particle. Here, the design of the fitness function aims to minimize the classification error and also favor the subset with fewer genes, which is defined as

$$f(\mathbf{x}_i) = Acc_{LOOCV} + 1/n, \tag{7}$$

where $Acc_{LOOCV} = \frac{C}{M} \times 100\%$ is the *leave one out cross validation* (LOOCV), also known as the *jackknife* approach, classification accuracy [22], where M is the total number of samples, C is the number of samples that are correctly classified, and n is the number of informative genes selected. More discussion on experiment design and performance evaluation of classifiers is provided in the following section.

3. Compare the fitness value of each particle with its $pbest$. If the current value is better than $pbest$, reset both the $pbest$ value and location to the current value and location.
4. Compare the fitness value of each particle with $gbest$. If the current value is better than $gbest$, reset $gbest$ to the current particle's array index and value.
5. Update the velocity and position of the particle with the following equations:

$$v_{id} = W_I \times v_{id} + c_1 \times rand_1 \times (pbest_{id} - x_{id}) + c_2 \times rand_2 \times (gbest_{id} - x_{id}), \quad (8)$$

$$x_{id} = \begin{cases} 1, & \text{if } rand_3 + \delta < S(v_{id}) \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

where x_{id} and v_{id} are the position and velocity of the dth dimensionality of the ith particle, respectively, W_I is the inertia weight, c_1 and c_2 are the acceleration constants, $rand_1$, $rand_2$, and $rand_3$ are uniform random functions in the range of [0, 1], δ is a parameter that limits the total number of genes selected to some certain range, and $S()$ is the sigmoid function. Compared with the original binary PSO in [30], we add the parameter δ in order to control the number of informative genes more flexibly.
6. Return to Step 2 until the stop condition is satisfied, usually a maximum number of iterations or high-quality solutions.

PSO has only four major parameters that need to be determined in advance. The inertial weight W_I specifies the trade-off between the global and local search. Larger values of W_I facilitate the global exploration, while lower values encourage local search. c_1 and c_2 are known as the cognition and social components, respectively, and are used to control the effects of a particle and its surrounding environment, which is achieved through adjusting the velocity toward $pbest$ and $gbest$. The velocity for each particle is restricted to a limit $V_{\max}$. During the evolutionary procedure, the velocity is reassigned to $V_{\max}$ if it exceeds$V_{\max}$. For binary PSO, this limits the probability that a bit in a particle takes on the value of one. Usually, the smaller $V_{\max}$ is, the higher the mutation rate [30]. Discussion on the parameter selection for PSO can be found in [21], [31], and [46].

2.3 Experiment Design

Since the data sets consist of only a small number of samples for each cancer type, it is important to choose an appropriate method to estimate the classification error of the classifier. In the experiment, we perform a double cross validation (*10-fold cross validation* (CV10) with LOOCV) instead of just the commonly used LOOCV to examine the performance of ssEAM/PSO. The reason lies in the fact that, although LOOCV error is unbiased, it has high variance, which is not preferred in cancer classification. A resampling strategy like bootstrap has lower variance. However, it may become largely biased for some data sets. Another consideration is the increasing computational cost [7], [33]. During the double cross validation procedure, the data set with N samples is divided into 10 mutually exclusive sets of approximately equal size, with each subset consisting of approximately the same proportions of labels as the original data set, known as stratified cross validation [33]. The classifier is trained 10 times, with a different subset left out as the test set and the other samples used to train the classifier at each time. During the training phase, gene selection is performed on 9 out 10 of the data (without considering the test data) in which LOOCV classification accuracy is used as the fitness function, as defined in (7). The prediction performance of the classifier is estimated by considering the average classification accuracy of the 10 cross-validation experiments, described as

$$Acc_{CV10} = \left(\frac{1}{N} \sum_{i=1}^{10} C_i \right) \times 100\%, \quad (10)$$

where C_i is the number of correctly classified samples. Previous study has shown the CV10 is more appropriate when considering the compromise between bias and variance [7], [33].

We also compare our approach with four other classifiers, i.e., multilayer perceptrons (MLPs), probabilistic neural networks (PNNs), learning vector quantization (LVQ), and k-nearest-neighbor (kNN). MLPs are being used as a processing system with their powerful capability in pattern recognition and nonlinear function approximation [22]. MLPs learn through the back-propagation algorithm, based on gradient descent. Here, we use a numerical optimization techniques-based variation, known as the one step secant algorithm, in order to achieve faster training. PNNs were introduced as an implementation of nonparametric Parzen window estimation with feed-forward neural network architecture and have the ability to approximate Bayesian optimal decision surfaces that can be arbitrarily complex [49]. LVQ is based on the concept of competitive neural networks and describes the potential data structure using the prototype vectors in the competitive layer [34]. We use the basic LVQ1 algorithm in our study. kNN is a nonparametic technique and assigns the label to a test sample based on the labels of the k nearest training samples [22]. We use Euclidean distance to determine the similarity between these samples. In the experiment, the value of k varies from 1 to 10 and we evaluate the average classification accuracy. All these methods have been reported in the literature for cancer classification with gene expression profiles [9], [10], [17], [18], [32].

For the above four methods, we use the Fisher discriminant criterion for informative genes selection, which is described as

$$D(i) = \frac{|\mu_+(i) - \mu_-(i)|^2}{\sigma_+^2(i) + \sigma_-^2(i)}, \quad (11)$$

where $\mu_+(i)$ and $\mu_-(i)$ are the mean values of gene i for the samples in class $+1$ and class -1, and $\sigma_+^2(i)$ and $\sigma_-^2(i)$ are the variances of gene i for the samples in class $+1$ and -1. The score aims to maximize the between-class difference and minimize the within-class spread. Other currently proposed rank-based criteria generally come from similar considerations and show similar performance [29]. Since our ultimate goal is to classify multiple types of cancer, we utilize a one-versus-all strategy to seek gene predictors. In other words, for a C-class prediction problem, we compare a particular class

with the other $C-1$ classes that are considered as a whole. We pick out genes according to the total score summed over all C comparisons, i.e., $\sum_{j=1}^{C} D_j(i)$, where $D_j(i)$ denotes the Fisher discriminant score for the ith gene at the jth comparison. Since microarray data generally are easily overfitted [7], [40], [54], we utilize the strategy that separates gene selection from cross validation operation and, therefore, overcomes the *selection bias*, which is caused by including the test samples in the process of feature selection and leads to an overoptimistic estimation of the performance for the classifier [7], [40]. Note that, in this case, the subsets of genes selected at each stage tend to be different.

3 Experimental Results

We test and analyze ssEAM/PSO performance in multiple cancer classification on the following three data sets:

3.1 NCI60 Data

The data set includes 1,416 gene expression profiles for 60 cell lines in a drug discovery screen by the National Cancer Institute [45]. These cell lines belong to nine different classes: eight breast (BR), six central nervous system (CNS), seven colorectal (CO), six leukemia (LE), nine lung (LC), eight melanoma (ME), six ovarian (OV), two prostate (PR), and eight renal (RE). Since the PR class only has two samples, they are excluded from further analysis. The gene expression levels are expressed as $-\log$ (red fluorescence/green fluorescence). There are 2,033 missing gene expression values in the data set, which are imputed by the method described by Berrar et al. [10], i.e., a missing value is estimated by the mean of all other existing values from the same class. This process leaves the final matrix in the form of $E = \{e_{i,j}\}_{58\times 1,409}$, where $e_{i,j}$ represents the expression level of gene j in tissue sample i. The original work by Scherf et al. with the average-linkage hierarchical clustering cannot effectively identify both BR and LC cancer cell lines [45]. Nguyen and Rocke employed the multivariate partial least squares, combined with polychotomous discrimination and quadratic discriminant analysis classification methods, to a subset of this data set and achieve the LOOCV error rate ranging from 5.7-42.9 percent when selection bias is considered [40]. Berrar et al. used probabilistic neural networks and achieved peak classification accuracy around 79 percent [10]. Li et al. compared the performance of decision tree, naive Bayes, support vector machines, and k-nearest-neighbor. Their hold-out classification accuracy for the NCI60 data set is less than 70 percent [36].

We set the parameters for ssEAM as follows: $\mu = 0.3$, $\rho = 0.4$, $\alpha = 2.5$, learning rate equal to 0.8, and adjust the value of e, which controls the amount of misclassification allowed in the training phase. The parameters of ssEAM are set based on a simple selection procedure in which the data set is randomly divided into training and validation sets. We compare the different parameter combination and choose the ones that lead to relatively better performance. The parameters W_I, c_1, and c_2 of PSO are set as 0.8, 2, and 2, respectively, which are the typical values recommended in the literature [21], [46]. The parameter δ controls the total number of genes selected in the subsets, and we run the program with δ at 0.5, 0.45, 0.4, 0.3, 0.2, 0.1, and 0.0. Each time, the evolution is processed for 300 generations with 50 particles included in the swarm. We ran the algorithm 20 times with different divisions of the data set and the performance is discussed based on the averages. The mean and standard deviation of the classification accuracies from the 20 runs are summarized in Table 1 and the best results are depicted in Fig. 5a. For the purpose of comparison, we also show the results of PNN, MLP, kNN, and LVQ1 in which the Fisher criterion is used for gene selection. For PNN, the smoothing parameter of the Gaussian kernel is set to 1. The MLP includes 20 nodes in the hidden layer with the sigmoid function as the transfer function. The number of prototypes in LVQ1 varies from 8 to 17 and we evaluate the average performance. From the table, we can see that ssEAM/PSO is superior to other methods used in our experiments or found in the literature. Specifically, the best result we obtain with ssEAM/PSO is 87.9 percent (79 genes are selected by PSO), which is better than other results reported in the literature. The confusion matrix is shown in Table 2, in which the numbers along the diagonal indicate the correct assignment of cancer samples by ssEAM. We see that ssEAM can achieve high classification accuracy for most of cancer types, particularly, 100 percent classification rates for central nervous system, colorectal, and leukemia. The worst performance is for ovarian cancer, namely, two out of six samples are misclassified. The best classification accuracy for PNN, kNN, and LVQ1 is 79.3 percent, 75.9 percent, and 75.9 percent, respectively. On the other hand, the best trained MLP architecture can only achieve 60 percent classification accuracy on this data set and the mean performance is less than 50 percent. We performed the t-test to compare the difference between the best overall results of ssEAM and other methods. All p-values are less than 10^{-15}, which indicates the classification accuracy for ssEAM is statistically better than those of other methods, at a 5 percent significance level. The same conclusion can also be obtained from nonparametric Wilcoxon rank test and Kruskal-Wallis test.

We compare the top 100 genes selected by the Fisher criterion with those selected by PSO. We find that there is only a small fraction of overlap between the genes chosen by these two methods. For example, for the 79 genes that lead to the best classification result, only seven are also selected by the Fisher criterion. Although the Fisher criterion can work well in binary classification [57], it does not achieve effective performance in the multiclass discrimination case. The reason may lie in the fact that the criterion tends to choose many highly correlated genes, ignoring genes that are really important in classification. Also, the use of the Fisher discriminant criterion is justified when the data follow an approximately Gaussian distribution. This may not be true for this data set. To examine the consistency of the feature selection method, we run the program 10 times with δ set to 0.45 (usually, 60-120 genes are selected). Still, each swarm includes 50 particles and evolves for 300 generations. We calculate the frequency of each gene appearing in the subset chosen in each particle (500 subsets in total). We find that selection frequencies of 14 genes are more than 20 percent and 32 genes are more than 15 percent. In particular, the highest frequency is 50.8 percent. This shows that PSO tends to choose important

TABLE 1
Classification Accuracy for the NCI60 Data Set

NCI60		Features (Genes)						
		10	79	135	252	385	555	695
EAM (ε =0)	PSO	65.52(1.86)	83.02(1.51)	79.40(1.42)	76.64(2.47)	71.21(1.59)	68.10(1.90)	67.76(1.49)
ssEAM (ε =0.1)	PSO	65.78(2.19)	84.66(1.36)	81.12(1.98)	78.62(1.89)	75.26(1.79)	73.10(2.27)	72.50(2.34)
PNN	Fisher Criterion	24.05(2.27)	71.12(2.23)	72.24(2.09)	74.65(2.02)	76.81(2.27)	76.03(1.24)	76.12(1.40)
MLP	Fisher Criterion	16.81(3.21)	39.14(5.82)	39.40(4.38)	44.91(5.08)	45.43(5.46)	45.17(4.25)	47.59(7.51)
kNN	Fisher Criterion	41.90(2.86)	69.22(2.64)	69.74(1.63)	72.59(1.57)	73.71(1.28)	72.59(1.10)	71.81(1.28)
LVQ1	Fisher Criterion	42.67(2.73)	71.81(2.87)	72.76(2.48)	73.10(1.80)	73.88(1.28)	72.33(1.98)	72.24(1.47)

Given are the mean and standard deviation (in parentheses) of percent of correct classification of 58 tumor samples with CV10 ($\rho = 0.4$, $\mu = 0.3$, learning rate $= 0.8$, $\alpha = 2.5$).

genes that contribute to the discrimination of different cancer types in spite of different initial conditions.

Another observation from Table 1 is that, for ssEAM, the classification rate usually decreases when more genes are chosen. Likewise, the performance is also deteriorating when too few genes (less than 20) are used in the subset. For PNN, kNN, and LVQ1, their best performance is achieved when more genes (385) are selected compared to ssEAM (79) and classification accuracy decreases as either fewer or more genes are used. These results reflect the importance of gene selection in the context of tumor classification, to some extent. Many genes are not related to the discrimination of certain cancer type of interest and including them in the data set only introduces additional noise into the classification system. On the other hand, important information will be wrongly discarded if inadequate genes are selected.

Fig. 6 shows the effect of the error tolerance parameter ε with respect to the classification accuracy. The value of ε changes from 0.0 (corresponding to EAM) to 0.12, with a step 0.02, and then from 0.15 to 0.5, with a step 0.05, other parameters were set as before. We plot the changes for three different situations, with 79, 135, and 695 genes selected by PSO, respectively. This strategy provides an effective method to increase generalization and decrease overfitting, which is frequently encountered in cancer classification. Together with the results summarized in Table 1, we see that the performance of the classifier can usually be improved with the selection of an appropriate value of ε (0.1 for this data set). The performance drops sharply when ε is larger than 0.15, which is caused by overrelaxing (more than necessary for this problem) the misclassification tolerance criterion during the category formation process in ssEAM training. If the data set had higher overlap, most likely a larger value of ε would achieve better generalization.

3.2 Acute Leukemia Data

This data set is a benchmark data set for cancer classification with gene expression profiles. It is comprised of 72 samples (including bone marrow samples, peripheral blood samples, and childhood AML cases) that belongs to three different leukemia types, i.e., 25 acute myeloid leukemia (AML), 38 B-cell acute lymphoblastic leukemia (ALL), and nine T-cell ALL [25]. These samples are divided into two groups in the original research, 38 for training and 34 for testing. In our experiments, we combine the original training and test sets and perform double cross validation as before. Gene expressions for 7,129 genes (including 312 control genes) were measured using oligonucleotide microarrays. We ranked genes based on their variance across all the samples and chose the top 1,000 for further analysis. The final matrix is in the form of $E = \{e_{i,j}\}_{72\times1,000}$. Nguyen and Rocke used the multivariate partial least squares to address this data set and achieved prediction accuracy higher than 95 percent [40]. Similar results were reported in [35], [36], where several machine learning techniques were used.

As before, we compare the performance of our method with PNN, MLP, and kNN, based on the average results for 20 runs with different splitting. The parameters W_I, c_1, and c_2 for PSO are still set as 0.8, 2, and 2, respectively, and the parameters for ssEAM are $\mu = 0.9$, $\rho = 0.45$, $\alpha = 4$, and learning rate equal to 0.8. δ is set as 0.5, 0.45, 0.4, 0.3, 0.2, 0.1, and 0.0. The smoothing parameter of the Gaussian kernel is set to 1, as before. The MLP includes 15 nodes in the hidden layer with the sigmoid function as the transfer function. The number of prototypes in LVQ1 varies from 3 to 12. For this data set, we can usually achieve good results when the evolution goes for only 100 generations. Each swarm still consists of 50 particles. The results are given in Table 3 and Fig. 5b. The best classification performance is achieved by ssEAM when 63 or 97 genes are selected with PSO, only one sample is misclassified (T-cell ALL67 is misclassified as B-cell ALL). Still, classification performance deteriorates as too many or too few genes are chosen, particularly for

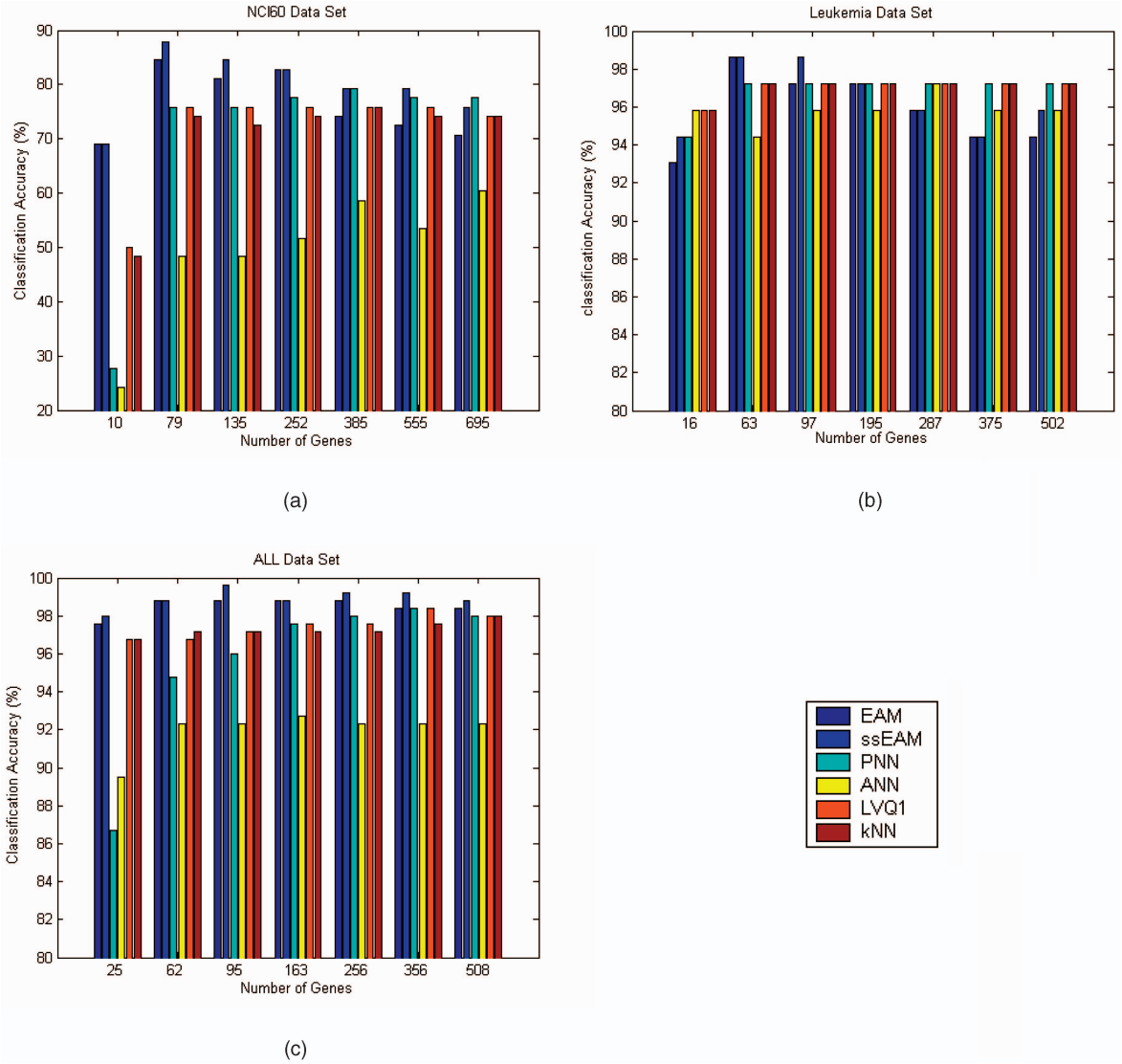


Fig. 5. The best classification accuracy of the 20 runs for the (a) NCI60, (b) leukemia, and (c) ALL data sets. The order for the bars is EAM, ssEAM, PNN, ANN, LVQ1, kNN, from left to right.

TABLE 2
Confusion Matrix of ssEAM for Eight Tumor Types in the NCI60 Data Set

ssEAM Predicted Class \ Actual Class	BR	CNS	CO	LC	LE	ME	OV	RE	S
BR	7	-	-	1	-	-	-	-	8
CNS	-	6	-	-	-	-	-	-	6
CO	-	-	7	-	-	-	-	-	7
LC	-	-	-	7	-	1	2	1	11
LE	-	-	-	-	6	-	-	-	6
ME	-	-	-	-	-	7	-	-	7
OV	1	-	-	1	-	-	4	-	6
RE	-	-	-	-	-	-	-	7	7
S	8	6	7	9	6	8	6	8	58
Accuracy	87.5%	100%	100%	77.8%	100%	87.5%	66.7%	87.5%	

Overall accuracy with LOOCV is 87.9 percent (79 genes, $\rho = 0.4$, $\mu = 0.3$, learning rate $= 0.8$, $\alpha = 2.5$, $\varepsilon = 0.1$).

ssEAM. The number of genes in the data does not much affect PNN, LVQ1, and kNN, although there is some slight decrease. In contrast with the performance of the NCI60 data set, kNN and LVQ1 work well for this data set. The p-values for between ssEAM and PNN, ANN, LVQ1, kNN are 0.0015, 9.9×10^{-9}, 1.6×10^{-4}, and 3.1×10^{-7}, which

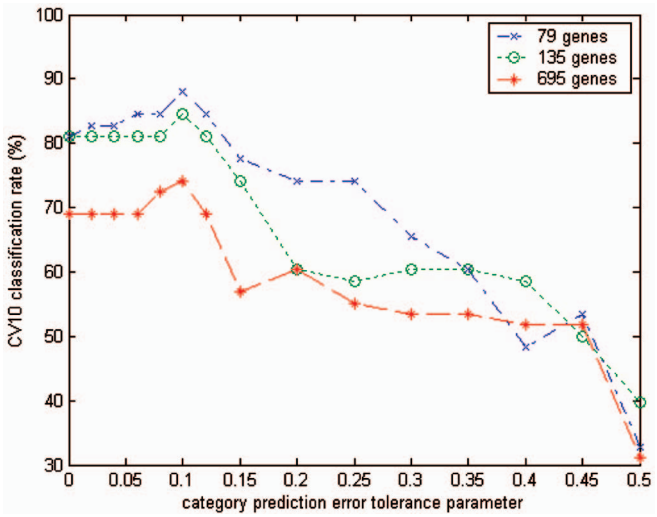


Fig. 6. The effect of the category prediction error tolerance parameter on CV10 classification rate (percent correct classification). The generalization of the classifier is increased with the selection of an appropriate ε. Here, the best classification rate is achieved at $\varepsilon = 0.1$. The performance drops for larger ε as a result of overrelaxing the misclassification tolerance criterion during the category formation process in ssEAM training.

again shows significantly better performance for ssEAM (at a 5 percent significance level).

Among all examples, sample AML66 and T-cell ALL67 were easily misclassified (for example, when $\delta = 0.45$, 23 out of 50 particles misclassified AML66 as ALL and 32 out of 50 particles misclassified T-cell ALL67 either as B-cell ALL or AML). This is similar to the results from other analyses [25], [40]. For the acute leukemia data set, the effect of the introduction of the category prediction error tolerance parameter e $(e > 0)$ is not as pronounced as in the NCI60 data set. The reason may lie in the fact that the amount of overlap among the data is not as high as with the NCI60 data set. The improvement of performance is more effective for semisupervised training when applied to a higher overlap data set.

It is interesting to check whether the genes selected by PSO are really meaningful in biological sense. Among the top 50 genes selected, many of them have already been identified as the important markers for the differentiation of AML and ALL. Specifically, genes like NME4, MPO, CD19, CTSD, LTC4S, zyxin, and PRG1 are known to be useful in AML/ALL diagnosis [25]. Also, some new genes are selected that previously were not reported to be relevant to the classification and more investigation is required. Moreover, we find that the Fisher discriminant criterion can also pick up genes that contribute to the identification of these three leukemia types, like gene zyxin, HoxA9, and MB-1. The reason may lie in the fact that the data set is much better separated than NCI60 and genes express themselves quite differently under different tumor types. Furthermore, we observe that different feature selection methods usually lead to different subsets of selected informative genes with only very small overlap, although the classification accuracy does not change much. Genes that have no biological relevance can still be selected as an artifact of the feature selection algorithms. This suggests that feature selection may provide effective insight in cancer identification; however, careful evaluation is critically necessary due to the problems caused by insufficient data.

3.3 All Data

The ALL data set consists of six different acute lymphoblastic leukemia subtypes, specifically, 15 BCR-ABL, 27 E2A-PBX1, 64 Hyperdiploid > 50, 20 MLL, 43 T-ALL, and 79 TEL-AML1 (248 samples in total) [59]. These samples are divided into training (163 cases) and test (85 cases) groups in the original research. Expression levels for 12,588 genes were measured using oligonucleotide microarrays. We followed the same way as for the acute leukemia data set and selected the 1,000 for further analysis. The data matrix is represented as $E = \{e_{i,j}\}_{248\times1,000}$. The overall classification accuracy for the test group with

TABLE 3
Classification Accuracy for the Acute Leukemia Data Set

Acute Leukemia Data		Features (Genes)						
		16	63	97	195	287	375	502
EAM (ε =0)	PSO	89.72(2.08)	94.44(2.67)	95.07(1.65)	94.31(1.68)	93.26(1.58)	93.13(1.23)	92.36(1.39)
ssEAM (ε =0.1)	PSO	91.60(1.23)	97.15(0.95)	97.50(0.73)	95.83(0.90)	94.65(0.82)	93.68(0.71)	92.64(1.43)
PNN	Fisher Criterion	90.00.(2.61)	96.32(0.68)	96.74(0.68)	96.46(0.71)	96.39(0.70)	96.25(0.91)	96.18(0.99)
MLP	Fisher Criterion	93.61(2.27)	92.50(1.23)	93.47(2.07)	91.60(4.19)	91.74(3.94)	91.60(2.90)	91.67(3.60)
LVQ1	Fisher Criterion	94.86.(0.65)	95.83(0.45)	96.45(0.84)	95.97(0.77)	96.04(0.93)	96.10(0.86)	95.90(0.71)
kNN	Fisher Criterion	95.07(0.71)	95.83(0.45)	96.18(0.62)	96.11(0.73)	95.90(0.84)	95.69(1.00)	95.69(0.77)

Given are the mean and standard deviation (in parentheses) of percent of correct classification for 72 tumor samples with CV10 ($\rho = 0.454$, $\mu = 0.9$, learning rate $= 0.8$, $\alpha = 4$).

TABLE 4
Classification Accuracy for the ALL Data Set

ALL Data		Features (Genes)						
		25	62	95	163	256	356	508
EAM (ε =0)	PSO	96.65(0.57)	97.72(0.42)	97.94(0.45)	97.70(0.39)	97.88(0.43)	97.80(0.36)	97.48(0.54)
ssEAM (ε =0.1)	PSO	96.35(0.63)	97.94(0.49)	98.29(0.45)	98.04(0.40)	98.23(0.46)	98.21(0.42)	97.68(0.50)
PNN	Fisher Criterion	86.05.(0.38)	94.31(0.34)	95.50(0.48)	97.04(0.40)	97.44(0.42)	97.76(0.33)	97.32(0.38)
MLP	Fisher Criterion	87.52(1.25)	90.12(1.64)	89.94(1.35)	89.68(1.62)	89.70(1.67)	88.97(1.66)	86.79(2.66)
LVQ1	Fisher Criterion	96.17.(0.40)	96.39(0.31)	96.65(0.35)	97.22(0.34)	97.22(0.34)	97.42(0.48)	97.52(0.38)
kNN	Fisher Criterion	96.39(0.36)	96.77(0.32)	96.83(0.35)	96.73(0.26)	96.69(0.31)	97.28(0.29)	97.44(0.30)

Given are the mean and standard deviation (in parentheses) of percent of correct classification for 248 tumor samples with CV10 ($\rho = 0.45$, $\mu = 0.9$, learning rate $= 0.8$, $\alpha = 4$).

support vector machines is about 96 percent, as reported by Yeoh et al. [59]. Li et al. performed a comparative analysis with four machine learning algorithms and eight different feature selection strategies based on the original division [36]. The best performance results for all these methods are all higher than 90 percent. In our experiments, we still performed the double cross validation as described before with 20 different runs.

We set the parameters of the PSO/ssEAM system at exactly the same values as those used for the leukemia data set analysis, i.e., $W_I = 0.8$, $c_1 = c_2 = 2$, $\mu = 0.9$, $\rho = 0.45$, $\alpha = 4$, and learning rate equal to 0.8. δ varies from 0.5 to 0.0, generating different gene subsets. The parameters for PNN, MLP, LVQ1, and kNN are all set as before. The mean and best performance of the 20 runs is shown in Table 4 and Fig. 5c, respectively. Again, the best overall classification performance is achieved by ssEAM when 95 genes are selected with PSO. Particularly, the best classification achieved is 99.6 percent, indicating only one misclassification, where the #8 BCR-ABL sample is misclassified into the $\mathrm{Hyperdiploid} > 50$ category. PNN, LVQ1, and kNN also achieve good performance for the data set, while the classification accuracy for ANN is lower than all other methods. Comparison of ssEAM with PNN, ANN, LVQ1, and kNN stills shows the statistically better performance for ssEAM at a significance level of 0.05 (the corresponding p-values are 1.6×10^{-4}, 0, 9.6×10^{-7}, 2.5×10^{-8}, respectively). Also, the number of genes in the data still has effects on the classification results, but only causes some slight changes, which are not as that important as in the case of the NCI60 data set. For example, the average accuracy for ssEAM with 508 genes used is 97.68 percent, which is only 0.61 percent lower than the best performance.

4 Conclusions

Classification is critically important for cancer diagnosis and treatment. Microarray technologies provide a new and effective avenue for discriminating different kinds of cancer types, while simultaneously bringing many new challenges. Here, we utilized Semisupervised Ellipsoid ARTMAP combined with particle swarm optimization to distinguish tumor tissues with more than two categories through analyzing gene expression profiling. The proposed combination of methods achieves qualitatively good results on three publicly accessible benchmark data sets, particularly with the NCI60 data set, which is not effectively dealt with by previous methods. The comparison with four other important machine learning techniques shows that ssEAM/PSO can outperform them on all three data sets and the difference in classification accuracy is found to be statistically significant.

With all the improvement we obtain, we also note that there are still many problems that remain to be solved in cancer identification with gene expression profiles, especially how to effectively cope with the noise introduced in the different stages of the microarray experiment and the problem of the curse of dimensionality. The latter problem is particularly serious due to the rapidly and persistently increasing capability of gene chip technologies that also follow Moore's law [39], in contrast to the existing limitations in conditions like sample collections. This makes the published data sets consist of only a small set of samples for each tumor type (typically, less than 30), however, along with tens of thousands of gene expression measurements. Efforts have been made in order to identify the potential genes relevant to the cancer discrimination, but questions, such as how many genes are really needed and are these feature (gene) subsets selected really meaningful in the biological sense, are not answered satisfactorily. From the results observed in our experiments, we can see the important effects of feature (gene) selection, based on the final classification rates. Nevertheless, it is very difficult to propose some useful rules or criteria to determine the optimal number of genes for disease diagnosis, especially when the data sets studied consist of a wide range of cancer categories, such as in the case of NCI60. Without doubt, more samples would be greatly helpful in effectively

evaluating different kinds of classifiers and constructing a cancer discrimination system. In the meantime, more advanced feature selection approaches are required in order to find informative genes that are more efficient in prediction and prognosis.

ACKNOWLEDGMENTS

Partial support for this research from the US National Science Foundation and from the M.K. Finley Missouri endowment is gratefully acknowledged. The authors would also like to thank the associate editor and the anonymous reviewers for their valuable comments.

REFERENCES

[1] A. Alizadeh, M. Eisen, R. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, T. Moore, J. Hudson, Jr, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Bostein, P. Brown, and L. Staudt, "Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling," *Nature,* vol. 403, pp. 503-511, 2000.

[2] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. Levine, "Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays," *Proc. Nat'l Academy of Sciences USA '96,* pp. 6745-6750, 1999.

[3] G.C. Anagnostopoulos and M. Georgiopoulos, "Hypersphere ART and ARTMAP for Unsupervised and Supervised Incremental Learning," *Proc. IEEE-INNS-ENNS Int'l Joint Conf. Neural Networks (IJCNN '00),* vol. 6, pp. 59-64, 2000.

[4] G.C. Anagnostopoulos, "Novel Approaches in Adaptive Resonance Theory for Machine Learning," doctoral thesis, Univ. of Central Florida, Orlando, 2001.

[5] G.C. Anagnostopoulos and M. Georgiopoulos, "Ellipsoid ART and ARTMAP for Incremental Unsupervised and Supervised Learning," *Proc. IEEE-INNS-ENNS Int'l Joint Conf. Neural Networks (IJCNN '01),* vol. 2, pp. 1221-1226, 2001.

[6] G.C. Anagnostopoulos, M. Georgiopoulos, S. Verzi, and G. Heileman, "Reducing Generalization Error and Category Proliferation in Ellipsoid ARTMAP via Tunable Misclassification Error Tolerance: Boosted Ellipsoid ARTMAP," *Proc. IEEE-INNS-ENNS Int'l Joint Conf. Neural Networks (IJCNN '02),* vol. 3, pp. 2650-2655, 2002.

[7] C. Ambroise and G. McLachlan, "Selection Bias in Gene Extraction on the Basis of Microarray Gene-Expression Data," *Proc. Nat'l Academy of Sciences USA '99,* pp. 6562-6566, 2002.

[8] A. Bagirov, B. Ferguson, S. Ivkovic, G. Saunders, and J. Yearwood, "New Algorithms for Multi-Class Cancer Diagnosis Using Tumor Gene Expression Signatures," *Bioinformatics,* vol. 19, no. 14, pp. 1800-1807, 2003.

[9] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue Classification with Gene Expression Profiles," *Proc. Fourth Ann. Int'l Conf. Computational Molecular Biology,* pp. 583-598, 2000.

[10] D. Berrar, C. Downes, and W. Dubitzky, "Multiclass Cancer Classification Using Gene Expression Profiling and Probabilistic Neural Networks," *Proc. Eighth Pacific Symp. Biocomputing,* pp. 5-16, 2003.

[11] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent, "Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling," *Nature,* vol. 406, pp. 536-540, 2000.

[12] U. Braga-Neto and E. Dougherty, "Is Cross-Validation Valid for Small-Sample Microarray Classification," *Bioinformatics,* vol. 20, no. 3, pp. 374-380, 2004.

[13] G. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing,* vol. 37, pp. 54-115, 1987.

[14] G. Carpenter, S. Grossberg, N. Markuzon, J. Reynolds, and D. Rosen, "Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps," *IEEE Trans. Neural Networks,* vol. 3, pp. 698-713, 1992.

[15] G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System," *Neural Networks,* vol. 4, pp. 759-771, 1991.

[16] T. Caudell, S. Smith, G. Johnson, and D. Wunsch, "An Application of Neural Networks to Group Technology," *Proc. SPIE Applications of Neural Networks II,* 1991.

[17] S. Cho and H. Won, "Machine Learning in DNA Microarray Analysis for Cancer Classification," *Proc. First Asia-Pacific Bioinformatics Conf. (APBC '03),* Y.-P.P. Chen, ed., pp. 189-198, 2003.

[18] L. Conde, A. Mateos, J. Herrero, and J. Dopazo, "Unsupervised Reduction of the Dimensionality Followed by Supervised Learning with a Perceptron Improves the Classification of Conditions in DNA Microarray Gene Expression Data," *Proc. 12th IEEE Workshop Neural Networks for Signal Processing,* pp. 77-86, 2002.

[19] L. Deng, J. Pei, J. Ma, and D. Lee, "A Rank Sum Test Method for Informative Gene Discovery," *Proc. 2004 ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 410-419, 2004.

[20] M. Dettling and P. Bühlmann, "Boosting for Tumor Classification with Gene Expression Data," *Bioinformatics,* vol. 19, pp. 1061-1069, 2003.

[21] S. Doctor, G. Venayagamoorthy, and V. Gudise, "Optimal PSO for Collective Robotic Search Applications," *Proc. Congress Evolutionary Computation '04,* vol. 2, pp. 1390-1395, 2004.

[22] R. Duda, P. Hart, and D. Stork, *Pattern Xlassification,* second ed. Wiley & Sons, 2001.

[23] M. Eisen and P. Brown, "DNA Arrays for Analysis of Gene Expression," *Methods Enzymology,* vol. 303, pp. 179-205, 1999.

[24] M. Garber, O. Troyanskaya, K. Schluens, S. Petersen, Z. Thaesler, M. Pacyna-Gengelbach, M. Rijn, G. Rosen, C. Perou, R. Whyte, R. Altman, P. Brown, D. Botstein, and I. Petersen, "Diversity of Gene Expression in Adenocarcinoma of the Lung," *Proc. Nat'l Academy of Sciences USA '98,* pp. 13784-13789, 2001.

[25] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science,* vol. 286, pp. 531-537, 1999.

[26] S. Grossberg, "Adaptive Pattern Recognition and Universal Encoding II: Feedback, Expectation, Olfaction, and Illusions," *Biological Cybernetics,* vol. 23, pp. 187-202, 1976.

[27] M. Healy and T. Caudell, "Acquiring Rule Sets as a Product of Learning in the Logical Neural Architecture LAPART," *IEEE Trans. Neural Networks,* vol. 8, pp. 461-474, 1997.

[28] M. Healy, T. Caudell, and S. Smith, "A Neural Architecture for Pattern Sequence Verification through Inferencing," *IEEE Trans. Neural Networks,* vol. 4, pp. 9-20, 1993.

[29] J. Jaeger, R. Sengupta, and W. Ruzzo, "Improved Gene Selection for Classification of Microarrays," *Proc. Pacific Symp. Biocomputing,* vol. 8, pp. 53-64, 2003.

[30] J. Kennedy and R. Eberhart, "A Discrete Binary Version of the Particle Swarm Optimization," *Proc. Conf. Systems, Man, and Cybernetics,* pp. 4104-4108, 1997.

[31] J. Kennedy, R. Eberhart, Y. Shi, *Swarm Intelligence.* Morgan Kaufmann, 2001.

[32] J. Khan, J. Wei, M. Ringnér, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. Antonescu, C. Peterson, and P. Meltzer, "Classification and Diagnostic Prediction of Cancers Using Gene Expression Profiling and Artificial Neural Networks," *Nature Medicine,* vol. 7, pp. 673-679, 2001.

[33] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Proc. Int'l Joint Conf. Artificial Intelligence,* 1995.

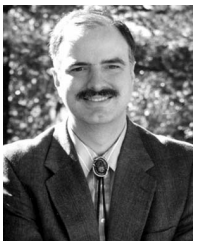
[34] T. Kohonen, *Self-Organizing Maps,* third ed. Springer, 2001.

[35] Y. Lee and C. Lee, "Classification of Multiple Cancer Types by Multicategory Support Vector Machines Using Gene Expression Data," *Bioinformatics,* vol. 19, pp. 1132-1139, 2003.

[36] T. Li, C. Zhang, and M. Ogihara, "A Comparative Study of Feature Selection and Multiclass Classification Methods for Tissue Classification Based on Gene Expression," *Bioinformatics,* vol. 20, no. 15, pp. 2429-2437, 2004.
[37] R. Lipshutz, S. Fodor, T. Gingeras, and D. Lockhart, "High Density Synthetic Oligonucleotide Arrays," *Nature Genetics,* vol. 21, pp. 20-24, 1999.
[38] A. Mateos, J. Herrero, J. Tamames, and J. Dopazo, "Supervised Neural Networks for Clustering Conditions in DNA Array Data after Reducing Noise by Clustering Gene Expression Profiles," *Proc. Microarray Data Analysis II,* pp. 91-103, 2001.
[39] S. Moore, "Making Chips to Probe Genes," *IEEE Spectrum,* vol. 38, pp. 54-60, 2001.
[40] D. Nguyen and D. Rocke, "Multi-Class Cancer Classification via Partial Least Squares with Gene Expression Profiles," *Bioinformatics,* vol. 18, pp. 1216-1226, 2002.
[41] C. Ooi and P. Tan, "Genetic Algorithms Applied to Multi-Class Prediction for the Analysis of Gene Expression Data," *Bioinformatics,* vol. 19, pp. 37-44, 2003.
[42] P. Park, M. Pagano, and M. Boneti, "A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data," *Proc. Pacific Symp. Biocomputing,* vol. 6, pp. 52-63, 2001.
[43] C. Perou, T. Sørlie, M. Eisen, M. Rijn, S. Jeffrey, C. Rees, J. Pollack, D. Ross, J. Johnsent, L. Akslen, Ø. Fluge, A. Pergamenschlkov, C. Williams, S. Zhu, P. Lønning, A. Børresen-Dale, P. Brown, and D. Botstein, "Molecular Portraits of Human Breast Tumors," *Nature,* vol. 406, pp. 747-752, 2000.
[44] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. Mesirov, T. Poggio, W. Gerald, M. Loda, E. Lander, and T. Golub, "Multiclass Cancer Diagnosis Using Tumor Gene Expression Signatures," *Proc. Nat'l Academy of Sciences USA '98,* pp. 15149-15154, 2001.
[45] U. Scherf, D. Ross, M. Waltham, L. Smith, J. Lee, L. Tanabe, K. Kohn, W. Reinhold, T. Myers, D. Andrews, D. Scudiero, M. Eisen, E. Sausville, Y. Pommier, D. Botstein, P. Brown, and J. Weinstein, "A Gene Expression Database for the Molecular Pharmacology of Cancer," *Nature Genetics,* vol. 24, pp. 236-44, 2000.
[46] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization," *Proc. Seventh Ann. Conf. Evolutionary Programming,* pp. 591-601, 1998.
[47] M. Schummer, W. Ng, R. Bumgarner, P. Nelson, B. Schummer, D. Bednarski, L. Hassell, R. Baldwin, B. Karlan, and L. Hood, "Comparative Hybridization of an Array of 21500 Ovarian cDNA for the Discovery of Genes Overexpressed in Ovarian Carcinomas," *Gene,* vol. 238, pp. 375-385, 1999.
[48] T. Serrano-Gotarredona, B. Linares-Barranco, and A. Andreou, *Adaptive Resonance Theory Microchip.* Kluwer Academic, 1998.
[49] D. Specht, "Probabilistic Neural Networks," *Neural Networks,* vol. 3, pp. 109-118, 1990.
[50] C. Tang, A. Zhang, and J. Pei, "Mining Phenotypes and Informative Genes from Gene Expression Data," *Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining,* pp. 655-660, 2003.
[51] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, "Diagnosis of Multiple Cancer Types by Shrunken Centroids of Gene Expression," *Proc. Nat'l Academy of Sciences USA '99,* pp. 6567-6572, 2002.
[52] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman, "Missing Value Estimation Methods for DNA Microarrays," *Bioinformatics,* vol. 17, pp. 520-525, 2001.
[53] D. Wigle, I. Jurisica, N. Radulovich, M. Pintilie, J. Rossant, N. Liu, C. Lu, J. Woodgett, I. Seiden, M. Johnston, S. Keshavjee, G. Darling, T. Winton, B. Breitkreutz, P. Jorgenson, M. Tyers, F. Shepherd, and M. Tsao, "Molecular Profiling of Non-Small Cell Lung Cancer and Correlation with Disease-Free Survival," *Cancer Research,* vol. 62, pp. 3005-3008, 2002.
[54] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. Marks, and J. Nevins, "Prediction the Clinical Status of Human Breast Cancer by Using Gene Expression Profile," *Proc. Nat'l Academy of Sciences USA '98,* pp. 11462-11467, 2001.
[55] D. Wunsch II, T. Caudell, D. Capps, R. Marks, and A. Falk, "An Optoelectronic Implementation of the Adaptive Resonance Neural Network," *IEEE Trans. Neural Networks,* vol. 4, pp. 673-684, 1993.
[56] D. Wunsch II, "An Optoelectronic Learning Machine: Invention, Experimentation, Analysis of First Hardware Implementation of the ART1 Neural Network," PhD dissertation, Univ. of Washington, 1991.
[57] R. Xu and D. Wunsch II, "Probabilistic Neural Networks for Multi-Class Tissue Discrimination with Gene Expression Data," *Proc. Int'l Joint Conf. Neural Networks (IJCNN '03),* pp. 1696-1701, 2003.
[58] R. Xu and D. Wunsch II, "Survey of Clustering Algorithms," *IEEE Trans. Neural Networks,* vol. 16, no. 3, pp. 645-678, 2005.
[59] E. Yeoh, M. Ross, S. Shurtleff, W. William, D. Patel, R. Mahfouz, F. Behm, S. Raimondi, M. Reilling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C. Pui, W. Evans, C. Naeve, L. Wong, and J. Downing, "Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Acute Lymphoblastic Leukemia by Gene Expression Profiling," *Cancer Cell,* vol. 1, pp. 133-143, 2002.
[60] K. Yeung and W. Ruzzo, "Principal Component Analysis for Clustering Gene Expression Data," *Bioinformatics,* vol. 17, pp. 763-774, 2001.

Rui Xu received the BE degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, Hubei, China, in 1997, the ME degree in electrical engineering from Sichuan University, Chengdu, Sichuan, in 2000, and the PhD degree in electrical engineering from the University of Missouri-Rolla in 2006 His research interests include machine learning, neural networks, pattern classification and clustering, and bioinformatics. He is a member of the IEEE.

Georgios C. Anagnostopoulos received the MSc and PhD degrees from the School of Electrical Engineering and Computer Science at the University of Central Florida, Orlando, in 1997 and 2001, respectively. Currently, he is an assistant professor in the Department of Electrical and Computer Engineering at the Florida Institute of Technology in Melbourne, Florida. His main research area is machine learning with emphasis on pattern recognition and detection as they apply to data modeling and mining, machine vision, remote sensing, and bioinformatics, among other fields.



Donald C. Wunsch II received the BS degree in applied mathematics from the University of New Mexico, Albuquerque, and the MS degree in applied mathematics and the PhD degree in electrical engineering from the University of Washington, Seattle. He is the Mary K. Finley Missouri Distinguished Professor of Computer Engineering at the University of Missouri-Rolla, where he has been since 1999. His prior positions were associate professor and director of the Applied Computational Intelligence Laboratory at Texas Tech University, senior principal scientist at Boeing, consultant for Rockwell International, and technician for International Laser Systems. He has well over 200 publications and has attracted more than $5 million in research funding. He has supervised 11 PhD recipients—six in electrical engineering, four in computer engineering, and one in computer science. Dr. Wunsch received the Halliburton Award for Excellence in Teaching and Research and the US National Science Foundation CAREER Award. He served as a voting member of the IEEE Neural Networks Council, technical program cochair for IJCNN '02, general chair for IJCNN '03, an International Neural Networks Society Board of Governors member, and 2005 president of the International Neural Networks Society.