Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 71 (2008) 1345–1358
NEUROCOMPUTING
www.elsevier.com/locate/neucom

A sequential multi-category classifier using radial basis function networks

S. Suresh*, N. Sundararajan, P. Saratchandran

School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore

Received 31 August 2006; received in revised form 30 May 2007; accepted 10 June 2007
Communicated by K. Li
Available online 6 July 2007

Abstract

This paper presents a new sequential multi-category classifier using radial basis function (SMC-RBF) network for real-world classification problems. The classification algorithm processes the training data one by one and builds the RBF network starting with zero hidden neuron. The growth criterion uses the misclassification error, the approximation error to the true decision boundary and a distance measure between the current sample and the nearest neuron belonging to the same class. SMC-RBF uses the hinge loss function (instead of the mean square loss function) for a more accurate estimate of the posterior probability. For network parameter updates, a decoupled extended Kalman filter is used to reduce the computational overhead. Performance of the proposed algorithm is evaluated using three benchmark problems, viz., image segmentation, vehicle and glass from the UCI machine learning repository. In addition, performance comparison has also been done on two real-world problems in the areas of remote sensing and bio-informatics. The performance of the proposed SMC-RBF classifier is also compared with the other RBF sequential learning algorithms like MRAN, GAP-RBFN, OS-ELM and the well-known batch classification algorithm SVM. The results indicate that SMC-RBF produces a higher classification accuracy with a more compact network. Also, the study indicates that using a function approximation algorithm for classification problems may not work well when the classes are not well separated and the training data is not uniformly distributed among the classes.

Keywords: RBF network; Hinge loss function; Sequential learning; Multi-category classification; Decoupled extended Kalman filter

1. Introduction

In many real-world classification problems, such as medical diagnosis applications, the complete set of training data are not available at the same time; rather they become available only over a period of time. For example, in a cancer detection problem involving micro-array gene expression data, the tumor marker records are available initially only from a small number of patients. However, the size of the database increases with more tumor patients over a period of time. Also, tumor marker labels may change after clinical surgeries requiring modifications in the existing database. Both the addition of new samples and changes of class labels in the existing database normally requires retraining the designed classifiers. This retraining process requires a large amount of computational time and memory. A learning scheme that handles these sequential data without retraining on the entire data will be quite useful for various applications. In this paper, we present such a scheme for multi-category classification problems.

Among various classification methods, neural network based classifiers have been successfully used in a number of applications such as pattern recognition [6], remote sensing [11] and in biomedical areas [8]. Neural networks are able to construct complex classification boundaries [12] without any prior assumption on the input data statistics. Radial basis function networks (RBFN) have been widely used in all these applications due to its approximation ability as well as its simple architecture.

RBFN is a three layered feed forward network with a radially symmetric Gaussian function as an activation

*Corresponding author.
E-mail address: ssundaram@ntu.edu.sg (S. Suresh).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.06.003

function in the hidden layer [19]. Let $\mathbf{X}$ be an n-dimensional input pattern vector ($\mathbf{X} = [x_1, x_2, \ldots, x_n]$). The RBFN with m outputs and K Gaussian neurons is given by

$$\hat{y}_i = \sum_{j=1}^{K} w_{ij} \exp\left(-\frac{\|\mathbf{X} - \boldsymbol{\mu}_j\|}{2\sigma_j^2}\right),$$
$$\boldsymbol{\mu}_j \in \Re^n, \ \ \sigma_j^2 \in \Re^+, \ \ w_{ij} \in \Re, \ \ i = 1, 2, \ldots, m, \qquad (1)$$

where y_i is the ith output, $\boldsymbol{\mu}_j$ and σ_j are the center and width of the Gaussian basis function of the jth neuron, respectively, and w_{ij} is the interconnection weight between the ith output neuron and the jth Gaussian neuron. $\Re^+$ indicates the set of all positive real numbers.

In classical batch learning approach for RBFN (i.e., repeated presentation of training samples), the number of hidden neurons are fixed a priori. Usually, the centers and widths of the Gaussian neurons are calculated using the K-means clustering method and the weights are estimated using the least square method [16,17]. The generalization ability [18] and compactness [1,3,13] of RBFNs have been addressed in batch scenarios by many researchers. In the sequential learning scenario, the RBFN parameters are updated sequentially based on the error computed using the new training sample [7,10,15,26]. Since, the sequential learning algorithm does not require retraining of network whenever new data arrives, it is preferred over classical batch algorithms for many practical applications. In a sequential learning algorithm, the training samples are presented one by one such that the algorithm does not require prior information about the total number of training samples. Hence, the sequential learning scheme requires less computational effort and minimal storage space over the batch learning scheme [10,15,19,25].

Recently Huang et al. [7] have developed a sequential algorithm called growing and pruning RBFN (GAP-RBFN) and have presented its better performance on a number of function approximation and classification problems. The GAP-RBFN algorithm introduced the concept of neuron significance for growing and pruning of the hidden neurons and does not require fixing the number of neurons a priori.

In GAP-RBFN algorithm [7], the criterion for growing, updating the parameters and pruning are given by

- *Growing criterion*: If $\|\mathbf{X}_n - \boldsymbol{\mu}_{nr}\| > \varepsilon_n$ and $E_{\text{sig}}(nr) > e_{\min}$ then add a new neuron to the network. Here, nr is the nearest neighbor neuron to the new training sample (i.e., neuron closest to sample $\mathbf{X}_n$) and E_{sig} is the significance of nrth neuron defined as its average output over all the input samples it has received so far [7]. The control parameters $e_{\min}$ and ε_n are thresholds chosen a priori by trial and error.
- *Parameter update*: If the growth criterion is not satisfied then adjust the network parameters only for the nearest neighbor neuron (nr) using an extended Kalman filter (EKF) algorithm [7].
- *Pruning criterion*: If the significance of the nearest neighbor neuron is less than the expected accuracy ($E_{\text{sig}}(nr) < e_{\min}$) then the nrth hidden neuron can be removed from the network.

In GAP-RBFN, the criteria for growing, pruning and updating parameters are carried out for the nearest (to the current input sample) neuron. This is similar to the other sequential algorithm known as extended minimal resource allocation network (EMRAN) [25].

The performance of MRAN and GAP-RBFN algorithms have been evaluated [7,15,26] using various benchmark function approximation and classification problems. For function approximation problems, the generalization performance of these sequential algorithms has been shown to be better than the batch algorithms. For classification problems, the generalization performance depends heavily on the optimal selection of control parameters of the learning algorithms and the input data distribution as they use the basic function approximation approach for classification. Hence, sequential learning algorithms available in the literature may not work well for classification problems as they use criteria based on only concepts of function approximation and not classification.

To show this clearly, a motivational example is presented below to indicate how the growing/pruning criteria developed in GAP-RBFN, which is good for function approximation problems, is inadequate for classification problems. The motivational example also shows that one has to consider the Euclidian distance measure within the class and the misclassification error in the criteria for better performance.

1.1. Motivational example

Let us consider a simple two class problem as shown in Fig. 1(a). Here, each input pattern has two input features, namely x_1 and x_2. If $x_1 < 0.5$ and $x_2 < 0.5$ then the sample is assigned to class C_1. Similarly, if $x_1 > 0.5$ and $x_2 > 0.5$ then the sample is assigned to class C_2. The class label C_1 and C_2 are coded as 1 and -1, respectively. The training samples are generated randomly. Next, we use the GAP-RBFN learning algorithm to illustrate our point that the criteria appropriate for function approximation are inadequate for classification.

The objective here is to approximate the ideal decision surface shown in Fig. 1(b) with an approximation error ($e_{\min} = 1.0 \times 10^{-3}$). It should be noted here that in this figure and subsequent figures the scale for x_2 reads from right (0) to left (1) to make the visualization of the decision surface clear.

The threshold for distance growing criterion ε_n is fixed at 0.2. The training samples are presented sequentially to the GAP-RBFN classifier. The first six samples presented to the network satisfies the growing criterion. Hence, six hidden neurons are added to the network. The centers of the hidden neurons are nothing but the actual sample

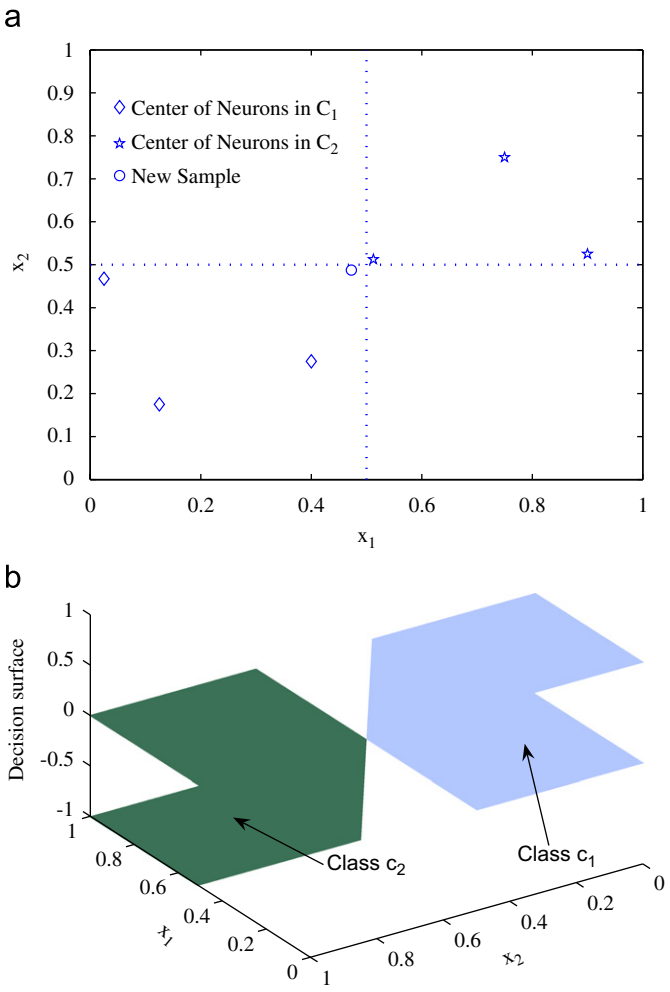

Fig. 1. Actual decision surface for neural approximation and training samples for two-class problem. (a) Case I: training samples in two dimensional feature plane. (b) Case II: ideal decision surface to be approximated by neural classifier.

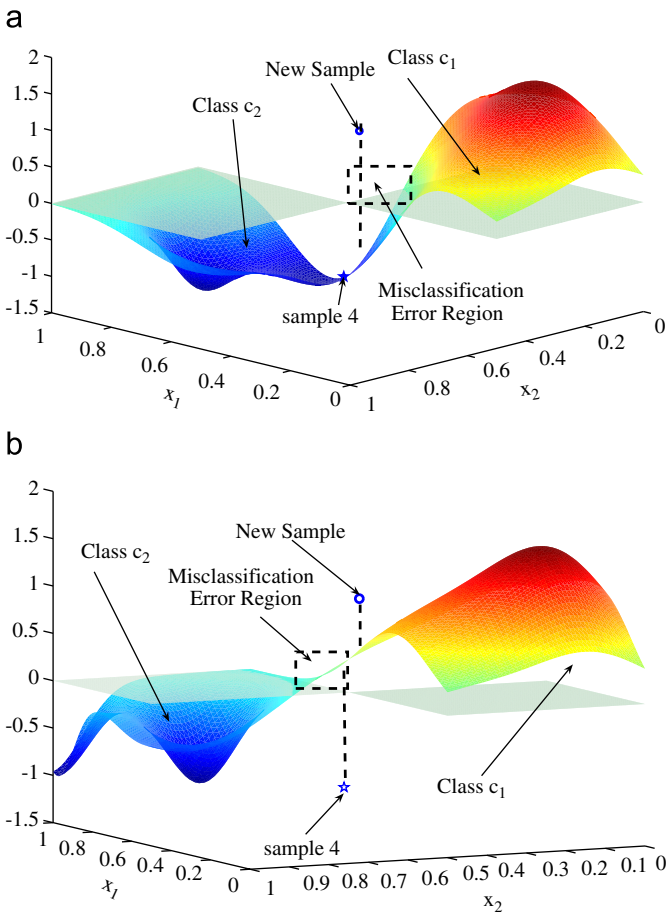

Fig. 2. Decision surface approximation of GAP-RBFN algorithm in classification problems. (a) Case I: GAP-RBFN decision surface before presenting the new sample. (b) Case II: GAP-RBFN decision surface after presenting the new sample.

locations as shown in Fig. 1(a). The decision boundary approximated by the GAP-RBFN network after learning the six samples is shown in Fig. 2. The misclassification region is marked as a rectangular box in Fig. 2(a). Here, the misclassification region falls in the class C_1 decision boundary and hence the output of GAP-RBFN is negative.

Now, let $\mathbf{X} = [0.4725\ 0.4825]$ be the new training sample and it belongs to class C_1 (represented as 'O' in Fig. 1(a)). We can observe from the Fig. 2(a) that the new sample 'O' falls in the misclassification region. Hence, the output of GAP-RBFN classifier for this new sample is -1.0755 and this indicates that the predicted class label for the new sample will be C_2, whereas its true class label is C_1. Hence, the learning error (actual output minus predicted output) is $1-(-1.0756)$. The nearest neuron to the new training sample is $\boldsymbol{\mu}_4 = [0.5125\ 0.5125]$. The distance from the nearest neuron is $\|\mathbf{X}-\boldsymbol{\mu}_4\| = 0.0472$ and the significance (E_{sig}) is 0.025. Since, the growth criterion: ($\|\mathbf{X}-\boldsymbol{\mu}_{nr}\| > 0.2$ and $E_{\text{sig}}(K+1) > 1.0\times 10^{-3}$) (see Ref. [7] for significance calculation) does not satisfy the growth criteria, GAP-RBFN updates the network parameters of the nearest neuron using the EKF. The updated parameters of the nearest neuron are: $\boldsymbol{\mu}_4 = [1.0114\ 0.9774]$, $\sigma_4 = 0.1212$, $w_4 = -0.9910$ (old value: $\boldsymbol{\mu}_4 = [0.5125\ 0.5125]$, $\sigma_4 = 0.2319$, $w_4 = -1.359$).

Now, let the network be tested with a new sample from class C_2 which is the fourth sample in this class given by [0.5125 0.5125]. This happens to be the original center of the nearest neuron. For this, the output of the network is 0.2895 and hence the sample is misclassified as belonging to class C_1. This is caused due to the update for the nearest neuron. The centers of the hidden neurons and the decision boundaries after parameter updates are shown in Fig. 2(b). The misclassification region falls in class C_2 decision boundary and is marked as a rectangular box in Fig. 2(b). The samples in this region are misclassified as class C_1 instead of C_2. From Fig. 2(a) and (b), we can see that the decision boundary for the class C_2 is reduced significantly after the parameter update. At the same time the decision boundary increases for class C_1, i.e., the misclassification error region shifts from class C_1 decision boundary to C_2 decision boundary. Hence, the classifier developed using

GAP-RBFN does not capture the decision boundary accurately and results in poor classification performance. In case of multi-category classification problems with a strong overlap between the classes, this effect will be significant.

The above example clearly shows that the criteria used in GAP-RBFN originally developed for function approximation may not be adequate for accurate classification. In function approximation problems, for the growth criterion one uses the nearest neuron to the current training sample. However, if one does this for classification problems, one may select the neuron which may not belong to the same class as the sample. Hence, it will be appropriate to consider the nearest neuron in the same class to develop the growing and pruning criterion.

In the above example, the nearest neuron in the same class C_1 for the new training sample $\mathbf{X}$ is neuron 3 at [0.4 0.275]. If one picks this neuron as the nearest neuron and updates its parameters, then the decision surface generated by the classifier will be as shown in Fig. 3. The corresponding updated parameters for this neuron 3 are: $\boldsymbol{\mu}_3 = [0.4371\ 0.477]$, $\sigma_3 = 0.1255$ and $w_3 = 0.6869$. With this scheme, it can be seen from Fig. 3 that the new training sample and sample 4 are classified correctly.

Another issue in the GAP-RBFN algorithm is the use of the neuron significance (average contribution of the neuron to the network output) as a growth criterion. The calculation of neuron significance requires uniform distribution of the input data, which is not common in many classification problems due to the imbalance in the training data (for example, in the cancer problem the variation between different tumour samples is not uniform). In case of function approximation problems, the approximation error (target minus predicted output) based on neuron growth criterion is often used. As shown in the motivational example, for classification problem, GAP-RBFN criteria does not approximate the decision surface accurately. One has to consider the misclassification error and the Euclidian distance from the nearest neuron of the same class in the growth criterion. The presence of misclassification error in growing criterion reduces the classification error considerably.

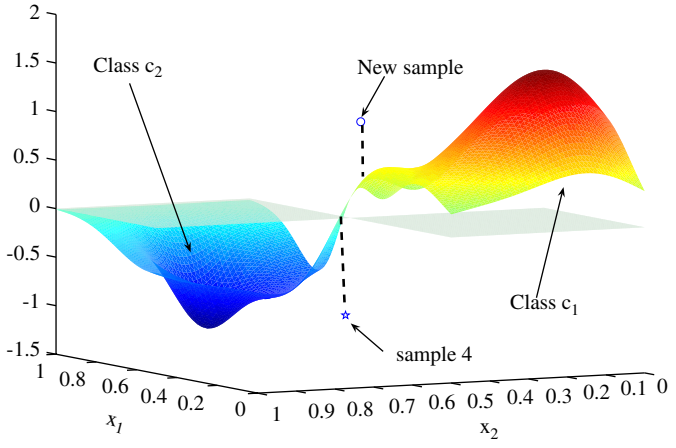

Fig. 3. Decision surface after updating the network parameters of nearest neuron in the same class.

In view of the above points, in this paper, we propose a new classification algorithm that is based exclusively on classification objectives. The proposed algorithm is sequential, can handle multi-category classification problems and uses an RBF network and is referred to as SMC-RBF. The SMC-RBF algorithm incorporates the Euclidian distance measure within the class (defined by the distance of the current input sample to the nearest neuron in the same class) and the misclassification error in the growth criterion. This results in a more compact network with a higher classification accuracy. Also, for classification problem, it has been shown that using a hinge loss function one obtains a better estimate of posterior probability compared to the mean square error loss function [27]. Hence, we use a hinge loss function in this paper to reduce this probability estimation error.

The performance of the proposed SMC-RBF is evaluated by comparing it with other sequential RBF learning algorithms such as MRAN [15,26] and GAP-RBFN [7]. Also, the performance is compared with the other well-known classifier algorithm such as the batch support vector machines (SVM) [5] and recently developed online sequential extreme learning machine (OS-ELM) [14]. Experimental results on benchmark problems from UCI machine learning repository [2] show that the proposed SMC-RBF produces better classification performance. The problems considered in this study are (i) image segmentation, (ii) vehicle and (iii) glass. Apart from these, two real-world problems one from the remote sensing area and the other from the bio-informatics area are also considered. The objective in the remote sensing problem is to develop an accurate classifier using the spectral information to correctly identify the different types of vegetation from satellite imagery [23]. For the bio-informatics problem, the objective is to correctly carry a multi-category classification of different cancers from the micro-array gene expression data [21].

The organization of this paper is as follows: Section 2 describes the proposed SMC-RBF algorithm. In Section 3, we present the performance evaluation of the proposed classifier for three benchmark multi-category problems and two real-world applications. Section 4 summarizes the main conclusions from this study.

2. SMC-RBF algorithm

In this section, we describe the principles behind SMC-RBF first and then provide the various steps involved in the algorithm and finally summarize the algorithm in a pseudo-code form.

A multi-category classification problem can be described mathematically in the following manner. Let the training samples be $(\mathbf{X}_i, \mathbf{Y}_i)$, $i = 1, 2, \ldots, N$. $\mathbf{X}_i$ is an n-dimensional feature vector and $\mathbf{Y}_i$ is an n_c-dimensional vector of the

coded class labels, where n_c represents the total number of classes. For notational convenience, subscript i is left out in all further discussion.

If the training sample $\mathbf{X}$ belongs to the lth class, then the corresponding class label c will be l, i.e., the lth element of $\mathbf{Y}$ is one and all other elements are -1:

$$\begin{aligned} y_j &= 1, \quad j = l \\ &= -1 \quad \text{otherwise}, \ j = 1, 2, \ldots, n_c, \ j \neq l. \end{aligned} \tag{2}$$

The objective of the classification problem is to predict accurately the class label c of a new observation $\mathbf{X}$. This requires us to estimate the functional relationship between the class label and the feature space from the training set.

Generally, the output of the RBFN classifier has the following form:

$$\hat{y}_i = \sum_{j=1}^{K} w_{ij} \exp\left(-\frac{\|\mathbf{X} - \boldsymbol{\mu}_j^l\|}{2\sigma_j^2}\right), \quad i = 1, 2, \ldots, n_c, \tag{3}$$

where $\boldsymbol{\mu}_j^l$ is the jth neuron center corresponding to lth class and σ_j is the width of the jth neuron.

The predicted class label $\hat{c}$ for the new training sample is given by

$$\hat{c} = \arg\max_{j \in 1,2,\ldots,n_c} \hat{y}_j. \tag{4}$$

SMC-RBF classifier involves the allocation of new Gaussian hidden neurons as well as adapting the network parameters. The network begins with a zero hidden neuron (i.e., the network output is zero for the first sample). When the observation data are received sequentially, the network starts growing by using some of them as new hidden neurons based on certain criteria.

The criteria given below must be satisfied for a new training sample $(\mathbf{X}, \mathbf{Y})$ to add a new hidden neuron.

$$\begin{aligned} &\textbf{IF } (\hat{c} \neq c) \quad AND \quad [\|\mathbf{X} - \boldsymbol{\mu}_{nr}^c\| \geqslant \varepsilon_1 \ OR \ ERR \geqslant \varepsilon_2] \\ &\textbf{THEN} \text{ add a neuron,} \end{aligned} \tag{5}$$

where c is the actual class label, $\boldsymbol{\mu}_{nr}^c$ is the nearest neuron center of the class c, ERR is the sum of squared error ($ERR = \sum_i (e^i)^2$) for a new training sample (where e^i is given by Eq. (6)).

ε_1 and ε_2 are the thresholds to be selected appropriately. The threshold ε_1 defines the maximum spread of the Gaussian neurons, i.e., if the new training sample falls away from the nearest neuron of the same class by ε_1 then a new hidden neuron will be added to the SMC-RBF network. The ε_1 controls the growth/maximum number of neurons in the network. The threshold ε_2 controls the expected accuracy in the prediction.

In other sequential RBF algorithms, the error $(\mathbf{e})$ is usually the difference between the actual output and the network predicted output $(\mathbf{Y} - \hat{\mathbf{Y}})$ and this is used in the mean square error loss function. For classification problems, the above definition of error restricts the outputs of the SMC-RBF network between ± 1. In [27], it is shown that the classifier developed using a hinge loss function can estimate the posterior probability more accurately than the mean square error loss function. Hence, in our formulation, we use a hinge loss function to calculate the error $\mathbf{e}$ and this are given by

$$\begin{aligned} e^i &= y_i - \hat{y}_i \quad \text{if } y_i \hat{y}_i < 1 \\ &= 0 \quad \text{otherwise.} \end{aligned} \tag{6}$$

With this hinge loss function, the network output can grow beyond ± 1 and prevent the saturation problems in the learning process. The truncated outputs of the classifier model approximate the posterior probability accurately [27]. The truncated output is defined as

$$T(\hat{y}_i) = \min(\max(\hat{y}_i, -1), 1). \tag{7}$$

Since the target vectors are coded as -1 or 1, the posterior probability of observation vector $\mathbf{X}$ belonging to class c is

$$\hat{p}(c|\mathbf{X}) = \frac{T(\hat{y}) + 1}{2}. \tag{8}$$

If the criterion given in Eq. (5) is satisfied then a new hidden neuron $K + 1$ is added and its parameters are set as follows:

$$\mathbf{W}_{K+1} = \mathbf{e}_n, \tag{9}$$

$$\boldsymbol{\mu}_{K+1}^c = \mathbf{X}_n, \tag{10}$$

$$\sigma_{K+1} = \kappa \|\mathbf{X} - \boldsymbol{\mu}_{nr}^c\|, \tag{11}$$

where κ is a positive constant which controls the overlap between the hidden neurons.

When the new training sample does not satisfy the criterion for adding a new hidden neuron, the network parameters of the nearest hidden neuron in the same class (i.e., $\boldsymbol{\alpha} = [\mathbf{W}_{nr}, \boldsymbol{\mu}_{nr}^c, \sigma_{nr}]$) are adapted using a decoupled extended Kalman filter (DEKF) [20]. For the sake of notational convenience, we call nr as the nearest neuron in the same class. The computational complexity is reduced considerably in DEKF by ignoring the interdependencies of mutually exclusive groups of neurons, i.e., the cross-correlation terms of the error covariance matrix are neglected. In the proposed algorithm, since only the parameters of a single nearest neuron are updated, the training time is reduced considerably.

For the DEKF, the parameter update equations for new training sample is given by

$$\mathbf{B} = \Delta_{\boldsymbol{\alpha}} f(\mathbf{X_n}), \tag{12}$$

$$\mathbf{K} = \mathbf{P}^{nr}(n)\mathbf{B}[\mathbf{R} + \mathbf{B}^{\mathrm{T}}\mathbf{P}^{nr}(n)\mathbf{B}]^{-1}, \tag{13}$$

$$\mathbf{P}^{nr}(n+1) = [\mathbf{I} - \mathbf{K}\mathbf{B}^{\mathrm{T}}]\mathbf{P}^{nr}(n) + \mathbf{I}P_Q, \tag{14}$$

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \mathbf{K}\mathbf{e}, \tag{15}$$

where $\mathbf{R}$ is the variance of measurement noise, $\mathbf{B}$ is the partial derivative for the output signal with respect to the parameters $(\boldsymbol{\alpha})$, $\mathbf{P}^{nr}(n)$ is the error covariance matrix for the nearest neuron in the same class, and $\mathbf{K}$ is the Kalman gain. The addition of artificial process noise P_Q helps in avoiding convergence to local minima [20].

Here, the size of error covariance matrix $\mathbf{P}$ used to calculate the Kalman gain for any given sample is

$n+n_c+1 \times n+n_c+1$, which is less than the size of error covariance matrix $\mathbf{P}$ ($K(n+n_c+1) \times K(n+n_c+1)$, where K is number of hidden neurons) in EKF algorithm. From the above discussion, one can say that the DEKF algorithm requires less computational effort and memory requirement than the EKF algorithm. Hence, the proposed algorithm updates the parameters faster.

To summarize, the SMC-RBF algorithm in a pseudo-code form is given below.

2.1. SMC-RBF classification algorithm

Given an input feature $\mathbf{X}$ *and its corresponding coded label* $\mathbf{Y}$:

1. **COMPUTE** *the network output*

$$\hat{y}_i = \sum_{j=1}^{K} w_{ij} \exp\left(-\frac{\|\mathbf{X}-\boldsymbol{\mu}_j^l\|}{2\sigma_j^2}\right), \quad i = 1,2,\ldots,n_c,$$

where K is the number of hidden neurons.

2. **CALCULATE** *the error* $\mathbf{e}$:

$$e^i = y_i - \hat{y}_i \quad \text{if } y_i\hat{y}_i < 1$$
$$= 0 \quad \text{otherwise}$$

and the predicted class label

$$\hat{c} = \arg\max_{i\in 1,2,\ldots,n_c} \hat{y}_i.$$

3. **APPLY** *the criterion for adding neuron*:

IF $(\hat{c} \neq c)$ *AND* $[\|\mathbf{X}-\boldsymbol{\mu}_{nr}^c\| \geqslant \varepsilon_1$ *OR* $ERR_n \geqslant \varepsilon_2]$ *criterion is satisfied* **then** *add a new hidden neuron and set* $K = K+1$. *Allocate new hidden neuron with*

$$\mathbf{W}_{K+1} = \mathbf{e}_n,$$
$$\boldsymbol{\mu}_{K+1}^c = \mathbf{X}_n,$$
$$\sigma_{K+1} = \kappa\|\mathbf{X}-\boldsymbol{\mu}_{nr}^c\|.$$

ELSE *update the network parameters* $(\boldsymbol{\alpha} = [\mathbf{W}_{nr}, \boldsymbol{\mu}_{nr}, \sigma_{nr}])$ *of nearest neuron of the same class using DEKF.*

$$\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \mathbf{K}\mathbf{e}_n,$$

where $\mathbf{K}$ *is the Kalman gain.*
END IF

4. *Repeat the steps* 1–3 *until all training samples are presented to the network sequentially.*

3. Performance evaluation of SMC-RBF classifier

In this section, we present results on three benchmark classification problems from UCI Machine Repository, viz., image segmentation, vehicle and glass [2]. The detailed specification on number of inputs, number of classes and number of training/testing samples are given in Table 1. The classification problems range from smaller (features and samples) to larger ones. Apart from these problems, we have also presented the comparison on two real-world problems from the areas of remote sensing and bio-informatics. The details are given in Sections 3.3 and 3.4.

Table 1
Specification of UCI classification data set

Data set	# features	# classes	# samples	
			Training	Testing
Image segmentation	19	7	210	2100
Vehicle	18	4	424	422
Glass	9	6[a]	109	105

[a]Actual number of classes are 7 but there are no samples available for one class.

The performance of the SMC-RBF is compared with MRAN [26] and GAP-RBFN [7]. Also, we compare its performance with the SVM [5]. Even though SVM is a batch learning algorithm this comparison has been included here since SVM was originally designed specifically for classification problems. We also compare the performance of SMC-RBF classifier with the recently developed OS-ELM [14]. In OS-ELM, the input weights are selected randomly and the output weights are calculated analytically. For new set of training samples, only output weights are updated using a recursive least squares algorithm. In case of classification problems with fewer samples per class, random selection of input weights affects the performance of OS-ELM considerably [24]. Hence, a K-fold scheme for input weights selection as explained in [24] is used to select the initial weights in OS-ELM algorithm. It should also be noted that in OS-ELM the number of hidden neurons are fixed a priori, whereas in case of MRAN, GAP-RBFN and the proposed SMC-RBF, number of neurons is estimated online.

All the simulations are conducted in MATLAB 7.0 environment on a desktop PC with Pentium IV processor and 1 GB memory. The simulation study for SVM is carried out using the popular LIBSVM package in C [4]. In all examples, the inputs to algorithm are scaled appropriately between ± 1. The performance measures used to compare the classifiers are described below.

3.1. Performance measures

Two performance measures have been used in this study for comparing the classifiers, viz., classification efficiency and a statistical significance test.

3.1.1. Classification efficiency

The confusion matrix ($\mathbf{Q}$) is used to obtain the statistical measures for both the class-level and global performances of the classifier. Class-level performance is indicated by the

percentage classification which tells us how many samples belonging to a particular class have been correctly classified.

The percentage classification η_i for class c_i is

$$\eta_i = \frac{q_{ii}}{N_i^{\mathrm{T}}}, \tag{16}$$

where q_{ii} is the number of correctly classified samples and N_i^{T} is the number of samples for the class c_i in the testing data set. The global performance measures are the average (η_{a}) and overall (η_{o}) classification efficiency, which are defined as

$$\eta_{\mathrm{a}} = \frac{1}{n_c}\sum_{i=1}^{n_c} \eta_i, \tag{17}$$

$$\eta_{\mathrm{o}} = \frac{1}{N^{\mathrm{T}}}\sum_{i=1}^{n_c} q_{ii}, \tag{18}$$

where n_c is the total number of classes and N^{T} is the number of testing samples.

3.1.2. Statistical significance test

The classification efficiency itself may not indicate the statistical significance of accurate classification of the SMC-RBF over the other existing methods [22]. Hence, in this paper, we conducted the 'binomial test' [22] between SMC-RBF classifier and other classifiers (MRAN, GAP-RBFN, OS-ELM and SVM). For the purpose of understanding, we consider binomial test between SMC-RBF and MRAN. A brief description of the binomial test is given below.

Let N be the number of test samples for which the proposed classifier (SMC-RBF) and MRAN classifier produce different results. Let S (success) be the number of times SMC-RBF predicts the class label correctly rather than MRAN and F (failure) be the number of times MRAN predicts the class label correctly rather than SMC-RBF. Now, we find the p-value (probability of S success in N trials) using a binomial distribution as

$$E = \sum_{j=S}^{N} \frac{N!}{j!(N-j)!} p^j q^{N-j}, \tag{19}$$

where p and q are the probability of success for SMC-RBF and MRAN. If we expect no difference between the two algorithms then $p = q = 0.5$. If E is a very small number then we can say that the proposed SMC-RBF algorithm is better than the MRAN with high confidence.

3.2. Performance comparison for image segmentation, vehicle and glass problems

The performance comparison results for the three benchmark problems from the UCI machine repository are given below.

3.2.1. Image segmentation

Image segmentation involves partitioning an image into several constituent components. Segmentation is an important subsystem in any automated image recognition system, because it extracts the interesting objects for further processing such as description or recognition. In image segmentation problem, each image pixel is classified in to any one of the image parts (class). For each pixel, 19 different features are extracted as input vector. The image has seven different parts/classes with 2310 samples. For classifier development, 30 pixels from each class (as given in UCI repository) are used and remaining 2100 pixels are used for evaluating the performance. In this problem, we have equal number of samples for each class in training/testing set, i.e., 30 samples per class in training set and 300 samples per class in testing set. In SVM, the cost parameter C and Gaussian kernel width (γ) are optimized using cross-validation [4]. In case of OS-ELM, the number of hidden neurons and input weights affect the classifier performance considerably [24]. Hence, we use a K-fold cross-validation scheme to select number of neurons and input weights in OS-ELM algorithm. The parameters of MRAN, GAP-RBFN and proposed classifier are given in Table 2. The parameters of MRAN, GAP-RBFN, and SMC-RBF are optimized using a genetic algorithm approach and the best parameter values obtained are given in Table 2.

Since, the number of control parameters to be optimized in SMC-RBF classifier is three, it requires lesser computational effort than the other RBF classifiers. The neuron history for the training sample is shown in Fig. 4. From this figure, one can observe that MRAN and GAP-RBFN require 80 neurons to approximate the decision surface, whereas the proposed classifier requires only 43 neurons.

Table 2
Parameters for sequential algorithms

Problem	Method	Parameters
Image segmentation	MRAN	$e_1 = 0.09$, $e_2 = 0.04$, $\varepsilon_{\max} = 1.17$, $\varepsilon_{\min} = 0.08$, $\gamma = 0.93$, $\kappa = 0.78$, $N_w = 7$, $S_w = 20$, $\delta = 0.02$
	GAP-RBFN	$\varepsilon_{\max} = 1.2$, $\varepsilon_{\min} = 0.2$, $\gamma = 0.99$, $\kappa = 0.8$
	SMC-RBF	$\kappa = 0.573$, $\varepsilon_1 = 0.085$, $\varepsilon_2 = 0.1$
Vehicle	MRAN	$e_1 = 0.1$, $e_2 = 0.02$, $\varepsilon_{\max} = 0.5$, $\varepsilon_{\min} = 0.06$, $\gamma = 0.96$, $\kappa = 0.83$, $N_w = 18$, $S_w = 56$, $\delta = 0.45$
	GAP-RBFN	$\varepsilon_{\max} = 0.5$, $\varepsilon_{\min} = 0.1$, $\gamma = 0.94$, $\kappa = 0.8$
	SMC-RBF	$\kappa = 0.51$, $\varepsilon_1 = 0.07$, $\varepsilon_2 = 0.2$
Glass	MRAN	$e_1 = 0.1$, $e_2 = 0.02$, $\varepsilon_{\max} = 0.8$, $\varepsilon_{\min} = 0.08$, $\gamma = 0.91$, $\kappa = 0.8$, $N_w = 7$, $S_w = 20$, $\delta = 0.25$
	GAP-RBFN	$\varepsilon_{\max} = 0.8$, $\varepsilon_{\min} = 0.1$, $\gamma = 0.93$, $\kappa = 0.8$
	SMC-RBF	$\kappa = 0.57$, $\varepsilon_1 = 0.23$, $\varepsilon_2 = 0.1$

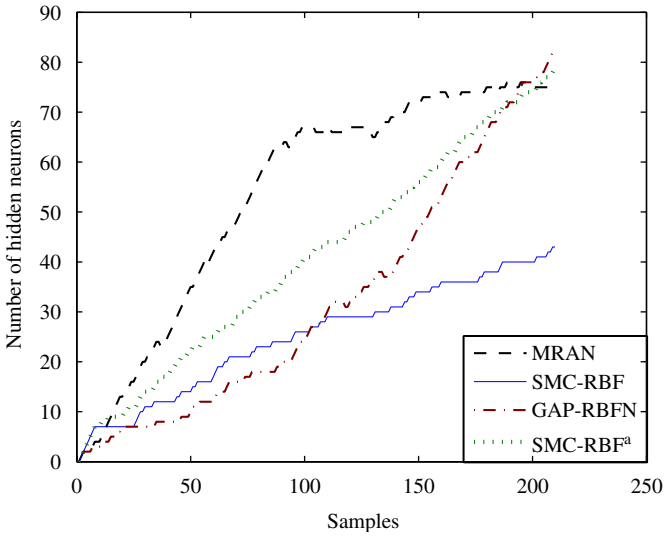

Fig. 4. Neuron history for image segmentation problem (SMC-RBF[a]) classifier without incorporating misclassification condition in the growing criterion).

Table 3
Performance comparison for image segmentation problem

Algorithm	N_H neurons	Training time (s)	Testing η_o	Binomial test		
				S	F	E
MRAN	76	783	86.52	160	66	$1.7e^{-10}$
GAP-RBFN	83	365	87.19	145	65	$1.7e^{-08}$
OS-ELM	100	21	90.67[a]	86	79	0.32
SVM	96	721	90.62	87	79	0.29
SMC-RBF[b]	79	148	89.64	102	85	0.12
SMC-RBF	43	142	91.00	–	–	–

Note that SVM is implemented in C, whereas the rest in MATLAB.
[a]Mean = 89.12 and variance = 1%.
[b]Without misclassification condition in growing criterion.

The number of hidden neurons, training time, and overall efficiency are given in Table 3. From the table, we can see that the proposed SMC-RBF classifier has better overall performance than the other classifiers. The overall classification efficiency is approximately 4% more than the MRAN and GAP-RBFN classifiers. Also, SMC-RBF requires minimal number of hidden neurons to approximate the decision surface. To study the effectiveness of the proposed classifier, we also compare the results with batch SVM algorithm and OS-ELM. In OS-ELM, the initial weights and output weights are calculated using randomly selected 150 training samples. The remaining 60 training samples are presented one-by-one and the output weights are updated online using recursive least square algorithm. The cost and width parameter of SVM are optimized using leave-one-out procedure [4]. SVM classifier model is developed for the best parameters. The performance of the OS-ELM and SVM classifiers are reported in Table 3.

The overall classification efficiency of proposed SMC-RBF classifier is marginally better than the batch SVM and OS-ELM classifiers. The SVM classifier development time is 5 times more than SMC-RBF classifier, and also the number of support vectors required to develop the classifier is twice the number of hidden neurons in SMC-RBF classifier. Classifier development time for OS-ELM algorithm is less but it requires more number of neurons to approximate the decision surface.

To study the statistical significance of the proposed SMC-RBF classifier, we conducted the binomial test with the MRAN, GAP-RBFN, OS-ELM and batch SVM classifiers. First, we conducted the test between the SMC-RBF classifier and MRAN classifier. From the 2100 test samples, the class labels obtained using MRAN and SMC-RBF classifiers differ only in 226 samples. Out of 226 samples, SMC-RBF classifier predicts class label accurately in 160 samples ($S = 160$) and MRAN classifier predicts the class label accurately in 66 samples ($F = 66$). The probability (E) for this case is 1.7×10^{-10}. From the result, we can say that the proposed SMC-RBF classifier is better than MRAN classifier with high confidence for this problem. Similarly, we conducted the binomial test for GAP-RBFN, OS-ELM and batch SVM classifiers and the results are given in Table 3. From the binomial tests, we can say that the performance of the SMC-RBF classifier is better than the MRAN and GAP-RBFN classifiers with high confidence and SMC-RBF performance is marginally better than OS-ELM and SVM classifiers.

To study the influence of misclassification condition in growing criterion, we also conducted another study for SMC-RBF classifier without incorporating misclassification condition in the growing criterion (SMC-RBF[b]). The training time, number of neurons and testing efficiencies are reported in Table 3. From the table, we can observe that the performance of SMC-RBF[b] is close to the batch SVM classifier and is inferior to the SMC-RBF classifier. Also, the SMC-RBF[b] classifier requires more number of hidden neurons than the SMC-RBF classifier. One can reduce the number of hidden neurons by increasing the value of distance criterion (ε_1). But increase in ε_1 value may not approximate the decision surface accurately in the class overlap region, which results in poor generalization efficiency. From the results, we can say that the distance criterion based on the nearest neighbor neuron in same class can improve the performance slightly but requires more number of neurons. Hence, we have to include the misclassification ($\hat{c} \neq c$) in growing condition explicitly to achieve better classification efficiency.

3.2.2. *Vehicle and glass—classification problems with sparse samples*

To study the behavior of the classifiers, we consider two classification problem called 'Vehicle' and 'Glass' problems from UCI machine learning repository. The vehicle and glass problem have fewer samples for classifier development and have strong overlap between the classes. The

number of training/testing samples and class information are given in Table 1. In case of vehicle problem, the number of training samples in each classes are: 119, 118, 98 and 89. In case of glass problem, the number of training samples in each classes are 35, 38, 9, 7, 5 and 15. For glass problem, the presence of fewer samples in class c_3, c_4 and c_5 affects the performance considerably for all the algorithms considered. Hence, for this case alone the training data are presented three times for all the algorithms. For OS-ELM algorithm, 75% of the training samples are presented first to estimate the input weights and output weights. The remaining 25% training samples are presented one-by-one for online adaptation of output weights. The classifier performances are evaluated using the test samples and the results are reported in Table 4.

From the table, we can say that the proposed classifier outperforms the other sequential algorithms and SVM approach in both vehicle and glass problems. The η_o and η_a for proposed SMC-RBF classifier are approximately 10–15% more than the MRAN and GAP-RBFN classifiers. Similar observation can be made with the OS-ELM and batch SVM classifiers. To study the statistical significance of the proposed algorithm, binomial tests are conducted and the results are given in Table 4. Since the probability values (E) are very small, we can say that the proposed classifier outperforms the MRAN, GAP-RBFN, OS-ELM and batch SVM classifiers with high confidence level.

From Table 4, we can observe that number of neurons required to develop a classifier for SMC-RBF is less than the other sequential classifiers. Also, from Table 4, the number of support vectors required in batch SVM classifier is more than the number of hidden neurons in SMC-RBF classifier. Since the proposed SMC-RBF classifier uses DEKF algorithm for network parameters update, the time taken to develop the model reduces considerably. Hence, the training time for the proposed algorithm is less than the MRAN, GAP-RBFN and batch SVM classifiers. Training time for OS-ELM classifier is less than the other classifiers but requires more number of hidden neurons.

3.2.2.1. Discussion. From the above studies, we can observe the following.

In the case when the decision surface between the classes are well separable and the number of samples per class are sufficient to describe the decision surface accurately, the performance of the different classifiers are almost same. This indicates that the criteria developed based on function approximation problem is sufficient to approximate the decision surface in such classification problems.

In the case where classes are not well separable and number of training samples for each class are not sufficient, the criteria developed based on function approximation problem may not be sufficient to approximate the decision surface for classification problems.

Glass and vehicle problems have high imbalance in number of training samples per class. Also, in case of glass problem, the feature set is greater than the samples per class (no. of samples are 9, 7 and 5 for class 3, 4 and 5, which is less than or equal to the no. of features 9). Hence, these training samples may not be sufficient to describe the decision surfaces between the classes accurately. In such problems, the criterion developed using function approximation problems is not sufficient. Hence, the MRAN, GAP-RBFN classifiers suffer from poor performances. Also, the abovementioned issues affect the performance of the OS-ELM [24] and batch SVM classifiers. Since the proposed SMC-RBF classifier uses misclassification error as a criterion, the performance is better than the other classifier algorithms.

The high imbalance in samples per class, large number of input features and strong overlap between classes are common problems in many real-world multi-category classification problems. In the next section, the results for two such examples (a remote sensing application [23] and a cancer classification problem in bio-informatics area [21]) are presented.

Table 4
Performance comparison for vehicle and glass benchmark problems

Data set	Algorithm	N_H neurons	Training time (s)	Testing		Binomial test		
				η_o	η_a	S	F	E
Vehicle	MRAN	100	520	59.94	59.83	103	43	$3.69e^{-07}$
	GAP-RBFN	81	452	59.24	58.23	98	35	$2.14e^{-08}$
	OS-ELM	300	36	68.95[a]	67.56	56	34	0.0131
	SVM	234	550	68.72	67.99	56	33	0.0095
	SMC-RBF	75	120	74.18	73.52	–	–	–
Glass[b]	MRAN	51	520	63.81	70.24	23	8	0.005
	GAP-RBFN	75	410	58.29	72.41	31	10	0.0007
	OS-ELM	60	15	67.62[c]	70.12	18	6	0.0113
	SVM	102	320	64.23	60.01	21	7	0.006
	SMC-RBF	58	97	78.09	77.96	–	–	–

[a]Mean = 62.27 and variance = 6%.
[b]For this problem, the training data was cycled for 3 times for all the algorithms.
[c]Mean = 62.52 and variance = 4.5%.

3.3. Remote sensing application

In this section, we consider a real-world remote sensing problem with large number of training samples but with only six spectral informations as input features and strong overlap among different classes. This together with the difficulty in obtaining the ground truth (original class number) increases the complexity for the model development. The objective here is to develop a classifier using the spectral information to identify correctly the different classes in the satellite imagery.

In our study, we consider a portion of high resolution, multi-spectral Landsat 7 Thematic-Mapper images acquired from southern region of India [23]. The wavelength range for the thematic mapper sensor is from the visible, through the mid-IR (Infra Red), into the thermal-IR portion of the electromagnetic spectrum. The portion of Landsat image is $15 \times 15.75\,km^2$ (500×525 pixels) and has 30 m spatial resolution, which corresponds to pixel spacing of 30 m. In our experimental study, the band 6 which has spatial resolution of 120 m is not used. The satellite image is radio-metrically and geometrically corrected using the satellite model and platform/ephemeris information. The image is further rotated and aligned to a user-defined map projection, using ground control points to improve the classifier. The color composite image of the study area and corresponding ground truth are given in Fig. 5. The image was taken when there is no cloud. Even though the image was taken during a no cloud day, small and minor cloud may present but not visible in the image. The presence of small and minor cloud may affect the performance of the classifier.

The aim of the study is to develop a neural network classifier to distinguish the nine classes described in the Fig. 5. Four classes are found in the image based on Level-I (Forest, Water, Vegetation and Built-up) standard [9]. Generally, Level-II (Deciduous, Deciduous-pine, Pine, Water, Agriculture, Bare ground, Grass, Urban and Shadow) would be of interest to the users in remote sensing application. Since the objective here is to develop a Level-II classifier model, the four classes are further divided into nine classes. The nine classes are defined based on the Level-II standards [9] and the map obtained using this approach can be directly used for practical applications. The ground truth is prepared for the Landsat image by visual inspection, auxiliary data (maps and aerial photography) and field work. For development of supervised neural classifiers, it is necessary to define the classes and select some of the region in the image for training of each class. The training data for each class are selected from the region where the field work has been carried-out [23]. The ground truth obtained for 10,000 pixels from field work are used to develop neural classifier models. The performance of the classifiers are evaluated using the remaining pixels. The details of the class, number of pixels obtained from field work and number of validation pixels are given in Table 5.

For OS-ELM, 3000 training samples (equally likely from all class) are selected to estimate the output weights for optimal input weights and number of hidden neurons (1500). The remaining 6000 samples are presented one-by-one to update the output weights. The parameters used in the other sequential algorithms are given in Table 6. The classifiers are tested with the remaining pixels class wise and global performance measures are calculated. The number of hidden neurons and performance measures for various classifiers are given in Table 7. From the table, we can observe that the SMC-RBF classifier requires more number of hidden neurons than MRAN and GAP-RBFN to approximate the decision surface accurately. But, the number of hidden neurons are much less than number of support vectors and OS-ELM classifier. The reason for more number of neurons in SMC-RBF classifier than GAP-RBFN and MRAN is due to the presence of strong overlap between classes in remote-sensing problem. For example, the deciduous-pine class and pine class falls in the same spectral range and are difficult to separate. Similar characteristic is observed in grass and agriculture classes. Hence, one has to use more number of neurons/support vectors to approximate the decision surface accurately.

From Table 7 we can see that the proposed SMC-RBF classifier has 93.43% average and 93.12% overall efficiency.

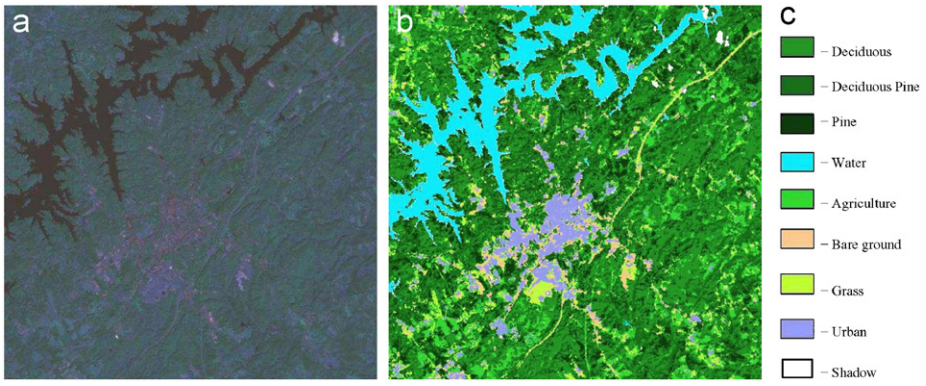

Fig. 5. Typical remotely sensed image and its ground truth. (a) Color composite image. (b) Ground truth. (c) False color for each class.

Table 5
Description of classes and ground truth available form field work and validation

Class no.	Hierarchial class		No. of pixels from field work	No. of pixels for validation
	Level-I	Level-II		
C1	Forest	Deciduous	1200	71,228
C2		Deciduous-pine	1300	80,848
C3		Pine	1100	24,911
C4	Water	Water	1075	12,518
C5	Vegetation	Agriculture	1100	23,070
C6		Bare ground	1100	26,986
C7		Grass	1050	7400
C8	Built-up	Urban	1075	11,636
C9		Shadow	1000	3547

Table 6
Parameters of sequential algorithms for remote sensing application

Method	Parameters
MRAN	$e_1 = 0.01$, $e_2 = 0.96$, $\varepsilon_{max} = 0.2$, $\varepsilon_{min} = 0.1$, $\gamma = 0.99$, $\kappa = 0.87$, $N_w = 30$, $S_w = 40$, $\delta = 0.05$
GAP-RBFN	$\varepsilon_{max} = 0.9$, $\varepsilon_{min} = 0.2$, $\gamma = 0.9$, $\kappa = 0.75$
SMC-RBF	$\kappa = 0.75$, $\varepsilon_1 = 1.5$, $\varepsilon_2 = 0.1$

Table 7
Performance measure for various classifier in remote-sensing application

	MRAN	GAP-RBFN	OS-ELM	SVM	SMC-RBF
N_H	226	275	1500	1298	631
Training time (min)	2542	1649	1255	1512	431
η_1	89.13	77.23	90.34	94.05	95.37
η_2	75.10	60.11	72.98	93.89	89.06
η_3	93.72	89.91	89.29	95.28	96.71
η_4	90.59	94.51	93.82	93.98	93.66
η_5	88.31	67.12	87.93	94.04	94.02
η_6	81.66	79.01	80.96	78.92	91.20
η_7	75.76	82.23	77.95	86.41	88.69
η_8	87.50	85.45	83.85	86.70	95.72
η_9	96.62	92.64	94.62	87.99	96.45
η_a	86.48	80.49	85.75	90.14	93.43
η_o	84.45	74.45	75.32	92.21	93.21
E	2.37×10^{-10}	6.78×10^{-15}	7.12×10^{-13}	4.21×10^{-6}	–

The performance is approximately 10% more than the other sequential classifiers. Here, 1% increase in cumulative class-level efficiency is due to 9% (9% because we have nine classes in this problem) increase in class-level efficiency. Hence, the proposed classifier performs better in the class-level as well as in overall sense. Also, the statistical significance test results show that the proposed SMC-RBF classifier is much better than the other classifiers. Fig. 6 shows the classified image using the proposed classifier algorithm. By comparing the ground truth (Fig. 5(b)) and the classified image (Fig. 6), we can see that the land cover map obtained using proposed classifier emulates the ground truth accurately and hence it is more suitable for practical applications.

3.4. Gene expression based cancer classification problem

Cancer detection and classification using standard clinical procedure is a difficult process. Also, the accurate estimation of human cancer based on anatomic size origin is an important component in the cancer treatment process. Hence, in recent years bio-medical research activities are focused towards identification of human cancer and molecular classification of cancer. Recently in [21], gene expression data is used to classify the human cancer types using SVM. Gene expression signatures database is a collection of micro-array data for human tumor and normal tissue specimens. The database was collected from six medical institutions for 14 different tumors. Each tissue specimen consists of 16,063 genes and the database has 198 primary tumor samples for 14 common tumor types. Since, eight samples belong to metadata, we neglect the metadata

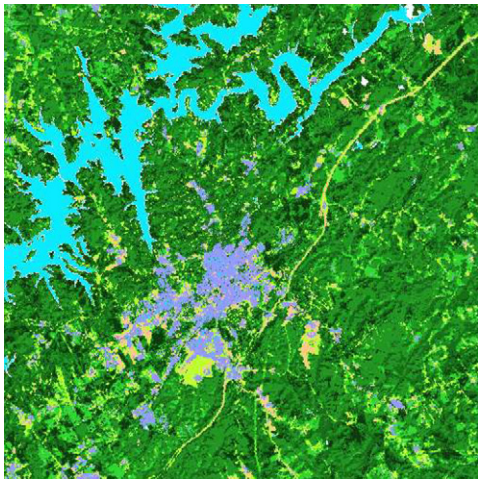

Fig. 6. Classified remotely sensed image using proposed classifier.

Table 8
Parameters of sequential algorithms for cancer detection problem

Method	Parameters
MRAN	$e_1 = 0.01$, $e_2 = 0.96$, $\varepsilon_{max} = 0.4$, $\varepsilon_{min} = 0.28$, $\gamma = 0.99$, $\kappa = 0.87$, $N_w = 10$, $S_w = 10$, $\delta = 0.000001$
GAP-RBFN	$\varepsilon_{max} = 2.0$, $\varepsilon_{min} = 0.4$, $\gamma = 0.97$, $\kappa = 0.8$
SMC-RBF	$\kappa = 0.75$, $\varepsilon_1 = 0.9$, $\varepsilon_2 = 0.1$

Table 9
Performance measure for various classifiers in cancer classification

Method	N_H neurons	Training (η_o)		Training time (min)	Testing (η_o)	
		Mean	Std. Dev.		Mean	Std. Dev.
MRAN	25	90.32	3.12	8.21	73.23	3.12
GAP-RBFN	45	91.34	2.09	4.51	78.56	4.96
OS-ELM	50	92.30	2.25	9.29	79.43	6.23
SVM	106	96.50	1.85	29.45	73.78	5.10
SMC-RBF	64	95.21	2.01	2.12	83.21	4.21

in our study. This classification problem is a typical example where the number of feature space is larger than the number of samples available. Hence, recursive feature elimination method is used to identify most significant genes. Using the genes as inputs to the neural network, the classifier models are developed for different number of most significant genes. The simulation results clearly show that the performance increases with increase in number of significant genes as input to the network. The performance saturates when the number of most significant genes reaches 100. Hence, we use maximum of 98 significant genes for the classifier development.

Out of 190 patterns, 144 patterns are randomly selected for training and 46 patterns are used for validation. From the training data one can easily observe that the number of samples per class varies from 8 to 30. The imbalance in the training data and fewer training samples affect the performances of classifier model considerably. Hence, in this paper, we select 98 most significant genes from the complete 16,063 genes for our study.

The parameters for MRAN, GAP-RBFN and SMC-RBF algorithms are given in Table 8. The classifier model is developed using randomly selected 144 samples from the 190 samples. The remaining samples are used to test the generalization performance of the classifier. Since we have small number of data for model development and testing, 100 random trails (100 different splits) are used to calculate the statistical significance parameters of the models. Since 100 random trials are conducted for this problem, we present the mean and standard deviation of performance than the statistical significance test. The presence of strong overlap between the classes and sparse distribution of samples are responsible for higher number of neurons in SMC-RBF classifier. The results for different learning algorithms are listed in the Table 9. The overall testing accuracy for the proposed sequential classifier is 83.21% and is approximately 4% more than the other classifiers. Also, the training time is less than that of the other sequential algorithms and SVM. The gene expression data based cancer detection problem is also a sparse data classification problem. For such problems, proposed SMC-RBF classifier performs better than the batch SVM and other online sequential classifiers. The factors such as misclassification and similarity measure within the class/cluster in growth criterion helps SMC-RBF to find a compact network with higher generalization performance.

4. Conclusion

In this paper, we present a new sequential multi-category classifier using RBF. The growth criterion in the SMC-RBF combines the similarity measure within the class, misclassification error and approximation error to achieve a more compact network with a higher classification accuracy. The computational complexity is also reduced by using a decoupled EKF. Performance of SMC-RBF has been evaluated using three benchmark multi-category classification problems from UCI and real-world problems from remote sensing and gene expression based cancer detection area. Comparison with other existing sequential classifiers clearly show that the use of similarity measure within the class and misclassification in the growth criterion improves the performance of the proposed SMC-RBF classifier considerably. The study indicates that using a function approximation algorithm for classification problems may not work well when the training data is not uniformly distributed among the classes which are not well separated.

Acknowledgments

The authors would like to thank the reviewers for their comments and suggestions which has improved the quality of presentation.

References

[1] U. Anders, O. Korn, Model selection in neural networks, Neural Networks 12 (1999) 309–323.
[2] C. Blake, C. Merz, UCI Repository of Machine Learning Databases, Department of Information and Computer Sciences, University of California, Irvine, URL: ⟨www.ics.uci.edu/m̃learn/MLRepository.html⟩, 1998.
[3] S. Chakravarthy, J. Ghosh, Scale-based clustering using the radial basis function network, IEEE Trans. Neural Networks 7 (5) (1996) 1250–1261.
[4] C.-C. Chang, C.-J. Lin, LIBSVM: A Library for Support Vector Machines, Taiwan, Department of Computer Science and Information Engineering, National Taiwan University, URL: ⟨http://www.csie.ntu.edu.tw/cjlin/libsvm/⟩, 2003.

[5] N. Cristianini, J.S. Taylor, An Introduction to Support Vector Machines, Cambridge University Press, Cambridge, UK, 2000.
[6] F. Fukushima, N. Wake, Handwritten alphanumeric character recognition by the neocognitron, IEEE Trans. Neural Networks 2 (3) (1991) 355–365.
[7] G.-B. Huang, P. Saratchandran, N. Sundararajan, An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks, IEEE Trans. Syst. Man Cybern. B 34 (6) (2004) 2284–2292.
[8] J.-N. Hwang, S.-Y. Kung, M. Niranjan, J.C. Principe, The past, present, and future of neural networks for signal processing, IEEE Signal Process. Mag. 14 (6) (1997) 28–48.
[9] J.R. John, Remote Sensing of the Environment: An Earth Resource Perspective, Prentice-Hall, Upper Saddle River, NJ, 2000.
[10] V. Kadirkamanathan, M. Niranjan, A function estimation approach to sequential learning with neural networks, Neural Comput. 5 (1993) 954–975.
[11] T. Kavzoglu, P.M. Mather, The use of backpropagating artificial neural networks in land cover classification, Int. J. Remote Sensing 24 (2003) 4907–4938.
[12] C. Lee, D.A. Landgrebe, Decision boundary feature extraction for neural networks, IEEE Trans. Neural Networks 8 (1) (1997) 75–83.
[13] A. Leonardis, H. Bischof, An efficient MDL-based construction of RBF networks, Neural Networks 11 (1998) 963–973.
[14] N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate on-line sequential learning algorithm for feedforward networks, IEEE Trans. Neural Networks 17 (6) (2006) 1411–1423.
[15] Y.W. Lu, N. Sundararajan, P. Saratchandran, A sequential minimal radial basis function (RBF) neural network learning algorithm, IEEE Trans. on Neural Networks 9 (2) (1998) 308–318.
[16] J. Moody, C.J. Darken, Fast learning in network of locally-tuned processing units, Neural Comput. 1 (1989) 281–294.
[17] M.T. Musavi, W. Ahmed, K.H. Chan, K.B. Faris, D.M. Hummels, On training of radial basis function classifiers, Neural Networks 5 (4) (1992) 595–603.
[18] J. Park, I.W. Sandberg, Universal approximation using radial basis function networks, Neural Comput. 3 (1991) 246–257.
[19] J.C. Platt, A resource allocating network for function interpolation, Neural Comput. 3 (1991) 213–225.
[20] G.V. Puskorius, L.A. Feldkamp, Neurocontrol of nonlinear dynamics systems with Kalman filter trained recurrent networks, IEEE Trans. Neural Networks 5 (2) (1994) 279–297.
[21] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, Multi-class cancer diagnosis using tumor gene expression signatures, Proc. Natl. Acad. Sci. 98 (26) (2002) 15149–15154.
[22] S.L. Salzberg, On comparing classifiers: pitfalls to avoid and a recommended approach, Data Min. Knowl. Discovery 1 (3) (1997) 317–328.
[23] S. Suresh, V. Mani, S.N. Omkar, N. Sundararajan, Multi-spectral satellite image classification using an evolving neural network approach, J. Aerosp. Sci. Technol.—Aero. Soc. India 58 (4) (2006).
[24] S. Suresh, S. Saraswathi, N. Sundararajan, Performance enhancement of extreme learning machine for multi-category sparse cancer classification, Pattern Anal. Appl., 2007, www.ntu.edu.sg/home/ssundaram/rcgaelm.pdf.
[25] L. Yan, N. Sundararajan, P. Saratchandran, Analysis of minimal radial basis function network algorithm for real-time identification of nonlinear dynamic systems, IEE Proc. D Control Theory Appl. 147 (4) (2000) 476–484.
[26] L. Yingwei, N. Sundararajan, P. Saratchandran, A sequential learning scheme for function approximation using minimal radial basis function (RBF) neural networks, Neural Comput. 9 (2) (1997) 461–478.
[27] T. Zhang, Statistical behavior and consistency of classification methods based on convex risk minimization, Ann. Stat. 32 (1) (2003) 56–85.

Sundaram Suresh received his B.E. degree in Electrical and Electronics Engineering from Bharathiyar University in 1999, his M.E. degree in Aerospace Engineering from Indian Institute of Science, in 2001 and Ph.D degree in Aerospace from Indian Institute of Science, in 2005. Since 2005, he is with School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore, where he is currently working as a Research Fellow. His research interest includes adaptive flight control, unmanned aerial vehicles, neural networks, parallel computing and computer vision.

Narasimhan Sundararajan received his B.E. in Electrical Engineering with First Class Honors from the University of Madras in 1966, M.Tech from the Indian Institute of Technology, Madras in 1968 and Ph.D. in Electrical Engineering from the University of Illinois, Urbana-Champaign in 1971.

From 1972 to 1991, he was working in the Indian Space Research Organization, Trivandrum, India, starting from Control System Designer to Director, Launch Vehicle Design Group contributing to the design and development of the Indian satellite launch vehicles. He has also worked as an NRC Research Associate at NASA—Ames in 1974 and as a Senior Research Associate at NASA Langley during 1981–1986. From February 1991, he is working in the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, first as an associate professor (from 1991 to August 1999), and presently as a Professor. He was a "Prof. I.G. Sarma Memorial ARDB Professor (an endowed visiting professor) during November 2002–February 2003, at the School of Computer Science and Automation, Indian Institute of Science, Bangalore, India.

His research interests are in the areas of aerospace control, neural networks and parallel implementations of neural networks and has published more than 130 papers and also four books titled "Fully Tuned Radial Basis Function Neural Networks for Flight Control" (Kluwer, Boston, MA, 2001), "Radial Basis Function Neural Networks with Sequential Learning" (World Scientific Publishing Co., 1999) "Parallel Architectures for Artificial Neural Networks" (IEEE Computer Society Press, 1998) and "Parallel Implementations of Backpropagation Neural Networks" (World Scientific Publishing Co., 1996).

Dr. Sundararajan is a Fellow of IEEE, an Associate Fellow of AIAA and also a Fellow of the Institution of Engineers (IES), Singapore. He was an Associate Editor for IEEE Transactions on Control Systems Technology, IFAC Journal on Control Engineering Practice (CEP), IEEE Robotics and Automation Magazine and for Control—Theory and Advanced Technology (C-TAT), Japan. He was also a member of the Board of Governors (BoG) for the IEEE Control System Society (CSS) for 2005. He has contributed as a program committee member in a number of international conferences and was the General Chairman for the Sixth International Conference On Automation, Robotics, Control and Computer Vision—ICARCV2000 held in Singapore in December 2000. He is listed in Marquis Who's Who in Science and Engineering, and Men of Achievement, International Biographical Center, Cambridge, UK.

P. Saratchandran received his Ph.D. degree in the area of control engineering from Oxford University, London, UK. He is an Associate Professor with the Nanyang Technological University, Singapore. He has several publications in refereed journals and conferences and has authored four books titled Fully Tuned RBF Networks for Flight Control (Kluwer, Norwell, MA, 2002), Radial Basis Function Neural Networks with Sequential Learning (World

Scientific, Singapore, 1999), Parallel Architectures for Artificial Neural Networks (IEEE Press, Piscataway, NJ, 1998), and Parallel Implementations of Backpropagation Neural Networks (World Scientific, Singapore, 1996). His interests are in neural networks, bioinformatics, and adaptive control. Dr. Saratchandran is an Editor for Neural Parallel and Scientific Computations. He is listed in the Marquis Who's Who in the World and in the Leaders in the World, International Biographics Centre, Cambridge, UK.