

行政院國家科學委員會專題研究計畫 成果報告

RBF Network 之遞增式學習演算法之研究

計畫類別：個別型計畫

計畫編號：NSC92-2213-E-002-095-

執行期間：92年08月01日至93年10月31日

執行單位：國立臺灣大學資訊工程學系暨研究所

計畫主持人：歐陽彥正

計畫參與人員：陳倩瑜 歐昱言

報告類型：精簡報告

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢

中 華 民 國 94 年 4 月 30 日

行政院國家科學委員會補助專題研究計畫 成果報告
 期中進度報告

RBF Network 之遞增式學習演算法之研究

計畫類別： 個別型計畫 整合型計畫

計畫編號：92-2213-E-002-095

執行期間： 92 年 8 月 1 日至 93 年 10 月 31 日

計畫主持人：歐陽彥正

共同主持人：陳倩瑜

計畫參與人員： 歐昱言

成果報告類型(依經費核定清單規定繳交)： 精簡報告 完整報告

本成果報告包括以下應繳交之附件：

- 赴國外出差或研習心得報告一份
- 赴大陸地區出差或研習心得報告一份
- 出席國際學術會議心得報告一份
- 國際合作研究計畫國外研究報告書一份

處理方式：除產學合作研究計畫、提升產業技術及人才培育研究計畫、
列管計畫及下列情形者外，得立即公開查詢

涉及專利或其他智慧財產權， 一年 二年後可公開查詢

執行單位：國立台灣大學資工學系

中 華 民 國 94 年 4 月 30 日

A Novel Radial Basis Function Network Classifier with Centers Set by Hierarchical Clustering

Yu-Yen Ou and Yen-Jen Oyang

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
E-mail: {yien,yjoyang}@csie.ntu.edu.tw

Chien-Yu Chen

Graduate School of Biotechnology and Bioinformatics
Department of Computer Science and Engineering
Yuan Ze University, Chung-Li, Taiwan
E-mail: cychen@mars.csie.ntu.edu.tw

摘要—本論文提出一個新方法建構半徑式基底函數網路分類器。我們的貢獻可分為兩個部分：其一是開發漸進式階層分群演算法於建構網路中的隱藏層，其二則是提升用於計算網路中隱藏層與輸出層之間權重的最小平方錯誤方法之品質。這篇論文討論使用漸進式階層分群演算法在建構 RBF 網路最佳化於資料分類的問題上所產生的影響。其資料群的形成是由訓練資料的所屬類別所控制，因此所產生的分群結果得以適當地描述訓練子集於各個區域空間的分布。除此之外，此漸進式的架構大大地減少處理大量資料時記憶體空間的需求。針對網路中權重參數的決定，我們使用迴歸理論來解決於尋找最佳權重時常常面臨的奇異矩陣的問題。實驗結果顯示我們所建構的分類器可以提供與支持向量機器分類器(SVM)或是我們最近提出的以核心密度推估方法為基礎的分類器一樣好的分類準確度，且同時提供高效能於處理那些有高度重複特性的資料集。

Abstract—This paper proposes a novel method to construct a radial basis function network (RBFN) classifier. Our contribution consists of two parts. The first one is an incremental hierarchical clustering algorithm for constructing the hidden layer, and the second one is to improve the least mean square error method that calculates the weights between the hidden and the output layers of an RBFN. This paper discusses the effects of incorporating an incremental hierarchical clustering algorithm for constructing an RBFN optimized for data classification applications. The formation of clusters is controlled by the class labels of training samples and therefore the clusters identified are well adapted to the local distributions of training instances. In addition, the incremental framework largely reduces the requirement of memory space when the training data set is large. In regard to the calculation of weights, we employ the regularization theory to solve the singular matrix problem that might happen in determining the optimal weights. Experimental results show that the data classifier constructed is capable of delivering comparable classification accuracy as the support vector machine (SVM) and the kernel density estimation based classifier that we have recently proposed, while enjoying significant execution efficiency in handling data sets that contains a high percentage of redundant training instances.

I. INTRODUCTION

The radial basis function network (RBFN) is a special type of neural networks with several distinctive features [1], [2], [3], [4], [5], [6]. Since its first proposal, the RBFN has attracted a high degree of interest in research communities. An RBFN consists of three layers, namely the input layer, the hidden layer, and the output layer. The input layer broadcasts the coordinates of the input vector to each of the nodes in the hidden layer. Each node in the hidden layer then produces an activation based on the associated radial basis function. Finally, each node in the output layer computes a linear combination of the activations of the hidden nodes. How an RBFN reacts to a given input stimulus is completely determined by the activation functions associated with the hidden nodes and the weights associated with the links between the hidden layer and the output layer. The general mathematical form of the output nodes in an RBFN is as follows:

$$c_j(x) = \sum_{i=1}^k \omega_{ji} \phi(\|x - \mu_i\|; \sigma_i), \quad (1)$$

where $c_j(x)$ is the function corresponding to the j -th output unit (class- j) and is a linear combination of k radial basis functions $\phi()$ with center μ_i and bandwidth σ_i . Also, w_j is the weight vector of class- j and w_{ji} is the weight corresponding to the j -th class and i -th center. The general architecture of RBFN is shown in Fig 1.

In this paper, we select the spherical (or symmetrical) Gaussian function as our basis function of RBFN, so the Eq.1 becomes:

$$c_j(x) = \sum_{i=1}^k \omega_{ji} \exp\left(-\frac{\|x - \mu_i\|^2}{2\sigma_i^2}\right) \quad (2)$$

From Eq.2, we can see that constructing an RBFN involves determining the number of centers, k , the center locations, μ_i , the bandwidth of each center, σ_i , and the weights, w_{ji} . That is, training an RBFN involves determining the values of three sets of parameters: the centers (μ_i), the bandwidths (σ_i), and the weights (w_{ji}), in order to minimize a suitable cost function.

Basically, there are two categories of learning algorithms proposed for RBFNs [5], [7]. The first category of learning algorithms simply places one radial basis function at each sample [8], [9]. On the other hand, the second category of learning algorithms attempt to reduce the number of hidden nodes in the network, or equivalently the number of radial basis functions in the linear function above [10], [11], [12], [13], [14]. One primary motivation behind the design of the second category of algorithms is to reduce the complexity of the network constructed. The typical procedure incorporated in the second category of learning algorithms conducts a clustering analysis on the training instances and then allocates one hidden node for each cluster of instances. In this regard, the effects of a wide variety of clustering algorithms have been investigated [4], [15]. Nevertheless, both the conventional agglomerative hierarchical clustering algorithm and the conventional partitioning algorithm suffer some kinds of deficiencies. The main problem with the conventional agglomerative hierarchical clustering algorithm is its space complexity of $O(n^2)$, where n is the number of training instances, due to the need to store pairwise distances or similarity scores between the training instances. The main problem with the conventional partitioning clustering algorithm is that the user needs to figure

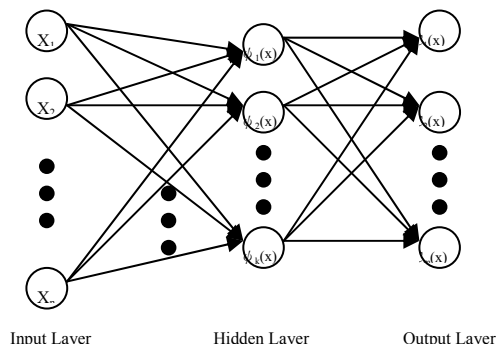


Fig. 1. General Architecture of Radial Basis Function

out how many clusters are appropriate for the given training data set.

In 1997, Hwang et al. [12] proposed an incremental clustering based approach for determining the locations of hidden nodes in the RBFN to be constructed. The incremental approach enjoys several advantages. First, it does not need to compute all the pairwise distances or similarity scores between training instances. The key issue in this regard is that the space complexity for storing the pairwise distances or similarity scores is greatly reduced, in addition to lower time complexity. Second, it figures out the number of clusters automatically based on a user-specified parameter. Third, it executes more efficiently than the conventional agglomerative hierarchical clustering algorithm and the conventional partitional clustering algorithm. Nevertheless, the incremental clustering algorithm proposed by Hwang employs a fixed threshold of radius to control the formation of clusters. As a result, the clusters identified may not be well adapted to the local distributions of training instances. For example, in a region with a low local density of training instances, the threshold of radius for controlling the formation of clusters should be set to a large value. On the other hand, in a region with a high local density of training instances, the threshold of radius should be set to a small value.

This paper proposes a novel method to construct an RBFN classifier by using an incremental hierarchical clustering algorithm for constructing an RBFN optimized for data classification applications. Our contribution consists of two parts. The first one is an incremental hierarchical clustering algorithm that constructs the hidden layer effectively and efficiently. Since the clustering algorithm is hierarchical, the formation of clusters is controlled by the class labels of training samples instead of a fixed threshold and therefore the clusters identified are well adapted to the local distributions of training instances. In addition, because the clustering algorithm is incremental, it does not need to compute all the pairwise distances or similarity scores between training instances. The second part is

A. Incremental framework

We adopt the incremental framework proposed in our previous work [20]. This section describes how the incremental algorithm works. The incremental algorithm operates in two phases, initial phase and incremental phase. In both phases, it invokes the complete-link algorithm to construct a clustering dendrogram.

1) Initial phase: In the incremental algorithm, it is assumed that all the incoming data instances are first buffered in an incoming data pool. In the first phase of the algorithm, a number of data instances are taken from the incoming data pool and the complete-link algorithm is invoked to cluster these instances build a tentative dendrogram. We can assume that these data instances are selected sequentially according to the order of input sequence. As demonstrated in our previous work [20], the proposed incremental framework employs two operations, split and merge, to reduce the influence from input ordering. When the

an improved least mean square error method that calculates the weights between the hidden and the output layers of an RBFN. In [12], authors proposed an improved method which uses a smaller matrix to compute the weights. The method proposed by [12] is more efficient and practical than the traditional one, but it may suffer the singular matrix problem and fails to build the classifier in such case. We solve the singular matrix problem by using the regularization theory in this paper, and then propose a method that can obtain the optimal weights analytically and efficiently.

Experimental results show that the data classifier constructed is capable of delivering comparable classification accuracy as the SVM [16] and the novel kernel density estimation (KDE) based classifier that we have recently proposed [8], while enjoying significant execution efficiency in handling data sets that contains a high percentage of redundant training instances. For example, in the experiment with the shuttle data set in the UCI repository [17], the mechanism proposed in this paper enjoys 1231 times and 259 times speedup over the SVM and the KDE based classifier that we have recently proposed, respectively, for constructing a data classifier. In addition, the mechanism proposed in this paper delivers comparable execution efficiency as the SVM in the prediction phase and enjoys 481 times speedup over the KDE based classifier in this regard. Experimental results also reveal that the approaches that have been proposed in recent years for solving the efficiency issues of the SVM and the KDE based classifier all lead to slight degradation of classification accuracy.

This paper is organized as follows. In next section, we introduce an incremental clustering method. In Section III and IV, we detail how to calculate the bandwidths and weights of the radial basis functions which are employed in constructing the RBFN. Next, numerical experiments are shown in Section V. Finally, we have some discussions and conclusions in Section VI.

II. DETERMINING THE CENTERS

In the proposed hierarchical approach, a hierarchical agglomerative clustering (HAC) algorithm [18], [19] is invoked to cluster all the instances in training data set. After hierarchical clustering terminates, the class labels are applied to the dendrogram to derive target clusters. Each node in the clustering dendrogram corresponds to a cluster of data instances. A node in the dendrogram is identified as a target cluster if it contains only data instances from a single class and its parent does not satisfy the criterion. The centroids of the target clusters are used as the centers in constructing the hidden layer of RBFN. In this paper, the complete-link algorithm [19] is employed. The reason of employing the complete-link algorithm is its tendency to find spherical clusters. Since the hierarchical clustering algorithms suffer higher time complexity, an incremental clustering framework for expediting the hierarchical clustering process is introduced as follows.

criterion invoked in the first phase are invoked again to find the target clusters from the new tentative dendrogram. During the reconstruction process, two original target clusters will have chance to form a new bigger target cluster. This is regarded as the so-called merge operation.

The incremental phase repeats until there is no data instances left in the incoming data pool. After the clustering process terminates, the centroids of all the target clusters are collected as the centers of RBFN when constructing the classifier in the following sections.

III. CALCULATION OF THE BANDWIDTHS

For the hidden layer of the RBFN classifier, we use the proposed hierarchical approach to determine the number of the nodes and their center locations. Another parameter to be decided for each node in the hidden layer is the bandwidth of its kernel function, σ_i . Here, we employ the method presented by Moody and Darken [21] to determine the bandwidth of each kernel function. The bandwidth of a kernel

complete-link algorithm terminates, target clusters are derived by the method described above, i.e. the class labels are used to identify the cluster boundaries in the clustering hierarchy.

There are four pieces of information recorded for each target cluster: (1) the centroid, (2) the radius, (3) the class label and (4) the number of instances in the cluster. The radius of a cluster is defined to be the maximum distance between the centroid and the data instances in this cluster.

2) Incremental phase: In the second phase of the incremental algorithm, the data instances remained in the incoming data pool are examined one by one. For each new data instance, the algorithm will find its nearest neighbor in the set of target clusters. If the distance between the new data instance and its nearest target cluster is smaller than the radius of the target cluster, the new data instance is inserted into the target cluster. If not, the data instance is currently an outlier to the set of target clusters and is therefore put into the tentative outlier buffer temporarily. The data instance, however, may form a target cluster with other data instances that are already in the tentative outlier buffer or that come in later.

If a data instance is successfully inserted into an existing target cluster, we should check if the new data instance possesses the same class label with the other data instances in the target cluster. If not, an additional operation called split should be invoked to identify new target clusters in this local area. In the split operation, we apply the complete-link algorithm only to the data instances in this target cluster, and identify new target clusters with pure property as we did in the first phase. After the split operation finishes, the number of target clusters will increase at least by one.

Once the number of data instances in the tentative outlier buffer exceeds a threshold, the complete-link algorithm is invoked again to construct a new tentative dendrogram. In this reconstruction process, the primitive objects are the target clusters and the data instances in the tentative outlier buffer. In this case, each target cluster is represented by its centroid and regarded as a single data instance. When a new tentative dendrogram has been generated, the same procedure and

function is set as β denemy, where denemy is the distance to the center of the nearest cluster which belong to a different class and β is a constant. In this paper, we follow the heuristic setting suggested by [12], i.e. $\beta = 5$.

IV. CALCULATION OF THE WEIGHTS

After the centers and bandwidths of the kernel functions in hidden layer have been determined, the transformation between the inputs and the corresponding outputs of the hidden units is now fixed. The network can thus be viewed as an equivalent single-layer network with linear output units. Then, we use the least mean square error method to determine the weights associated with the links between the hidden layer and the output layer.

In this section, we will show how the least mean square error method have been used in data classification field, and then propose a method which has a better theoretical foundation and practical use.

Assume h is the output of the hidden layer.

$$h = [\phi_1(x) \quad \phi_2(x) \quad \dots \quad \phi_k(x)]^T, \quad (3)$$

where k is the number of centers, $\phi_1(x)$ is the output value of first kernel function with input x . Then, the discriminant function $c_j(x)$ of class- j can be expressed by the following:

$$c_j(x) = \omega_j^T h, \quad j = 1, 2, \dots, m \quad (4)$$

where m is the number of class, and ω_j is the weight vector of class- j . We can show ω_j as:

$$\omega_j = [\omega_{j1} \quad \omega_{j2} \quad \dots \quad \omega_{jk}]^T \quad (5)$$

After calculating the discriminant function value of each class, we choose the class with the biggest discriminant function value as the classification result. We will discuss how to get the weight vectors by using least mean square error method in the following subsections.

A. Traditional Least Mean Square Error Method

The traditional least mean square error method was proposed by Broomhead and Lowe [22]. This method is originally proposed for function approximation, and is the most popular supervised learning method of constructing the weights of RBFN [2], [3], [5], [23]. In this method, the objective function of class- j can be shown as:

$$\min \sum_{i=1}^n [c_j(x_i) - v_j(x_i)]^2 \quad (6)$$

where

$$v_j(x_i) = \begin{cases} 1 & \text{if } x \in \text{class-}j, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

This system is overconstrained, being composed of n equations with k unknown weights, then the optimal solution of ω_j can be written as

$$\omega_j^* = \Phi^+ y_j, \quad (8)$$

where $y_j = [v_j(x_1) \quad v_j(x_2) \quad \dots \quad v_j(x_n)]^T$, $\Phi_{ji} = \phi_i(x_j)$ and Φ^+ is the pseudoinverse of Φ . The matrix Φ is rectangular ($n \times k$) and its pseudoinverse can be computed as

$$\Phi^+ = (\Phi^T \Phi)^{-1} \Phi^T,$$

provided that $(\Phi^T \Phi)^{-1}$ exists. The matrix $(\Phi^T \Phi)$ is square and its dimensionality is k , so that it can be inverted in time proportional to k^3 .

Although in theory the quantity of $(\Phi^T \Phi)^{-1}$ exists, the cost of computing Φ^+ is very high. First, we need to store Φ of size $(n \times k)$ in the

To find the optimal W that minimizes J , we set the gradient of $J(W)$ to be zero:

$$\nabla_w J(W) = 2 \sum_{j=1}^m P_j E_j \{h h^T\} W - 2 \sum_{j=1}^m P_j E_j \{h\} V_j^T = [0], \quad (11)$$

where $[0]$ is a $k \times m$ null matrix.

Let K_i denote the class-conditional matrix of the second order moments of h , i.e.

$$K_i = E_i \{h h^T\}. \quad (12)$$

If K denotes the matrix of the second-order moments under the mixture distribution, we have

$$K = \sum_{j=1}^m P_j K_j. \quad (13)$$

Then Eq. 11 becomes

$$KW = M, \quad (14)$$

where

$$K = \sum_{j=1}^m P_j E_j \{h\} V_j^T. \quad (15)$$

If K is nonsingular, the optimal W can be calculated by

$$W^* = K^{-1}M. \quad (16)$$

When compared to the traditional method, the size of K , $k \times k$, is much smaller than the Φ matrix of size $(n \times k)$ described in the previous subsection. Therefore, the improved method requires less memory space for storing the matrix, as well as consumes much less

memory. The value of n in some classification problems is very large, such that it may be impractical to have such large amounts of memory space for storage. Also, the process of calculating $(\Phi^T \Phi)^{-1}$ for large Φ is computationally expensive. In addition, this method needs a lot of computations for matrix multiplication and inversion. Therefore, this method may not be suitable for the use of classification problem.

B. Improved Least Mean Square Error Method

The improved least mean square error method for data classification was proposed by Devijver et. al.[24] and has been employed by Hwang et. al. in [12]. This method aims to calculate w_j for m classes at the same time. We detail the procedures as follows.

For a classification problem with m classes, let V_i designate the i -th column vector of an $m \times m$ identity matrix and W be an $k \times m$ matrix of weights:

$$\omega = [\omega_1 \ \omega_2 \ \dots \ \omega_m]$$

Then the objective function to be minimized is

$$J(W) = \sum_{j=1}^m P_j E_j \{ \| W^T h - V_j \|^2 \}, (10)$$

where P_j and $E_j \{ \}$ are the a priori probability and the expected value of class- j , respectively.

computation time for matrix multiplication. It is apparent that the improved method is more efficient than the traditional one.

However, there is a critical drawback of the improved method. That is, K may be singular and this will crash the whole procedure. By observing the matrix hh^T , we are aware of that the matrix hh^T is symmetric positive semi-definite (PSD) matrix with rank = 1. Since K is the summation of hh^T for each training instance, K is also a PSD matrix with rank $\leq n$. When $k \rightarrow n$, it is highly possible to have K be singular. From our experiences, if all the training instances are chosen as centers, this method is not going to work eventually. Thus, we solved this problem in the following subsection.

C. Proposed Least Mean Square Method

A very simple solution to solve the singular problem has been shown in the context of regularization theory [25]. It consists in replacing the the objective function as follows:

$$J(W) = \sum_{j=1}^m P_j E_j \{ \| W^T h - V_j \|^2 \} + \lambda \sum_{j=1}^m \omega_j^T \omega_j, (17)$$

where λ is the regularization parameter. Then the Eq. 14 becomes

$$(K + \lambda I)W = M. (18)$$

TABLE I
THE BENCHMARK DATA SETS USED IN THE EXPERIMENTS

	# of training samples	# of testing samples	If we set $\lambda > 0$, $(K + \lambda I)$ will be a positive definite (PD) matrix and therefore is nonsingular. The optimal W^* can be calculated by $W^* = (K + \lambda I)^{-1}M$. (19)
satimage	4435	2000	
letter	15000	5000	
shuttle	43500	14500	
iris	150	N/A	
wine	178	N/A	
vowel	528	N/A	
segment	2310	N/A	
glass	214	N/A	
vehicle	846	N/A	

Finally, we can get the optimal W_j^* for class- j from W^* , and then the optimal discriminant function $c_j(x)$ for class- j is derived. By using the regularization theory, the optimal weights can be obtained analytically and efficiently.

V. EXPERIMENTS IN THE PROBLEM OF DATA CLASSIFICATION

The experiments in this section are conducted to evaluate the performance of the proposed RBFN classifier against other famous classifiers, the KDE based classifier [8], SVM [16], and KNN. Also, the incremental hierarchical clustering algorithm is compared with the APC-III clustering algorithm employed in [12]. Our proposed RBFN classifier and the APC-III based classifier share the same procedures of determining bandwidths and weights in constructing the RBFN. The discussions of the experiments will focus on the following two issues: classification accuracy and execution efficiency.

Table I lists main characteristics of the nine benchmark data sets used in the experiments. All these data sets are from the UCI repository [17]. Among the nine data sets, three of them are considered as the larger ones, as each contains more than 5000 samples with separate training and testing subsets. The remaining six data sets are considered as the smaller ones and there are no separate training and testing subsets in these six smaller data sets. Accordingly, different evaluation practices have been employed for the smaller data sets and for the larger data sets. For the three larger data sets, 10-fold cross validation has been conducted on the training set to determine the optimal parameter values

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY WITH THE THREE LARGER

	KDE	SVM	1NN	3NN	APC-III	Proposedd
satimage	92.30	91.30	88.80	90.65	90.25	92.00
letter	97.12	97.98	95.68	95.16	91.16	97.48
shuttle	99.94	99.92	99.94	99.91	97.34	99.82
Average	96.45	96.40	94.84	95.24	92.92	96.43

DATA SETS
TABLE III
COMPARISON OF CLASSIFICATION ACCURACY WITH THE SIX SMALLER

	KDE	SVM	1NN	3NN	APC-III	Proposed
iris	97.33	97.33	96.00	95.33	95.33	96.00
wine	99.44	99.44	95.52	96.07	98.89	97.78
vowel	99.62	99.05	99.62	97.35	93.37	98.48
segment	97.27	97.40	97.27	96.14	94.98	97.53
glass	75.74	71.50	72.01	92.01	69.16	72.86
vehicle	73.53	86.64	69.73	71.39	78.25	79.19
Average	90.49	91.89	88.36	88.05	88.33	90.31

the size of training data set is larger than 20000. In regard to the parameter settings of other classifiers for comparison, we adopted the parameter settings suggested by the authors in their original papers.

Table II compares the accuracy delivered by alternative classification algorithms with the three larger benchmark data sets. As Table II shows, the proposed method basically deliver the same level of accuracy with other famous classifiers, SVM and KDE, while the KNN and APC-III based classifier do not produce comparable generation results. Table III lists the experimental results with the six smaller data sets. Table III shows that the proposed method basically deliver the same level of accuracy for these six data sets. The experimental results presented in Table III also show that the proposed method, KDE based classifier and the SVM generally deliver a higher level of accuracy than the KNN and APC-III based classifier.

Table IV compares the execution time of the KDE based classifier, the SVM, the APC-III based classifier and the proposed method with the three larger data sets presented in Table I. In Table IV, the total

to be used in the testing phase. On the other hand, for the six smaller data sets, 10-fold cross validation has been conducted on the entire data set and the average result is reported.

Our incremental algorithm has two key parameters, the size of initial data samples and the size of the tentative outlier buffer. In our experiments, both of the size of initial data instances and the size of tentative outlier buffer are set to 1000. We observed that these two buffers do not affect the quality of the classifier much but do influence the execution time. The larger the buffer size, the longer the reconstructing process. In the experiments, the incremental mechanism is turned on when

time taken to construct classifiers based on the given training data sets are listed in the rows marked by "Make classifier". The time listed in "Make classifier" row are the time of cross validation for KDE based classifier and the time of model selection for SVM. On the other hand, for both the APC-III based classifier and the proposed algorithm, the reported time include the time of clustering process and the time of calculating bandwidths and weights. In addition, the time taken by alternative classifiers to predict the classes of the testing instances are listed in the rows marked by "Prediction".

As we can see in Table IV, the mechanism proposed in this paper is much more efficient than the SVM and the KDE based classifier for constructing a data classifier. In addition, the mechanism proposed in this paper delivers comparable execution efficiency as the SVM in the prediction phase and

TABLE IV
COMPARISON OF EXECUTION TIME IN SECONDS

		KDE	SVM	APC-III	Proposed
Make Classifier	satimage	676	64644	136	274
	letter	2842	387096	712	5244
	shuttle	98540	467955	2595	380
Prediction Time	satimage	21.30	11.53	0.63	7.06
	letter	128.60	94.91	2.15	28.06
	shuttle	996.10	2.13	0.48	2.07

enjoys 30 times speedup over the KDE based classifier in this regard.

VI. CONCLUSION

In this paper we present an efficient method to construct an RBFN classifier whose performance was shown to be as good as the existing classification methods on the data sets used in this paper. Our contribution consists of two parts. First, we propose an incremental hierarchical clustering algorithm for constructing the hidden layer effectively and efficiently. Second, an improved least mean square error method that calculates the weights between the hidden and the output layers of an RBFN is introduced.

In the proposed clustering approach, the formation of clusters is controlled by the class labels of training samples and therefore the clusters identified are well adapted to the local distributions of training instances. In addition, it does not need to compute all the pairwise distances or similarity scores between training instances. Experimental results show that the data classifier constructed is capable of delivering comparable classification accuracy as the SVM and the kernel density estimation based classifier that we have recently proposed, while enjoying significant execution efficiency in handling data sets that contains a high percentage of redundant training instances.

Also, the proposed least mean square error method is efficient and with good theoretical foundations. The traditional least mean square method requires large memory to store the matrix and consumes a lot of execution time for the matrix multiplications and inversions. The improved method proposed by [12] is more efficient and practical than the traditional one, but it may suffer the singular matrix problem and fails to build the classifier in such case. In this paper, we solve the singular matrix problem by using the regularization theory, and this provides a good framework for constructing an RBFN in classification problems.

Experimental results also reveal that the approaches that have been proposed in recent years for solving the efficiency issues of the SVM and the kernel density estimation based mechanism all lead to slight degradation of classification accuracy. Thus, how to improve the efficiency of learning algorithms without sacrificing classification accuracy still deserves further studies.

REFERENCES

- [1] J. Park and I. W. Sandberg, "Universal approximation using radial-basisfunction networks," *Neural Computation*, vol. 3, no. 2, pp. 246–257, 1991.

- [2] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," *Tech. Rep. A.I. Memo 1140*, Massachusetts Institute of Technology, Artificial Intelligence Laboratory and Center for Biological Information Processing, Whitaker College, Jul 1989.
- [3] J. Ghosh and A. Nag, "An overview of radial basis function networks," *Radial Basis Function Neural Network Theory and Applications*, R. J. Howlerr and L. C. Jain (Eds), 2000.
- [4] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.
- [5] M. J. L. Orr, "Introduction to radial basis function networks," *tech. rep.*, Center for Cognitive Science, University of Edinburgh, UK, 1996.
- [6] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. The MIT Press, 2001.
- [7] C. M. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Computation*, vol. 3, no. 4, pp. 579–588, 1991.
- [8] Y.-J. Oyang, S.-C. Hwang, Y.-Y. Ou, C.-Y. Chen, and Z.-W. Chen, "Data classification with radial basis function networks based on a novel kernel density estimation algorithm," *IEEE Transactions on Neural Networks*, pp. 225 – 236, 2005.
- [9] D. G. Lowe, "Similarity metric learning for a variable-kernel classifier," *Neural Computation*, vol. 7, pp. 72–85, 1995.
- [10] A. Lyhyaoui, M. Martinez, I. Mora, M. Vazquez, J.-L. Sancho, and A. R. Figueiras-Vidal, "Sample selection via clustering to construct support vector-like classifiers," *IEEE Transactions on Neural Networks*, vol. 10, p. 1474, Nov 1999.
- [11] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, pp. 302–309, Mar. 1991.
- [12] Y. Hwang and S. Bang, "An efficient method to construct a radial basis function neural network classifier," *Neural Networks*, vol. 10, no. 8, pp. 1495–1503, 1997.
- [13] M. J. L. Orr, "Regularisation in the selection of radial basis function centres," 1995.
- [14] E. I. Chang and R. P. Lippmann, "A boundary hunting radial basis function classifier which allocates centers constructively," in *Advances in Neural Information Processing Systems*, vol. 5, pp. 131–138, Morgan Kaufmann, San Mateo, CA, 1993.
- [15] I. H. Witten and E. Frank, *Data mining*. Los Altos, US: Morgan Kaufmann, 2000.
- [16] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multi-class support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.
- [17] C. L. Blake and C. J. Merz, "UCI repository of machine learning databases," *tech. rep.*, University of California, Department of Information and Computer Science, Irvine, CA, 1998. Available at <http://www.ics.uci.edu/~mlern/MLRepository.html>.
- [18] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, pp. 264–323, Sept. 1999.
- [19] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice Hall International, 1988.
- [20] C.-Y. Chen, S.-C. Hwang, and Y.-J. Oyang, "An incremental hierarchical data clustering algorithm based on gravity theory," in *Proc. of PAKDD- 2002*, pp. 237–250, 2002.
- [21] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, vol. 1, no. 2, pp. 281–294, 1989.
- [22] D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks," *Complex Systems*, vol. 2, pp. 321–355, 1988.
- [23] I. Tarassenko and S. Roberts, "Supervised and unsupervised learning in radial basis function classifiers," in *IEE Proceedings-Vision, Image and Signal Processing*, vol. 141, pp. 210–216, 1994.
- [24] P. A. Devijver and J. Kittler, *Pattern recognition : a statistical approach*. Prentice Hall, 1982.
- [25] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. Washington D.C.: V.H. Winston & Sons, John Wiley & Sons, 1977.

計畫成果自評：本計畫已達成原計劃內容所規劃之進度，將所開發漸進式階層分群演算法應用於建構網路中的隱藏層，並提升用於計算網路中隱藏層與輸出層之間權重的最小平方錯誤方法之品質。此漸進式的架構大大地減少處理大量資料時記憶體空間的需求。實驗結果顯示我們所建構的分類器可以提供與支持向量機器分類器(SVM)或是我們最近提出的以核心密度推估方法為基礎的分類器一樣好的分類準確度，且同時提供高效能於處理那些有高度重複特性的資料集。執行期間已將階段性結果發表於國際會議。

行政院國家科學委員會補助國內專家學者出席國際學術會議報告

93 年 9 月 10 日

附件

報告人姓名	陳倩瑜	服務機構 及職稱	元智大學生物科技暨生物資訊研究所助理教授
時間 會議 地點	93 年 8 月 17 日至 93 年 8 月 19 日 美國 Stanford University	本會核定 補助文號	
會議 名稱	(中文) 2004 IEEE 計算系統生物資訊國際會議 (英文) 2004 IEEE Computational Systems Bioinformatics Conference		

報告內容：

一、參加會議經過

計算系統生物資訊國際會議在美國 Stanford 大學舉辦，自 2004 年 8 月 17 日起至 19 日止，為期三天，該會議由 IEEE Computer Society 主辦，每年舉辦一次，今年是第三次舉辦。由於意識到計算生物(Computational Biology)及生物資訊(Bioinformatics)的重要性，這三年的會議都受到全世界各學術與研究單位高度的注視，也因次會議的水準相當高。

此次會議，分別規劃了 4 場 Keynote Speech，7 場 Invited Speech，並有約 30 篇論文演講發表及 4 場 Poster Session，發表論文的國家總共包括五大洲 16 個國家。有來自世界各地學者、學生與廠商與會。

此次會議所涵蓋的層面非常廣，舉凡 whole genome analysis、gene expression analysis、protein motif analysis、pattern discovery、sequence search and alignment、protein family classification、protein structure and function prediction、molecular evolution and phylogeny、functional genomics 及 molecular biology databases and data mining 等主題都是本次會議之重點。

論文演講發表共分為 13 個議程，分別為「Structural Bioinformatics」、「Genomics」、「Transcriptomes」、「Evolution and Phylogeny」、「Proteomics」、「Applied Bioinformatics」、「Data Mining and Ontology」、「Fractals and Bioinformatics」、「Data Base and Ontology」、「Pathways and Networks」、「Protein motif analysis and pattern discovery」、「Fuzzy computing in biomedical applications」、「Bioinspired systems」、「Advances in biocomputing」及「Gene expression analysis」。這些議程的內容或從實務面、或從理論架構，都與「計算生物及生物資訊」的主題相關。

二、與會心得

在基因體學相關研究備受重視之此時，由於其相關資訊的不斷的累積，基因資料庫之建置，及新的分析方法不斷的推出，學者專家、研究機構、藥廠及生物技術公司皆湧入基因組織研究中。基因組研究毫無疑問將成為未來第二波工業革命—生物技術的基石。意識到此一研究之重要性，資訊工程者亦須責旁貸的參與此一重要的研究計畫。資訊工程者須研發先進的計算機工具及技能以協助生物學家辨識基因之基礎特徵，進而瞭解其結構及功能；有系統化的應用計算機系統提供新的有效方式以協助生物學家觀測生物的演化過程及更精確的描述生物系統；同時需能提供有效之儲存方法存放大量的基因資訊，及有效之檢索方法以綜覽基因資訊進而分析之。「生物資訊學」由此而生。

本次大會即涵蓋了這些主題，對於如何分析及處理分子層次之生物資訊，及以計算機、數學及統計模式分析分子生物現象，有相當多的主旨演講及論文發表。同時對於技術層面之技巧亦有著墨，例如資料結構之設計、機器學習、演化計算、模糊邏輯、類神經網路、訊息學及圖形識別。所以此次會議提對資訊工程與生物科學共同合作研究開發提供了一個良好之基石及遠景。

人類基因組之研究從「結構基因學」，到現在熱門的「功能基因學」，以至於未來的「演化基因學」，都讓我們對「生命」產生不同的認識，這是二十世紀科學上最重要的里程碑。資訊處理、數學模式、人工智慧、圖形識別、系統分析等都將成為未來生命科學研究之主流，缺乏這方面之認知，我們就無法培養出能因應未來這種研究趨勢的科學家。台灣的科學家已在全球的人類基因解讀計畫中缺席，我們應積極致力於培養未來生命科學家，同時應掌握「國際合作」及「跨領域合作」之趨勢，摒棄門戶之見，大家共同合作，並主動參與相關國際性事務，將有助於我國生物資訊科技產業之精進。

三、建議

各國已投資相當多的經費於生物資訊的研發，我國則尚在起步階段，相關產業界在此領域的投入相當有限，如何促進產業的投入並加強學術合作或產學合作，如何整合國內各研究於生物資訊之研究都是值得進一步討論的課題。

另外這是本人任教後第一次出國參加國際會議，令人印象深刻的是，國外的研究學者與廠商都非常熱心的參與會議之研討，積極發問並全程參與。尤其是在生物資訊已有盛名之學者都還全程參與至最後一場演講，其好學精神令人相當佩服。同時演講者提出頗多深具創意的論點，讓本人深覺此行收穫良多。本人深覺此行收穫良多，對於國科會贊助此行經費，深感謝意，並希望國科會能嘉惠更多的研究學者，尤其是莘莘學子。

四、攜回資料

1. 研討會論文集
2. 大會議程手冊