2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

A New Supervised Clustering Algorithm Based on Min-Max Modular Network with Gaussian-Zero-Crossing Functions

Jing Li and Bao-Liang Lu
Department of Computer Science and Engineering, Shanghai Jiao Tong University
800 Dong Chuan Rd., Shanghai 200240, China
Email: {jinglee, bllu}@sjtu.edu.cn

Abstract— **In this paper, we show that the shape, size and location of the receptive field around each instance are different and decided by the distribution of training data in a min-max modular network with Gaussian-zero-crossing functions. Based on this property, we propose a new supervised clustering algorithm which has the following features: First, the incremental clustering ability, which means the number of clusters need not to be predefined, it can grow up automatically, also, the training data need not to be processed iteratively; Second, attaching more importance to border instances than non-border instances, which guarantees the good generalization performance and training data reduction ratio; Third, outlier removal ability, which removes noise instances from training data; Last, cluster combination ability, which reduces the number of clusters further. Experiments on an artificial problem and several real-world applications demonstrate these attractive features of our new clustering algorithm.**

I. Introduction

Cluster analysis aims to organize a collection of data items into clusters, such that items within a cluster are more 'similar' to each other than they are to items in the other clusters [3]. If there is no information available concerning the membership of data items to predefined classes, the corresponding cluster analysis, such as the well-known k-means [16] or self-organizing map (SOM) [5], is called unsupervised clustering. But in many cases a small amount of knowledge is available concerning either pairwise constraints between data items or class labels for some items. This information can be used to 'guide' or 'adjust' the clustering process. The corresponding approach is called semi-supervised clustering [3], such as semi-supervised clustering by seeding [1]. If the class labels of all the data item are available, the cluster analysis belongs to supervised clustering, such as leaning vector quantization (LVQ) [6] and correlation clustering [2]. Supervised clustering may be very useful as a preprocessing to reduce training data set for further classification, since in real-world applications the training data set are always too large for traditional classifiers.

In many real-world applications, training data often become available in small and separate batches at different times. A clustering algorithm must have the incremental learning ability to deal with this situation. But solving clustering problems through optimization is NP-complete in most cases [14], many clustering methods, such as k-means, SOM and LVQ, have to use heuristic algorithms. They need predefine the number of clusters and process all the training data set iteratively, which make the incremental learning ability unattainable.

A straightforward approach of incrementally clustering data points is to group a new instance into an existing cluster that is closest to it or make the new instance as a new cluster. However, there exists the problem of how to decide in which circumstance the new instance should be grouped into an existing cluster and in which circumstance it should be treat as a new cluster. Li *et al.* have proposed a supervised clustering algorithm called Clustering and Classification Algorithm-Supervised (CCAS) [7], [8] to solve this problem. CCAS divide input space into some grid cells, only instances in the same grid cell can be clustered. But the problem of choosing the number of grid cells is still unsolved.

On the other hand, Lu *et al.* have proposed min-max modular network with Gaussian-zero-crossing functions (M^3-GZC) [9], [10] which has locally tuned response characteristic and emergent incremental learning ability. In this paper, we analyze the properties of receptive field around each instance in M^3-GZC network. We discover that the shape, size and location of each receptive field are determined by the distribution of training data. The receptive field can be viewed as the 'grid cells' in input space and can be used as a criterion to decide whether the new instance should be grouped into an existing cluster or should be treat as a new cluster. Also, we find the receptive fields around border instances are smaller than that around non-border instances in M^3-GZC network, which means the corresponding clustering algorithm will generate more clusters near the border and less clusters far from the border. More clusters near the border guarantees that the distortion of border instances is smaller while less clusters far from the border guarantees the higher reduction ratio of training data. We do not discard the non-border instances for further reduction because in the incremental learning process the 'non-border' instances may become 'border' instances after more training data available, discarding them may lead to great and unwanted changes to the decision boundaries. Since not all the non-border instance will become border instance in future, we pay less attention to them than the instances that are already border instances.

The remainder of the paper is organized as follows. In Section II, M^3-GZC network is introduced briefly. In Section III, we analyze the properties of receptive field in M^3-GZC network. Based on these properties, our supervised clustering method is proposed in Section IV, post-processing including

0-7803-9490-9/06/$20.00/©2006 IEEE

outlier removal and cluster combination are also presented in this section. Experimental results on an artificial problem and several real-world applications are listed in Section V. Finally, conclusions are given in Section VI.

II. Min-Max Modular Network with GZC Function

A. Min-Max Modular Network

Let $\mathcal{T}$ be the training set for a K-class problem,

$$\mathcal{T} = \{(X_l, D_l)\}_{l=1}^{L} \tag{1}$$

where $X_l \in R^n$ is the input vector, $D_l \in R^K$ is the desired output, and L is the total number of training data.

According to the min-max modular network [11], [12], a K-class problem defined in equation (1) can be divided into $K \times (K-1)$ two-class problems that are trained independently. The decomposition process is described as following. First we divide the input vectors into K subsets according to class relations.

$$X_i = \left\{X_l^{(i)}\right\}_{l=1}^{L_i}, \quad \text{for } i = 1, 2, \ldots, K \tag{2}$$

where L_i is the number of data of X_i, all of $X_l^{(i)} \epsilon X_i$ have the same desired outputs, and $\sum_{i=1}^{K} L_i = L$. Then we combine X_i and X_j as the training set of a two-class problem $\mathcal{T}_{i,j}$,

$$\mathcal{T}_{ij} = \left\{\left(X_l^{(i)}, 1-e\right)\right\}_{l=1}^{L_i} \cup \left\{\left(X_l^{(j)}, e\right)\right\}_{l=1}^{L_j} \quad \text{for } i, j = 1, \ldots, K \text{ and } j \neq i \tag{3}$$

After these two-class problems are trained in parallel, they are integrated according to a module combination rule, namely the minimization principle.

$$\mathcal{T}_i(x) = \min_{j=1}^{K} \mathcal{T}_{ij}(x) \tag{4}$$

where $\mathcal{T}_{ij}(x)$ denotes the transfer function of the trained network corresponding to the two-class subproblem $\mathcal{T}_{ij}$, and $\mathcal{T}_i(x)$ denotes the transfer function of distinguish class i from other classes.

If these two-class problems are still in large-scale or imbalanced, they can be further decomposed into relatively smaller two-class problems. Suppose the training set X_i defined in equation (2) is partitioned into N_i ($1 \leq N_i \leq L_i$) subsets in the form

$$X_{ij} = \{X_l^{(ij)}\}_{l=1}^{L_i^{(j)}}, \quad \text{for } j = 1, \ldots, N_i \tag{5}$$

where $L_i^{(j)}$ is the number of data of X_{ij}, and $\cup_{j=1}^{N_i} X_{ij} = X_i$. The training set of each smaller two-class problem can be given by

$$\mathcal{T}_{ij}^{(u,v)} = \left\{\left(X_l^{(iu)}, 1-e\right)\right\}_{l=1}^{L_i^{(u)}} \cup \left\{\left(X_l^{(jv)}, e\right)\right\}_{l=1}^{L_j^{(v)}} \quad \text{for } u = 1, \ldots, N_i,\ u = 1, \ldots, N_j,\ i, j = 1, \ldots, K \text{ and } j \neq i \tag{6}$$

where $X_l^{(iu)} \in X_{iu}$ and $X_l^{(jv)} \in X_{jv}$ are the input vectors belonging to class C_i and C_j, respectively.

After these smaller two-class problems $\mathcal{T}_{ij}^{(u,v)}$ have been trained, they will be integrated according to the minimization principle and maximization principle, respectively, as follows:

$$\mathcal{T}_{ij}^{(u)}(x) = \min_{v=1}^{N_j} \mathcal{T}_{ij}^{(u,v)}(x) \tag{7}$$

$$\mathcal{T}_{ij}(x) = \max_{u=1}^{N_i} \mathcal{T}_{ij}^{(u)}(x) \tag{8}$$

B. M^3-GZC Network

Suppose the training set of each subproblem has only two different instances, then they can be separated by a Gaussian zero-crossing discriminate function [9] defined by

$$f_{ij}(x) = exp\left[-\left(\frac{\|x - c_i\|}{\sigma}\right)^2\right] - exp\left[-\left(\frac{\|x - c_j\|}{\sigma}\right)^2\right] \tag{9}$$

where x is the input vector, c_i and c_j are the given training inputs belonging to class C_i and class C_j ($i \neq j$), respectively, $\sigma = \lambda\|c_i - c_j\|$, and λ is a user-defined constant.

The output of M^3-GZC network is defined as follows:

$$g_i(x) = \begin{cases} 1 & \text{if } y_i(x) > \theta_i \\ Unknown & \text{if } -\theta_j \leq y_i(x) \leq \theta_i \\ -1 & \text{if } y_i(x) < -\theta_j \end{cases} \tag{10}$$

where θ_i and θ_j are the threshold limits of class C_i and C_j, respectively, and y_i denotes the transfer function of the M^3 network for class C_i, which discriminates the pattern of the M^3 network for class C_i from those of the rest of the classes.

From equations (9) and (10), we can see that the interpolation and extrapolation capabilities can be easily controlled by selecting different values of threshold limits, as shown in Fig.1. And in the next section, we will further analyze the relationship between threshold limits and decision boundaries.

III. Properties of Receptive Field in M^3-GZC Network

Definition 1) Receptive Field: the input space that can be classified to one class in a M^3-GZC network.

$$RF = \{x | x \epsilon R^n, \exists i, g_i(x) = 1\} \tag{11}$$

Lemma 1: Suppose there are only two instances c_i and c_j, and we only concentrate on the receptive field around c_i. Then the relationship between the longest receptive field radius r_{max} and the distance between c_i and c_j can be expressed as

$$r_{max} = k_1\|c_i - c_j\| \tag{12}$$

where k_1 is only correlated with λ and θ_i.

Proof: According to the axiom of norm, the following equation is satisfied.

$$\|c_i - c_j\| - \|x - c_i\| \leq \|x - c_j\| \leq \|c_i - c_j\| + \|x - c_i\| \tag{13}$$

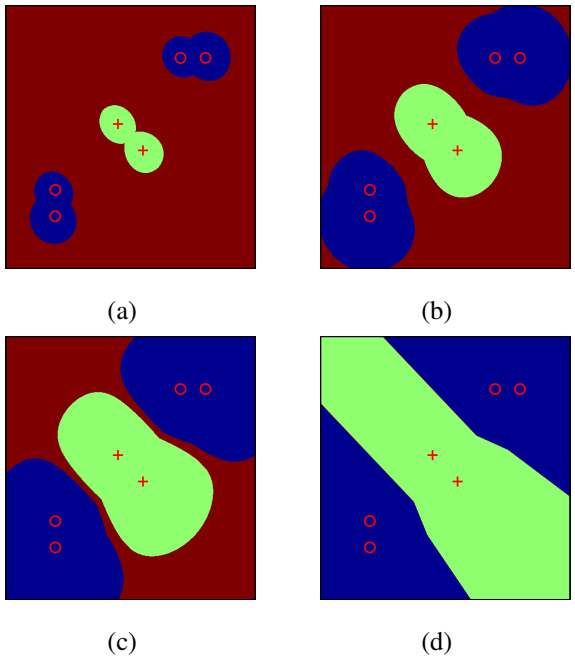


Fig. 1. Decision boundaries at different threshold limits. (a) $\theta_i = \theta_j = 0.8$; (b) $\theta_i = \theta_j = 0.4$; (b) $\theta_i = \theta_j = 0.1$; (d) $\theta_i = \theta_j = 0$; The red area denotes the '$Unknown$' decision regions.

So the longest receptive field radius r_{max} can be achieved when $\|x - c_j\| = \|c_i - c_j\| + \|x - c_i\|$. From equations (9) and (12), we get

$$\begin{aligned}\theta_i = & exp\left[-\left(\frac{k_1\|c_i - c_j\|}{\lambda\|c_i - c_j\|}\right)^2\right] \\ & - exp\left[-\left(\frac{k_1\|c_i - c_j\| + \|c_i - c_j\|}{\lambda\|c_i - c_j\|}\right)^2\right] \\ = & exp\left[-\left(\frac{k_1}{\lambda}\right)^2\right] - exp\left[-\left(\frac{k_1 + 1}{\lambda}\right)^2\right] \end{aligned} \tag{14}$$

which means k_1 is a function of λ and θ_i. This completes the proof.

Also, we can prove that the relationship between the shortest receptive field radius r_{min} and $\|c_i - c_j\|$ can be expressed as:

$$r_{min} = k_2\|c_i - c_j\| \tag{15}$$

where k_2 satisfies the following equation

$$\theta_i = exp\left[-\left(\frac{k_2}{\lambda}\right)^2\right] - exp\left[-\left(\frac{1 - k_2}{\lambda}\right)^2\right] \tag{16}$$

Theorem 1: Suppose instance c_j is the nearest instance to instance c_i, c_j and c_i belong to different classes, RF_i is the receptive field around c_i, then the radius r_i of RF_i is between r^i_{min} and r^i_{max}, where

$$\begin{aligned} r^i_{min} &= k_2\|c_i - c_j\| \\ r^i_{max} &= k_1\|c_i - c_j\| \end{aligned} \tag{17}$$

Proof: Suppose $RF_{i,j}$ is the receptive field around instance c_i based on c_i and c_j, according to *Lemma 1*, the radius r_j of $RF_{i,j}$ satisfies the following inequality

$$k_2\|c_i - c_j\| \le r_j \le k_1\|c_i - c_j\| \tag{18}$$

Suppose c_{k1}, c_{k2}, ... are other instances that belong to different classes from the class of instance c_i, $RF_{i,k1}$, $RF_{i,k2}$, ... are the receptive fields around instance c_i based on c_i and c_{k1}, c_{k2}, ..., r_{k1}, r_{k2}, ... are the radius of $RF_{i,k1}$, $RF_{i,k2}$, ..., respectively. According to *Lemma 1*, r_{k1}, r_{k2}, ... satisfy the following inequalities

$$\begin{aligned} k_2\|c_i - c_{k1}\| \le r_{k1} \le k_1\|c_i - c_{k1}\| \\ k_2\|c_i - c_{k2}\| \le r_{k2} \le k_1\|c_i - c_{k2}\| \\ \dots \end{aligned} \tag{19}$$

Since c_j is the nearest neighbor in different class to instance c_i, $\|c_i - c_{k1}\|$ and $\|c_i - c_{k2}\|$ satisfy the following inequalities

$$\begin{aligned} \|c_i - c_{k1}\| \ge \|c_i - c_j\| \\ \|c_i - c_{k2}\| \ge \|c_i - c_j\| \\ \dots \end{aligned} \tag{20}$$

Since the role of minimization principle is similar to the logical AND [11], the final receptive field RF_i around c_i satisfies the following equation

$$RF_i = RF_{i,j} \cap RF_{i,k1} \cap RF_{i,k2} \cap \dots \tag{21}$$

So

$$\begin{aligned} r_i \ge & min\left(k_2\|c_i - c_j\|, k_2\|c_i - c_{k1}\|, k_2\|c_i - c_{k2}\|, \dots\right) \\ = & k_2\|c_i - c_j\| \end{aligned} \tag{22}$$

and

$$\begin{aligned} r_i \le & min\left(k_1\|c_i - c_j\|, k_1\|c_i - c_{k1}\|, k_1\|c_i - c_{k2}\|, \dots\right) \\ = & k_1\|c_i - c_j\| \end{aligned} \tag{23}$$

This completes the proof.

From the analysis above, we can conclude that the receptive field around every instance has the following attribute:

1) The size, shape and location of receptive field around each instance are only correlated with λ, θ and the neighbors in different classes around this instance. The receptive field is local and its size is mainly determined by the nearest neighbor in different classes around it (as depicted in Fig.1).
2) Receptive fields around border instances are smaller than that around non-border instances since the distance between border instances are smaller than that between non-border instances (as depicted in Fig.1).
3) The radius of receptive field is decreased with the increase of threshold limits (as depicted in Fig.1).

IV. Clustering Algorithm Based on M^3-GZC Network

A. C-M^3-GZC

Since the receptive field of every instance is the local area around it, we can view a new instances that locates in the area belongs to the same cluster as this instance. Since the

receptive field of every instance may be overlapped, we only consider the nearest instance to the new instances. When a new instance (x, d) is available, we can find its nearest neighbor (x', d) in the same class as (x, d). If the output of the MIN unit around (x', d) is 1 (we say that (x, d) is accepted by the MIN unit around (x', d)), the final output of the M^3-GZC network will be d, which means that (x, d) locates in the receptive field around (x', d). So we treat (x, d) and (x', d) belong to a same cluster. If the output of the MIN unit around (x', d) is $'Unknown'$ or -1, the final output of the M^3-GZC network will be $'Unknown'$ or incorrect, we treat (x, d) as a new cluster center.

Also, since the radius of receptive field is decreased with the increase of threshold limits, if a test sample is accepted by a M^3-GZC network with high values of threshold limits, it will be accepted by the same network with lower values of threshold limits. The threshold limits can be viewed as a degree of confidence of correct classification. Since the confidence of correct classification will become higher and higher with more and more instances available, we can adjust the threshold limits during clustering process. At the beginning of clustering, there are only few instances available, they distribute sparsely and it is difficult to classify new instance correctly according to them. The distance between each instance may be very large, and the receptive field of each instance may be large, too. We can set a large value of θ to decrease the size of receptive field. With more and more instances available, the confidence of correct classification becomes higher and we can set a smaller value of θ. The time varying threshold limits can guarantee a more robust incremental clustering algorithm.

According to the above analysis, our clustering algorithm based on M^3-GZC network (C-M^3-GZC) is presented in Algorithm 1, where S is the clustering set, and it can start from scratch or have some members trained before. The items of S can be expressed as (x, d, n), where x is the centroid coordinates of the cluster, d is the class ID of this cluster, and n is the total number of instances in this cluster.

A simple illustration of the proposed clustering process is depicted in Fig.2. Here the time varying function of threshold limits is defined by

$$\theta = (\theta_{max} - \theta_{min}) exp\left[-\frac{t^2}{2\sigma^2}\right] + \theta_{min} \qquad (24)$$

We chose $\theta_{max} = 1.0$, $\theta_{min} = 0.2$, $\sigma = 1/2\sqrt{2}$, and $t = i/n$, where i indicates the ith instances that presented to the M^3-GZC network, and n indicates the total number of training data.

Fig.2 (a) is the distribution of training data that belong to two classes labelled by plus and circle, respectively. Fig.2 (b) shows the decision boundaries of M^3-GZC network after two instances have been presented. The third instance labelled by triangle locates in the '$Unknown$' area, so it will be treated as a new cluster center. But the forth instance locates in the receptive field of the third instance, and they belong to the

Algorithm 1 C-M^3-GZC

Input:
Training set: S_{new}
Previously trained cluster set: S
Parameter of M^3-GZC network: λ
Time varying function of threshold limits: $f(t)$
Output:
New cluster set: S

```
for each data (x, d) in S_new do
    Find its nearest neighbor (x', d', n) in S;
    if d = d' and (x, d) can be accepted by the MIN unit
    based on (x', d') then
        Update the centroid coordinates of (x', d', n): x' =
        (nx' + x)/(n + 1);
        Update the number of points in this cluster: n=n+1;
    else
        Treat (x, d) as a new cluster center: S = S∪(x, d, 1);
    end if
    Update the threshold limits: θ = f(t);
end for
```

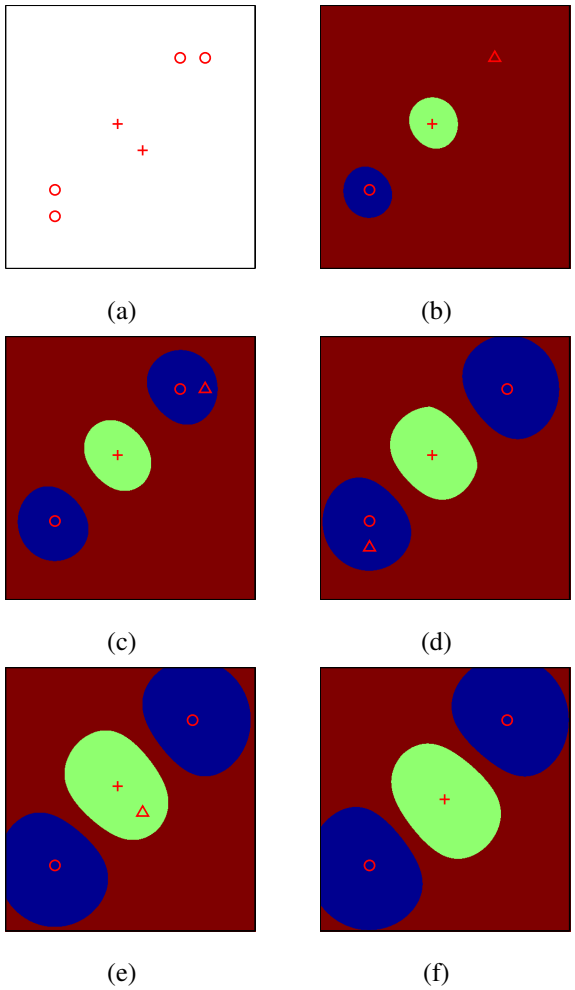


Fig. 2. A simple illustration of the proposed clustering process. The red area denotes the '$Unknown$' decision regions. (a) Original training data; (b)-(e) Decision boundaries of M^3-GZC network after more and more instances are learned, the triangles denote the instances waiting to be processed; (f) The final clusters.

same class, so they will be clustered as shown in Fig.2 (d). Fig.2 (f) shows the final results of three clusters as we have expected.

B. Post-Processing

In this subsection, we propose an outlier removal algorithm and a cluster combination algorithm for post-processing of C-M^3-GZC.

A cluster that has few instances may locate on border or be noise instances. If it is a border instance, it will have some neighbor clusters that belong to the same class as it. But if it a noise instance, its neighbor clusters will belong to different classes. So our noise removal or 'outlier removal' algorithm works as Algorithm 2.

Algorithm 2 Outlier Removal

Input:
Cluster set: S
Limit of instance number: n_1
Number of neighbor clusters to be considered: n_2
Output:
New cluster set: S

```
for each cluster (x, d, n) in S do
  if n ≤ n_1 then
    find its n_2 nearest neighbor clusters in S;
    if all the n_2 neighbors belong to different classes of
    (x, d, n) then
      Remove (x, d, n) form S: S = S\(x, d, n);
    end if
  end if
end for
```

After the outlier removal, some clusters that are divided by noise can be combined. Also, since the threshold limits are decreasing with time, some clusters that are divided under high threshold limits can be combined under low threshold limits. And with more training data available, some clusters that are divided under few training data can be combined under more training data. So we need a cluster combination algorithm to reduce cluster number for further compression.

If a cluster can be accepted by its nearest neighbor cluster, we treat the two clusters are similar and combine them into one cluster. Based on this idea, our cluster combination algorithm works as Algorithm 3. The cluster combination process can be implemented only once or repeated for many times, until a certain cluster number is reached.

C. Complexity Analysis

Let M be the final number of clusters, and N be the number of training data. In the clustering process, the time complexity of finding the nearest neighbor for one training instance is $O(M)$, deciding whether it is accepted by the corresponding MIN unit is $O(M)$. So the total time complexity is $O(MN)$. In the outlier removal process, the time complexity of finding the nearest neighbor for one cluster is $O(M)$. The total time complexity is $O(M^2)$. In the

Algorithm 3 Cluster Combination

Input:
Cluster set: S
Parameter of M^3-GZC network: λ
Threshold limits of M^3-GZC network: θ_{com}
Output:
New cluster set: S

```
for each cluster (x, d, n) in S do
  Find its nearest neighbor (x', d', n') in S\(x, d, n);
  if d = d' and (x, d) can be accepted by the MIN unit
  based on (x', d') then
    Update the centroid coordinates of cluster (x', d', n'):
    x' = (n'x' + nx)/(n' + n);
    Update the number of points in this cluster: n' =
    n' + n;
    Remove (x, d, n) from S: S = S\(x, d, n).
  end if
end for
```

cluster combination process, the time complexity of finding the nearest neighbor for one training instance is $O(M)$, deciding whether it is accepted by the corresponding MIN unit is $O(M)$. The total time complexity is $O(M^2)$. If we use some searching techniques (such as k-dimensional trees [15]) the time complexity of finding the nearest neighbor for one instance can be reduced to $O(logM)$. The total time complexity of C-M^3-GZC, outlier removal and cluster combination algorithms can be reduced to $O(NlogM)$, $O(MlogM)$ and $O(MlogM)$, respectively.

V. Experimental Results

In order to verify our method, we present three experiments. The first is an artificial problem and the other two are real-world problems. All the experiments were performed on a 2.8GHz Pentium 4 PC with 1GB RAM. The time varying function of threshold limits we used in theses experiments is the same as the simple illustration we presented in Fig.2.

A. Checkerboard Problem

A checkerboard problem is depicted in Fig.3 (a). The checkerboard divides a square into four quadrants. The points labelled by dot and plus are positive and negative instances, respectively. In this experiment, we randomly generate 1000 instances as shown in Fig.3 (a). We use the C-M^3-GZC algorithm to find clusters at different threshold limits on this data set. Since there is no noise in the training data, we just use the cluster combination process as the post processing. The cluster centers at different threshold limits are shown in Figs.3 (b), and (c). Results of LVQ are also shown in Fig.3 (d) for comparison.

We also randomly generate 10000 instances as training data set, and another 10000 instances as test data set. We use the C-M^3-GZC algorithm and the cluster combination process on the training data set, and then classify the test instances to the same class of their nearest clusters. Two ways of using cluster combination are listed in Table I. One

TABLE I

RESULTS AT DIFFERENT THRESHOLD LIMITS ON CHECKERBOARD PROBLEM.

θ_{min}	C1-M^3-GZC			LVQ		C2-M^3-GZC			LVQ	
	Number	Accuracy	Time	Accuracy	Time	Number	Accuracy	Time	Accuracy	Time
0.00	112	96.66%	5.3	95.37%	25037	4	99.43%	5.3	92.70%	29411
0.01	324	99.55%	5.4	99.24%	24492	309	99.55%	5.4	99.22%	32181
0.1	470	99.64%	6.3	99.24%	37676	460	99.67%	6.3	99.12%	37049
0.2	591	99.74%	5.9	99.31%	45995	576	99.72%	6.0	99.28%	41956
0.3	684	99.76%	6.5	99.15%	46322	672	99.76%	6.6	99.07%	49082
0.4	799	99.76%	7.2	99.30%	52045	786	99.76%	7.3	99.23%	51397
0.5	912	99.76%	8.0	99.20%	53956	902	99.76%	8.1	99.22%	53173

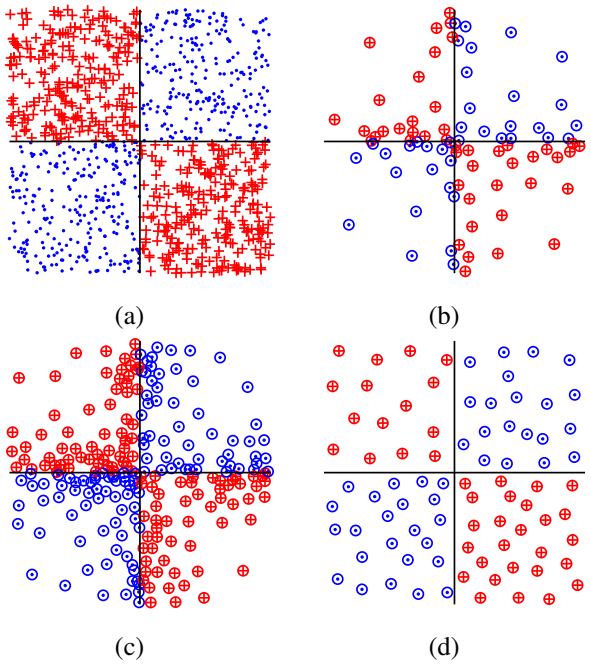


Fig. 3. A checkerboard problem and its clustering result at different threshold limits. The points labelled by dot and plus are positive and negative instances, respectively. The points labelled by dot with circle and plus with circle are cluster centers. (a) A checkerboard problem; (b) $\theta_{min} = \theta_{com} = 0.01$, $\theta_{max} = 1.0$; (c) $\theta_{min} = \theta_{com} = 0.5$, $\theta_{max} = 1.0$; (d) Results of LVQ with same cluster number as (b).

is using cluster combination only once, the other is repeating it until the cluster number is stable. They are indicated as C1-M^3-GZC and C2-M^3-GZC in Table I, respectively. The results of using LVQ are also listed for comparison. Since the cluster number of LVQ is defined by user, we do the experiments of each data set twice, which have the same cluster number as that of C1-M^3-GZC and the that of C2-M^3-GZC, respectively.

From Fig.3 and Table I, we can draw the following conclusions.

1) The number of training data can be greatly reduced by C-M^3-GZC, and it is determined by the threshold limits. The higher the threshold limits are, the more clusters are generated. We also find that if we set $\theta_{min} = 0$ and use the cluster combination process until a stable cluster number is reached, the final cluster number is 4, which is just the same as the minimum cluster number as we have expected;
2) Using higher threshold limits can obtain higher classification accuracy;
3) As shown in Fig.3 (b) and (c), clusters near the border are distributed more densely than that far away from border. That is because the receptive fields around border instances are smaller than that around non-border instances. More clusters around border can obtain a better generalization performance, and fewer clusters around non-border can obtain a higher data reduction rate. We do not discard the non-border instances for further reduction because in the incremental learning process the 'non-border' instances may become 'border' instances after more training data available, discarding them may lead to great and unwanted changes to the decision boundaries. Since not all the non-border instance will become border instance in future, we pay less attention to them than the instances that are already border instances. Attaching more importance to border instances is a balance between data reduction rate and generalization performance. Compared with Fig.3 (d), we can see that clusters distribute evenly in LVQ. And if the cluster numbers of LVQ and C-M^3-GZC are same, the generalization performance of the latter is better than the former, as shown in Table I.
4) Compared with LVQ, our algorithm spend less time, only 0.016% of that of LVQ on average. That is because LVQ will process all the training data set iteratively to achieve a good result, while our method will only treat each input once.

To test our noise removal program, we randomly generate 10000 instances with different ratio of noise instances among them. The parameters of each experiments is $\theta_{min} = \theta_{com} = 0.2$, $\theta_{max} = 1.0$, $n_1 = 2$, and $n_2 = 2$. The result are shown in Fig.4. Compared with Table I, the number of clusters are larger at the same parameters, that is because the noise instances separate some clusters into smaller clusters.

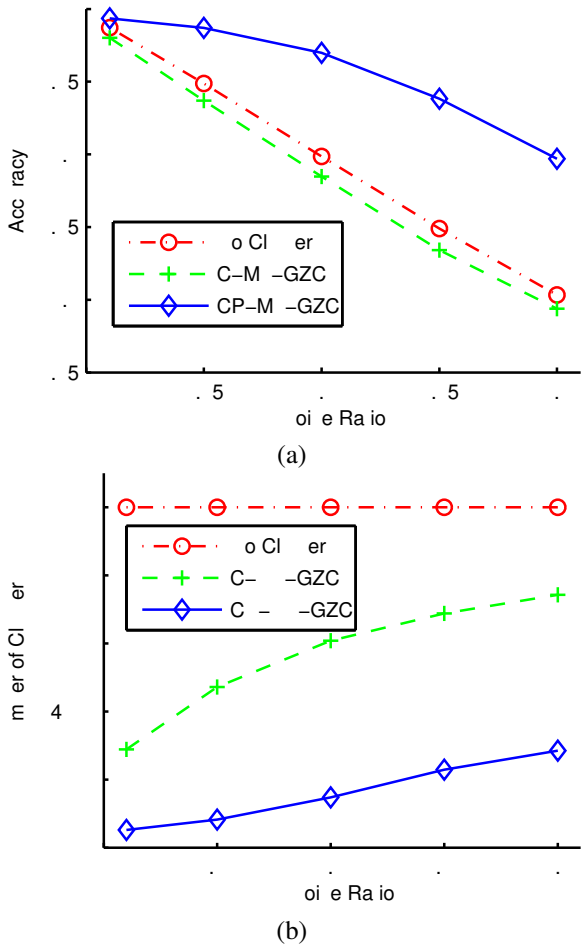
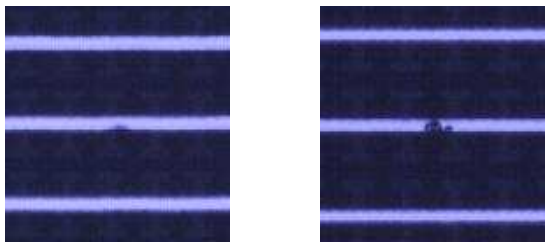


Fig. 4. Clustering result of checkerboard problem with noise instances. (a) Accuracy; (b)Number of clusters.

The results show that after noise removal programm, the classification accuracy can be improved greatly, and is better than the original training data set.

B. UCI Database

In this experiment, our algorithm is tested on three benchmark data sets from the Machine Learning Database Repository [13]. The parameters of each experiments are $\theta_{max} = 1.0$, $\theta_{min} = 0.2$, and $\theta_{com} = 0.2$. The results are listed in Table II, where C-M^3-GZC denotes our clustering method without post-processing and CP-M^3-GZC denotes our clustering method with post-processing. Experiments results of using LVQ with the same cluster number as that of C-M^3-GZC and CP-M^3-GZC are also listed for comparison. From Table II, we can see that the accuracy of our methods is comparative or better than original data set while maintaining small size of instances. The size ratio of each data set are different because each data set has different redundant training data. The accuracy of our method is also better than LVQ and the training time is much faster than that of LVQ.



(a) (b)

Fig. 5. Glass-board images in the industry image classification project. (a) A good glass-board; (b) A fault glass-board.

C. Industry Image Classification

We also use our clustering method on an industry image classification project [4]. The purpose of this project is to distinguish fault glass-boards from good glass-boards in an industrial product line, as shown in Fig.5. The ability of incremental clustering is urgently needed in this project since the glass-board images can be obtained everyday and vary with the changes of circumstances.

In our experiment, each glass-board image is converted into a 4096 dimension vector, and we divided the glass-board images into eight groups according to the time that they were collected. The number of images in each group is 1133, 1227, 1149, 1160, 1138, 1147, 1088, and 1197, respectively. We use the first to the seventh groups as the training set and the eighth group as the test set. At first, we use C-M^3-GZC to obtain the cluster C_1 on the first training set. Then we presented the second training set to C_1 and use C-M^3-GZC to obtain the cluster C_2. After all the training set have been presented, we use outlier removal and cluster combination algorithm on C_7 to get the final clusters. The classification accuracy and the number of clusters are shown in Fig.6. We also show the result of not using clustering method for comparison. Since LVQ has not the ability of incremental learning, experiments of using LVQ are not done for comparison.

From Fig.6, we can see that the size of training data can be greatly reduced while maintaining the generalization performance by using C-M^3-GZC algorithm. Also, after the post-processing, the noise can be removed, the generalization performance becomes better, and the number of clusters can be decreased further.

VI. Conclusions

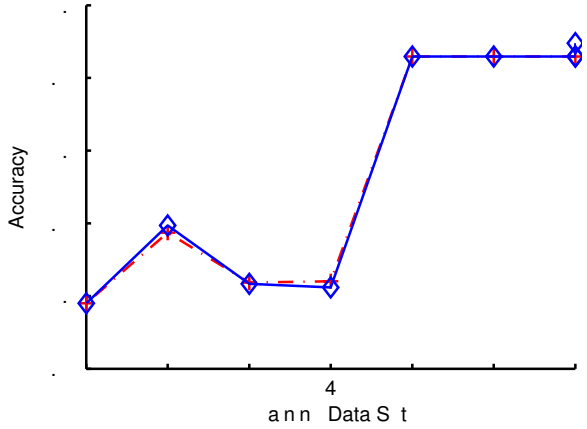
In this paper we have analyzed the properties of receptive field in M^3-GZC network. Based on these properties, we propose a new supervised clustering algorithm. It has the following attractive features.

- The incremental cluster ability. Unlike traditional clustering method, it need not retreat training instances. The number of cluster need not be predefined, it grows automatically and is determined by the distribution of training data.

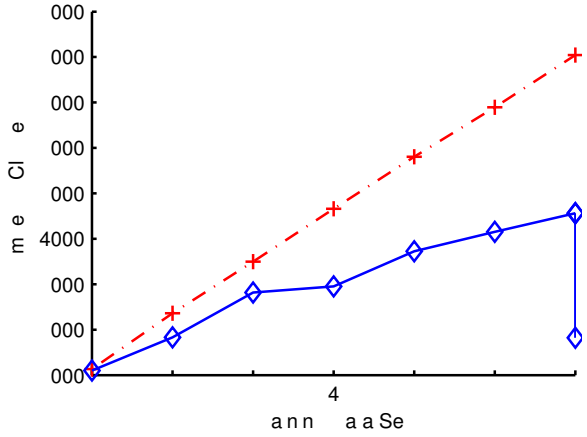
TABLE II

RESULTS ON UCI DATABASE. PARAMETERS OF EACH NET: $\lambda = 0.5$; $\theta_{max} = 1.0$; $\theta_{min} = theta_{com} = 0.2$; $n_1 = 2$; $n_2 = 2$. THE LEFT COLUMNS IN 'ACCURACY' AND 'TIME' INDICATE THE RESULTS OF C-M^3-GZC OR CP-M^3-GZC WHILE THE RIGHT COLUMNS INDICATE THE RESULTS OF LVQ.

Data set	C-M^3-GZC / LVQ					CP-M^3-GZC / LVQ					No Cluster
	Size	Accuracy		Time		Size	Accuracy		Time		Accuracy
balance	92.0%	84.00%	87.20%	0.016	186	65.0%	**88.00%**	86.40%	0.048	110	84.00%
iris	16.0%	**97.33%**	90.67%	0.001	14	13.3%	**97.33%**	90.67%	0.002	13	94.67%
optdigits	77.8%	97.94%	75.63%	6.4	26638	38.8%	**98.11%**	75.85%	21	11092	98.00%



(a)



(b)

Fig. 6. Clustering result of industry image classification problem. The solid lines and dashdot lines represent the results of using our clustering method and not using clustering method, respectively. (a) Accuracy; (b) Number of clusters.

- It attach different attention to border and non-border instances.
- The post processing of outlier removal make the algorithm robust and cluster combination reduce the size of cluster further.

Experimental results on the artificial checkerboard problem and several real-world applications verify the validity of our algorithm. But in our experiments, we simply set $n_1 = 2$ and $n_2 = 2$ for all the data base. How to choose the optimum values of n_1 and n_2 for real-world applications needs to be analyzed in the future work.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60375022 and NSFC 60473040.

REFERENCES

[1] S. Basu, A. Banerjee, and R. J. Mooney, "Semi-supervised Clustering by Seeding," *Proc. 19th International Conference on Machine Learning*, pp.19-26, 2002.

[2] N. Bansal, A. Blum, and S. Chawla, "Correlation Clustering," *Machine Learning*, vol.56, pp.89-113, 2004.

[3] N. Grira, M Crucianu, and N Boujemaa, "Unsupervised and Semi-supervised Clustering: a Brief Survey," *A Review of Machine Learning Techniques for Processing Multimedia Content*, Report of the MUSCLE European Network of Excellence (FP6).

[4] B. Huang and B. L. Lu, "Fault diagnosis for industrial images using a min-max modular neural network," *ICONIP 2004, Lecture Notes in Computer Science*, vol.3316, pp.842-847, 2004.

[5] T. Kohonen, "The Self-Organizing Map," *Proc. Institute of Electrical and Electronics Engineers*, vol.78, pp.1464-1480, 1990.

[6] T. Kohonen, "improved versions of learning vector quantization," *IEEE International Joint Conference on Neural Networks*, vol.1, pp. 545-550, 1990.

[7] X. Li and N. Ye, "Grid-And Dummy-Cluster-Based Learning Of Normal And Intrusive Clusters For Computer Intrusion Detection," *Quality and Reliability Engineering International*, vol.18, pp.231-242, 2002.

[8] X. Li and N. Ye, "A Supervised Clustering Algorithm for Computer Intrusion Detection," *Knowledge and Information Systems*, vol.8, pp.498-509, 2005.

[9] B. L. Lu and M. Ichikawa, "A Gaussian zero-crossing discriminant function for min-max modular neural networks," *proc. 5th Inte'l Conf. Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*, pp.298-302, N.Baba et al. Eds., IOS Press, 2001.

[10] B. L. Lu and M. Ichikawa, "Emergent On-Line Learning with a Gaussian Zero-Crossing Discriminant Function," *IJCNN '02*, vol.2 pp.1263–1268, 2002.

[11] B. L. Lu and M. Ito, "Task decomposition based on class relations: a modular neural network architecture for pattern classification," *Lecture Notes in Computer Science*, vol.1240 pp.330-339, 1997

[12] B. L. Lu and M. Ito, "Task Decomposition and Module Combination Based on Class Relations: a Modular Neural Network for Pattern Classification," *IEEE Trans. Neural Networks,* Vol.10, pp.1244-1256, 1999.

[13] P. M. Murphy and D. W. Aha, "UCI Repository of Machine Learning Database," Dept. of Information and Computer Science, Univ. of Calif., Irvine, 1994.

[14] C. M. Procopiuc, "Clustering problems and their applications (a survey)," Department of Computer Science,Duke University.1997

[15] R. Sproull, "Refinements to Nearest-Neighbor Searching in *k*-Dimensional Trees," *Algorithmica*, pp.579-589, 1991.

[16] A. R. Webb, *Statistical Pattern Recognition, 2nd, Ed.* London, U.K.: Wiley, 2002.