A Local Learning Framework for Pattern Classification

Jianxiong Dong, Adam Krzyzak and C.Y. Suen
Centre for Pattern Recognition and Machine Intelligence,
Concordia University
Montreal, Quebec, Canada H3G 1M8
{jdong,suen}@cenparmi.concordia.ca, krzyzak@cs.concordia.ca

Abstract

The paper presents a local learning framework for pattern classification by partitioning a pattern space into different overlapped subsets and combining decisions in a local space. In contrast to designing a classifier on the global space, the advantage of the local learning framework is to reduce the complexity of component classifier which helps to enhance generalization. Our experimental results on handwritten digit database (CENPARMI and MNIST) are comparable with current best classifiers, which indicates that the proposed method is effective for classifying real world patterns. In addition, we have analyzed the characteristics of overlapped subsets and discovered that a simple average of the outputs of component classifiers achieves the best performance by optimizing the weights.

1 Introduction

In the classical pattern recognition problems, we select a classifier and tune it on a given training set. Our goal is to keep the misclassification error as small as possible on testing patterns extracted from the same distribution as the training patterns. The classification function approximates the global properties of the target concepts. But in real world problems, the size of the training set is not large enough to achieve this goal. Although the small values of the empirical risk are easily achieved on the training sets, this does not guarantee a good generalization without controlling the complexity of the designed classifier using the framework of statistical learning theory [1]. When the size of the training set is small compared to the complexity of the classifier, the learning machine usually overfits the noise in the training data. Thus effective control of complexity of a classifier plays a key role in achieving good generalization. Some theoretical results (see [1], [2]) and experimental results (see [3], [4]) indicate that a local learning algorithm (that is learning machine trained on the training subset) provides a feasible solution to this problem.

In recent years, learning based on the training subset has been an exciting research topic and some important theoretical and experimental results have been obtained. In fact, local learning is not a new concept and it has appeared in the early years of pattern recognition. The obvious example is the nearest neighbor method: given a testing pattern, we estimate its class from the closest pattern in the training set. Cover and Hart [5] proved that the asymptotic probability of error does not exceed twice the Bayes error. A well-known technique to estimate probability density function is the Parzen kernel estimate. For small training sets the performance of both methods is not very good because they do not use information about the class boundary when they are applied to pattern classification. Vapnik and Bottou [6] proposed theoretical model of a local learning algorithm and obtained bounds for the local risk minimization estimator for pattern recognition and regression problems using structural risk minimization principle. Although these estimated bounds are independent of the distribution, experiments have shown that they are quite loose in practice. Radial basis functions (RBF) [7] which have been justified by regularization theory [8] share many interesting properties with local learning algorithms. Compared with multi-layer perceptron, RBF interprets the data more intuitively. However, effective determination of the radial basis centers lacks satisfactory solution.

Another promising approach is the combination of classifiers in which individual classifiers are constructed on the training subsets and combined into a powerful decision system. Among these combination techniques, the best methods are bagging [9] and boosting [10]. These two methods rely on a random "sampling" technique to generate a training subset. In bagging, subsets of the raw training samples, randomly and independently selected with replacement

(bootstrap sampling) according to a uniform probability distribution, and used to construct a set of component classifiers, which are combined by a majority vote in the final decision system. In boosting, the creation of each subset depends on previous classification results. Consequently, AdaBoost adjusts the distribution of training instances and gradually concentrates it on the subsets of "hard" patterns(boundary patterns). It is important to discover boundary patterns using AdaBoost, which is similar to support vector machine (SVM) which extracts from the training data the support vectors that are most important in the classification. However, AdaBoost has two obvious weaknesses. One is that AdaBoost tends to overfit the noise when the training set contains some "outliers". The other disadvantage is that AdaBoost has not provided an ideal framework for classifier combination with some intrinsic structures. As a result, AdaBoost seems to enhance the performance of a weak classifier where complementary information from additional features can not be used to design the classifier.

In this paper, we attempt to construct a hierarchical local learning framework for classification. The framework is expected to satisfy three basic principles. First, a complex classification task is decomposed into sub-tasks that can be easily solved through partitioning a pattern space (or feature space) into different subsets. Second, the component classifiers with low complexity are constructed from these subsets and combined into an effective decision rule. Finally, the framework is formalized and its characteristics and performance can be analyzed by means of mathematical tools. The initial idea originates from Suganthan [3] who proposed a hierarchical overlapped SOM's for pattern classification. Two-level maps are constructed and trained by standard Kohonen's SOM algorithm [11]. For each neuron in the low-level layer, an upper-level SOM is constructed by using the training subset from the winner neuron and runner-up neurons. Consequently, a degree of overlap in the upper layer is achieved. This overlap makes it possible to make a final decision by fusing the decisions of several maps for every training and testing example. Atukorale [4] obtained good performance for handwritten digit recognition by replacing SOM with neural gas algorithm [12], which is an extension of SOM. But in the above models, SOM or neural gas are not ideal choices for the classification function of upper-level layer because a small training subset for upper-level maps is insufficient to estimate the probability density. Our experimental results also confirm this observation. In addition, we found that better performance can be achieved if we replace SOM or neural gas in the upper layer with other classifiers such as MLP taking into account the characteristics of the training subset.

The organization of this paper is as follows. We first describe the model. Then we discuss the problem of selecting the category of classification functions in two layers. In section 4, the model is applied to handwritten digits recognition and experimental results are analyzed. In addition, we compare the performance of our model with that of Atukorale. Finally, we draw conclusions.

1.1 Problem formulation

Before we describe our model, it seems appropriate to formally define some problems that the model can solve. First, suppose that we have input/output pairs $(x, y) \in X \times \{1, \cdots, m\}$,where $X \subseteq R^n$ is an input pattern space and m is the number of classes. We assume that the training and test data are generated independently and identically (i.i.d) according to an unknown distribution $f(x, y)$. The training data is denoted by:

$$S = \{(x_1, y_1), \cdots, (x_l, y_l)\} \tag{1}$$

where l is the number of training samples.

Now we partition the pattern space X into different clusters whose centers are w_i, $i = 1, \cdots, N$.

Procedure 1 *Let $X \subseteq R^n$ be generated from an unknown marginal distribution function $f_x(x)$. Our goal is to find N clustering centers by minimizing*

$$\int \sum_{i=1}^{N} \frac{\lambda_i(x, w_i)}{\sum_{i=1}^{N} \lambda_i(x, w_i)} \parallel x - w_i \parallel^2 f_x(x) dx \tag{2}$$

where N is a fixed number and $\| \cdot \|$ denotes the Euclidean vector norm. $w_i \in R^n$, $i = 1, \cdots, N$, λ_i is a "soft-max" factor

$$\lambda_i(x, w_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp(-\frac{\parallel x - w_i \parallel^2}{2\sigma_i^2}) \tag{3}$$

After determining the centers we partition the feature space using the following procedure

Procedure 2 *Assume that w_i, $i = 1, \cdots, N$ are known. The sets S_i where $i = 1, \cdots, N$ are empty initially. Then for every sample (x, y) in the training set,*

$$\begin{aligned} k_1 &= \text{argmin}_i \parallel x - w_i \parallel \\ k_m &= \text{argmin}_{i \notin \{k_1, \cdots, k_{m-1}\}} \parallel x - w_i \parallel \end{aligned} \tag{4}$$

where $m < n$. $S_l = \{x\} \cup S_l$, $l \in \{k_1, \cdots, k_m\}$.

Since the N clustering centers are known, each sample in the training set is put into the m nearest neighborhood, where m is the number of the overlapped neighborhoods. So we get partially overlapping subsets in a local area.

Procedure 3 *Let $\mathcal{H}$ denote the function space from which the base classifiers are chosen. $S_i \subseteq X$. The cardinality of set S_i is K_i. $h(x,\alpha) \in \mathcal{H}$, $\alpha \in \Lambda$. We minimize the empirical risk, where $I(\cdot)$ is an indicator function.*

$$h(x,\alpha_i^*) = \min_{\alpha\in\Lambda} \frac{1}{K_i} \sum_{x\in S_i} I(y_i \neq h(x,\alpha)) \tag{5}$$

Procedure 4 *Let $h(x,\alpha_i^*)$, $i = 1,\cdots,N$ be the base classifiers constructed on set S_i. For each sample x, we get*

$$\begin{array}{ll} k_1 & = \text{argmin}_i \parallel x - w_i \parallel \\ k_m & = \text{argmin}_{i\notin\{k_1,\cdots,k_{m-1}\}} \parallel x - w_i \parallel \end{array} \tag{6}$$

Then the final decision function can be generated by taking a weighted average of classifiers from $h(x,\alpha_i^*)$.

$$\eta : x \mapsto \sum_{k\in\{k_1,\cdots,k_m\}} \alpha_k h(x,\alpha_k^*) \text{ where } \alpha_k \geq 0 \sum_k \alpha_k = 1 \tag{7}$$

In order to analyze the performance of our model, we define a measure of overlap between two different training subsets S_i and S_j given by

$$d(S_i,S_j) = \frac{1}{2}\left(\frac{1}{|S_i|}\sum_{x\in S_i}\min_{z\in S_j}\|x-z\| + \frac{1}{|S_j|}\sum_{z\in S_j}\min_{x\in S_i}\|z-x\|\right) \tag{8}$$

where $\|.\|$ is Euclidean norm and $|.|$ is the cardinality of a set. It can be seen from the equation 8, that the smaller the measure $d(s_i,s_j)$, the greater the overlap between clusters s_i and s_j.

2 Model selection

The framework introduced in section 1.1 can be applied to different learning models. Procedure 1 is called vector quantization. Typical examples of these include self-organizing maps, learning vector quantization(LVQ) [11], neural gas and other extension models based on the first two (see [13], [14]). In our paper we select neural gas algorithm which can in contrast to SOM minimize a global cost function and adapt the weights of reference vectors without any fixed topological arrangement of neuron units. Procedures 3 and 4 correspond to supervised learning and classifier combination, respectively. Procedure 2 is an important part of our model that partitions the pattern space into different local overlapping subsets. Local learning can lower the complexity of the design classifier and improve the generalization. In the rest of the paper we describe our local learning framework, which is similar to HONG network developed by Atukorale [4]. We change the learning model in HONG network from neural gas to multi-layer perceptrons and combination method. In this paper, we introduce a new learning framework and thoroughly analyze it in numerous experiments.

The learning framework basically consists of two layers: the lower layer is for unsupervised learning and the upper layer for supervised learning. In the first layer the neural gas (NG) is applied to vector quantization. The initial size of the network depends on the number of classes and the size of the training set. We select it by the rule of thumb. The components of reference vectors are initialized with different uniform distributions whose variances are consistent with that of component variable. Atukorale's modified version of NG algorithm is applied to train the network of the first layer. During the training stage, neurons that never win during an epoch are deleted. After having completed the unsupervised learning, the neurons in the base layer are labelled using a simple majority voting method. Then LVQ is employed to fine tune the reference vector to obtain a smaller quantization error.

Next we employ Procedure 2 to partition the training set into different overlapping subsets in order to train the second-level MLP network. A training pattern is not only assigned to the training subset for a winning neuron but also to these training subsets for a predefined number of runner-up neurons. Thanks to this procedure the training samples become non-uniformly distributed among the neurons. Figure 1 depicts our model framework.

In order to combine the outputs of the overlapped second-level network, we consider a generalized committee prediction given by a weighted combination of the outputs:

$$\begin{array}{ll} \vec{y}_{gen} & = \sum_{i=1}^{L} \alpha_i \vec{h}(x,\alpha_k^*) \\ & = \vec{h}(x,\alpha^*) + \sum_{i=1}^{L} \alpha_i \epsilon_i(x) \end{array} \tag{9}$$

where $\sum_{i=1}^{L} \alpha_i = 1$, and L denotes the number of committee members. $\vec{h}$ is a class of mapping: $\vec{h} : x \in R^n \mapsto R^c$ where c is the number of classes and $\vec{h}(x,\alpha^*)$ is the target function. We define $\epsilon_i(x) = \vec{h}(x,\alpha_k^*) - \vec{h}(x,\alpha^*)$.

We introduce the error correlation matrix C with

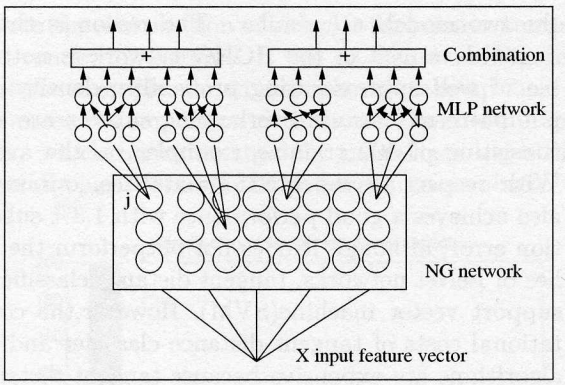

Figure 1: A framework of local learning

elements given by

$$C_{ij} = E[\epsilon_i^T(x)\epsilon_j(x)] \tag{10}$$

The error of generalized committee can be written

$$\begin{aligned} E_{gen} &= E[\|\vec{y}_{gen} - \vec{h}(x,\alpha^*)\|^2] \\ &= E[(\sum_{i=1}^{L}\alpha_i\epsilon_i^T(x))(\sum_{j=1}^{L}\epsilon_j(x))] \\ &= \sum_{i=1}^{L}\alpha_i\alpha_j E[\epsilon_i^T(x)\epsilon_j(x)] \end{aligned} \tag{11}$$

The optimal value of α_i can be obtained by minimizing E_{gen} under the linear constraint equation for α_i. By employing Lagrange multipliers, α_i was found to have the following form [15]

$$\alpha_i = \frac{\sum_{j=1}^{L}(C^{-1})_{ij}}{\sum_{k=1}^{L}\sum_{j=1}^{L}(C^{-1})_{kj}} \approx \frac{(C^{-1})_{ii}}{\sum_{i=1}^{L}(C^{-1})_{ii}} \tag{12}$$

In practice, matrix C is calculated by using a finite sample approximation

$$C_{ij} = \sum_{k\in\{k_1,\cdots,k_m\}} \frac{1}{N_k}\sum_{n=1}^{N_k}(\vec{y}_i(x^n) - t^n)^T(\vec{y}_j(x^n) - t^n) \tag{13}$$

where t^n is the target vector corresponding to the input vector x^n, N_k is the size of training subset that corresponds to the neuron unit k.

3 Experimental results

In this section we present the experimental results on testing and performance of our model. In addition we compare the performances of our model and other methods such as the HONG network.

Our experiments were performed on two well-known handwritten digit databases: MNIST database and CENPARMI database. In MNIST database the original gray-level images are centered in 28×28 box; therefore, binarization is first applied to generate binary images. MNIST database consists of 60,000 training samples and 10,000 test samples. CENPARMI database consists of 6000 unconstrained handwritten numerals originally collected from dead letter envelopes by the U.S. postal service at different locations. The numerals in this database are stored in bi-level format, whose resolution is approximately 166 PPI [16]. We used 4000 numerals for training and 2000 for testing. Figure 2 shows typical samples from MNIST database

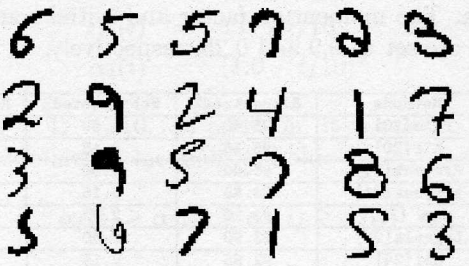

Figure 2: Typical samples in MNIST database.

In our experiments linear normalization and feature extraction based on the stroke edge are applied. All character images are size-normalized to fit the 32×32 box while preserving their aspect ratios. Besides normalization, a directional feature based on the gradient of gray scale image [17] is extracted by using the Robert edge operator. Several experiments have shown that this feature was effective for handwritten character recognition (see [18], [19]). After 400-dimensional feature vector has been extracted, principal component analysis was employed to compress the high dimensional feature vector to a low dimensional vector with only 60 dimensions in order to avoid the "curse of dimensionality". At the same time the computational cost was reduced dramatically.

Our first experiment evaluates the performance of our model on the two databases and compares it with other methods. Before we present the experimental results, some related parameter configurations are described. For neural gas algorithm the initial network size is set to 250. The decay factor λ decreases exponentially with the number of adaptation steps as $\lambda(t) = \lambda_i(\lambda_f/\lambda_i)^{t/t_{max}}$ with $\lambda_i = 10$, $\lambda_f = 0.01$. t_{max} is set to the maximal training time which is equal to epoch (≥ 200) timing the number of training samples. The step size ϵ[1] has the same time depen-

[1]Here, ϵ differs from that in section 3.

dence as λ i.e. $\epsilon(t) = \epsilon_i(\epsilon_f/\epsilon_i)^{t/t_{max}}$ with $\epsilon_i = 0.5$ and $\epsilon_f = 0.005$. For LVQ training, the learning step size decreases linearly with the number of steps t, i.e. $\alpha(t) = \alpha_0 \times (1.0 - t/t_{max})$ with $\alpha_0 = 0.005$. The predefined number of overlapping subsets was set to 10. Finally, in our model MLP is a fully-connected network with a simple structure $(60 \times 5 \times 10)$. We use the sigmoid activation function, i.e. $1.0/(1.0 + \exp(-x))$. All MLP's are trained using the gradient method with a momentum factor, which can avoid oscillations common with the standard backpropagation algorithm when the error surface has a very narrow minimum area. The momentum factor and initial learning step size are set to 0.9 and 0.25, respectively.

Methods	Recognized%	Substituted%	Rejected%
Kim[20]	95.40	4.60	0.00
Kim[20]	95.85	4.15	0.00
Krzyzak[21]	86.40	1.00	12.60
krzyzak[21]	94.85	5.15	0.00
Lam [22]	93.10	2.95	3.95
Legault[23]	93.90	1.60	4.50
Mai[24]	92.95	2.15	4.90
Suen[16]	93.05	0.00	6.95
Cho[25]	96.05	3.95	0.00
Dong[19]	98.00	2.00	0.00
HONG network	96.30	3.70	0.00
proposed method	97.50	2.50	0.00

Table 1: Comparison of performances of our model with other methods

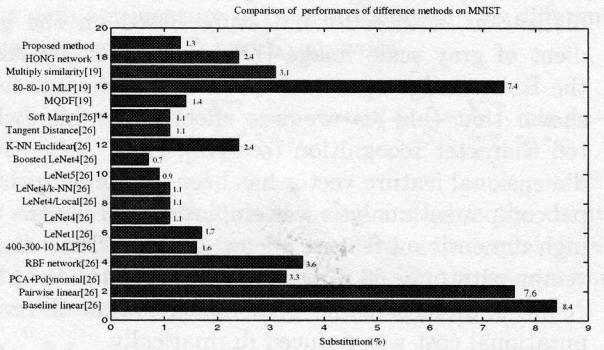

Figure 3: Error rate on the test set of MNIST database, each bar represents a classifier.

Table 1 and Figure 3 compare performances of different algorithms on the CENPARMI and MNIST databases, respectively. In our experiments, when HONG network is constructed, almost the same parameter configuration is applied to the first-level and second-level neural gas algorithm. For CENPARMI database, the proposed method which outperforms HONG network ranks 2 although the structures of the two models are similar. The reason is that the second-level NG in the HONG network is not capable of well approximating probability density of input patterns because overlapping subsets are small, consisting of 300 training examples on the average. With respect to the MNIST database, our method also achieves a good performance with 1.3% substitution error, although it does not outperform the families of LeNet networks, tangent distance classifier and support vector machine(SVM). However,the computational costs of tangent distance classifier and SVM algorithms are expensive because tangent distance is the type of nearest neighbor classifier and SVM needs to solve a quadratic programming problem during the test stage. Furthermore, our model has an inherent advantage in problems of pattern classification involving a large training set and a great number of categories such as Chinese character recognition due to our local learning model that effectively controls complexity of component classifiers without sacrificing the generalization performance.

Having completed the above performance comparison, we now analyze the characteristics of overlapping subsets to give an insight into our model and explain how the quality of subsets affects the overall performance. First, the distribution in the subsets is calculated to evaluate clustering quality of the first-level neural gas algorithm. Figure 4 gives the pseudocode of the main procedure.

Calculate the distribution of subsets

Input: a series of subsets A_i, $i = 1, \cdots, n$.

Output: distribution histogram $h[j]$, $j = 1, \cdots, c$, where c is the number of classes.

Initialize: $h[j] \leftarrow 0$, $j = 1, \cdots, c$.

for i=1 to n

1. Calculate frequency of examples in each class for A_i

2. Store them to array $freq[j]$, $j = 1, \cdots, c$.

3. Sort the array $freq$ in the decreasing order.

4. $h[j] \leftarrow h[j] + freq[j]$, $j = 1, \cdots, c$

end {for}

Figure 4: Procedure for calculating the distribution of subsets.

In the above procedure, the obtained histograms es-

timate the distribution of samples for winner neurons and runner-up neurons. It can be seen from Figure 4 that the examples for winner neurons take up most of resources in comparison with those of runner-up neurons, which indicates that the neural gas algorithm basically partitions similar patterns into a subset.

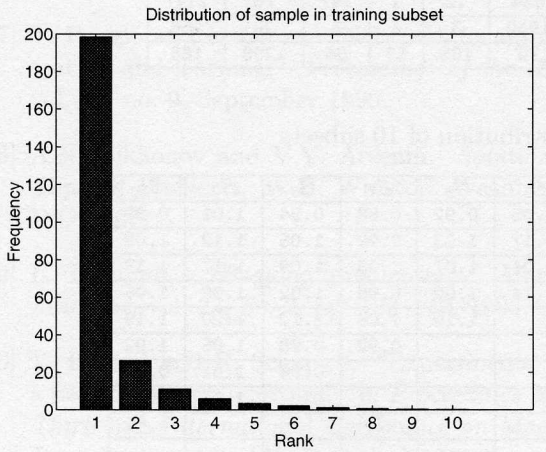

Figure 5: The distribution of samples for winner and runner-up neurons.

Now we take into account a case in which relation between characteristics of overlapping subsets and classification is investigated. We randomly pick an input pattern from the test set. When the input pattern whose class label is 9 is fed into our model, the final recognition is correct although the label of the winner neuron is 4. The class distribution of 10 subsets that corresponds to winner neuron and runner-up neurons is illustrated in Table 2. All neurons in the neural gas are indexed from 0 to 249.

In Table 2, S_1 is the subset corresponding to the winner neuron 210. From this table, we can observe that examples that belong to class 4 and 9 take up most resources, which indicates that in the local space, patterns that belong to classes 4 and 9 are similar. What is the topological structure between these subsets? Here we use a measure denoted by equation 5 to investigate their relationships.

We find from Table 3 that under this measure the distances between subsets are almost equal. It is interesting to note that these subsets are symmetric. It indicates that boundary patterns are contained in most subsets. Then a simple averaging will smooth the function for component classifier in the local neighborhood of "boundary patterns".

We now consider how to optimize the majority weights for each component classifier to obtain the best performance. Suppose that the farther the runner-up neuron from winner neuron is, the smaller contribution the component classifier that corresponds to this neuron makes to the final decision. If we index these classifiers such that the number for winner classifier is 0, and the number for the second best winner is 1 and so on, the weight will be a nondecreasing function whose variable is the index number. The following four weighted functions are applied:

$$\alpha_1(i) = 1 \quad (14)$$

$$\alpha_2(i) = \exp(-i/10) \quad (15)$$

$$\alpha_3(i) = 1.0 - i/10 \quad (16)$$

$$\alpha_4(i) = (1.0 - i/10)^2 \quad (17)$$

where $i = 1, \cdots, 10$. In addition, it is easy to verify that the following inequalities hold:

$$\alpha_1(i) \geq \alpha_2(i) \geq \alpha_3(i) \geq \alpha_4(i) \quad (18)$$

Our experimental results show that the equal weight function (α_1) achieves the best performance. This indicates that the above assumption is not completely correct. We employ the method from Section 3 to optimize the weights. The C^{-1} calculated on the overlapping subset as above is given in Table 4.

We can see from Table 4 that diagonal elements far exceed the sum of the non-diagonal elements in any column. So we can ignore non-diagonal elements and optimal weights are determined by diagonal elements. Thus the obtained optimal weights are almost identical, which confirms that a simple average vote achieves the best performance.

4 Conclusion

We have presented a local learning framework for pattern classification through partitioning a pattern space into different overlapping subsets and combining decisions in a local space. Local learning can reduce the complexity of component classifiers and improve the generalization performance although the global complexity of the system can not be guaranteed to be low. Our experimental results on handwritten character recognition exhibited excellent performance of the proposed method. This indicates that it is possible to obtain a feasible solution to problems of pattern classification in the real world by local learning because approximating global target function is hard given that usually not enough training samples are available. Besides extensive comparison of performance of different algorithms we have analyzed the characteristics of overlapping subsets in detail and found that topological relationships among these subsets are symmetric

Subset	Neuron	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
S_1	210	11	1396	9	30	1932	47	86	111	236	759
S_2	37	2	116	3	75	749	26	1	662	83	2080
S_3	230	8	839	9	116	696	49	2	792	217	2222
S_4	146	7	21	11	22	1577	20	3	252	217	2619
S_5	65	2	54	11	50	890	33	1	173	123	1659
S_6	90	1	18	2	3	2561	2	2	48	19	1484
S_7	120	1	23	2	15	1693	13	1	95	31	1324
S_8	179	2	3	18	25	534	12	1	147	78	2147
S_9	198	1	9	2	2	1960	3	4	8	6	456
S_{10}	32	83	135	68	260	74	165	57	64	1298	168

Table 2: Class conditional distribution of 10 subsets

Neuron	Subset	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
210	S_1		0.99	1.03	0.96	1.05	0.92	0.88	0.94	1.01	0.98
37	S_2			1.15	1.07	1.17	1.03	0.99	1.05	1.12	1.09
230	S_3				1.11	1.21	1.07	1.03	1.09	1.17	1.13
146	S_4					1.14	1.00	0.96	1.02	1.09	1.06
65	S_5						1.10	1.06	1.12	1.20	1.16
90	S_6							0.92	0.98	1.05	1.02
120	S_7								0.94	1.01	0.98
179	S_8									1.07	1.04
198	S_9										1.11
32	S_{10}										

Table 3: Overlapping distance between subsets

i/j	0	1	2	3	4	5	6	7	8	9
0	0.93*	0.07	-0.11	-0.11	-0.05	-0.13	-0.03	0.02	-0.07	-0.04
1	-0.07	0.65*	-0.19	-0.06	-0.16	-0.04	-0.01	-0.03	-0.03	-0.05
2	-0.11	-0.19	0.72*	-0.14	-0.08	-0.08	0.03	0.025	-0.10	-0.09
3	-0.11	-0.06	-0.14	0.95*	-0.00	-0.12	-0.01	-0.37	0.02	-0.04
4	-0.06	-0.16	-0.08	-0.00	0.93*	-0.05	-0.23	-0.07	-0.11	-0.02
5	-0.14	-0.05	-0.08	-0.12	-0.05	0.79*	0.02	-0.15	-0.21	0.03
6	-0.03	-0.02	0.03	-0.12	-0.23	0.02	0.95*	-0.27	-0.17	-0.03
7	0.02	-0.03	0.03	-0.37	-0.07	-0.15	-0.27	0.93*	-0.04	-0.02
8	-0.07	-0.03	-0.10	0.02	-0.11	-0.21	-0.17	-0.04	0.79*	0.00
9	-0.04	-0.05	-0.09	-0.04	-0.02	0.03	-0.03	-0.02	0.00	0.36*

Table 4: Inverse correlation matrix

under the defined measure. Moreover, by optimizing the combined weights our experiments showed that a simple averaging method is the best. In the future we will investigate the performance of this model on the classification problem with a large number of categories.

References

[1] V.N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[2] F. Girosi, M. Jones, and T. Poggio, "Regularization theory and neural networks architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, 1995.

[3] P.N. Suganthan, "Hierarchical overlapped som's for pattern classification," *IEEE Trans. Neural Networks*, vol. 10, no. 1, pp. 193–199, January 1999.

[4] A.S. Atukorale, P.N. Suganthan, and T. Downs, "On the performance of the hong network for pattern classification," in *Proceedings of the International Joint Conference on Neural Network(IJCNN'2000)*, pp. 285–290, Italy, July 2000.

[5] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Information Theory*, vol. 13, pp. 21–27, 1967.

[6] L. Bottou and V. Vapnik, "Local learning algorithm," *Neural Computation*, vol. 4, no. 6, pp. 888–901, 1992.

[7] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of the IEEE*, vol. 78, no. 9, September 1990.

[8] A.N. Tikhonov and V.Y. Arsenin, *Solutions of ill-posed Problems*, W.H. Winston, Washington, D.C., 1977.

[9] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[10] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148–156, Bari, Italy, 1996.

[11] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, Berlin, Germany, 2nd edition, 1997.

[12] T.M. Martinetz, S.G. Berkovich, and K.J. Schulen, "Neural gas network for vector quantization and its application to time-series predications," *IEEE Trans. Neural Networks*, vol. 4, no. 4, pp. 558–569, July 1993.

[13] B. Fritzke, "Growing cell structures–a self organizing network for unsupervised and supervised learning," *Neural Network*, vol. 7, no. 9, pp. 1141–1460, 1994.

[14] B. Fritzke, "A growing neural gas network learns topologies," in *Advances in Neural Information Processing Systems*, G. Tesauro, D.S. Touretsky, and T.K. Leen, Eds., vol. 7, pp. 625–632. MIT Press, Cambridge, MA, 1995.

[15] C.M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, 1995.

[16] C.Y. Suen, C. Nadal, T.A. Mai, R. Legault, and L. Lam, "Recognition of totally unconstrained handwritten numeral based the concepts of multiple experts," in *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pp. 131–143, Montreal, Canada, 1990.

[17] Y. Fujisawa, M. Shi, T. Wakabayashi, and F. Kimura, "Handwritten numeral recognition using gradient and curvature of gray scale image," in *Proceedings of International Conference on Document Analysis and Recognition*, pp. 277–280, India, August 1999.

[18] F. Kimura, T. Wakabayashi, S. Tsuruoka, and Y. Miyake, "Improvement of handwritten japanese character recognition using weighted direction code histogram," *Pattern Recognition*, vol. 30, no. 8, pp. 1329–1337, 1997.

[19] J. Dong, C.Y.Suen, and A. Krzyzak, "Comparision of algorithms for handwritten numeral recognition," Tech. Rep., CENPARMI, Concordia University, Montréal, Canada, November 1999.

[20] Y.J. Kim and S.W. Lee, "Off-line recognition of unconstrained handwritten digits using multilayer backpropagation neural network combined with genetic algorithm," in *Proc. 6th Workshop Image Processing Understanding*, pp. 186–193, 1994.

[21] A. Krzyzak, W. Dai, and C.Y. Suen, "Unconstrained handwritten character classification using modified backpropagation model," in *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pp. 155–166, Montréal, Canada, 1990.

[22] L. Lam and C.Y. Suen, "Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers," *Pattern Recognition*, vol. 21, no. 1, pp. 19–31, 1988.

[23] R. Legault and C.Y. Suen, "Contour tracing and parametric approximations for digitized patterns," in *Computer Vision and Shape Recognition*, A. Krzyzak, T. Kasvand, and C.Y. Suen, Eds., pp. 225–240. World Scientific, Singapore, 1989.

[24] T. Mai and C.Y. Suen, "A generalized knowledge-based system for the recognition of unconstrained handwritten numerals," *IEEE Trans. Systems, Man and Cybernetics*, vol. 20, pp. 835–848, 1990.

[25] S.B. Cho, "Neural-network classifiers for recognizing totally unconstrained handwritten numerals," *IEEE Trans. Neural Networks*, vol. 8, no. 1, pp. 43–53, January 1997.