PAPER

CombNET-III: a Support Vector Machine Based Large Scale Classifier with Probabilistic Framework

Mauricio KUGLER[†a)], ***Nonmember***, **Susumu KUROYANAGI**[†b)], **Anto Satriyo NUGROHO**[††c)], ***and*** **Akira IWATA**[†d)], ***Members***

SUMMARY Several research fields have to deal with very large classification problems, e.g. handwritten character recognition and speech recognition. Many works have proposed methods to address problems with large number of samples, but few works have been done concerning problems with large numbers of classes. CombNET-II was one of the first methods proposed for such a kind of task. It consists of a sequential clustering VQ based gating network (stem network) and several Multilayer Perceptron (MLP) based expert classifiers (branch networks). With the objectives of increasing the classification accuracy and providing a more flexible model, this paper proposes a new model based on the CombNET-II structure, the CombNET-III. The new model, intended for, but not limited to, problems with large number of classes, replaces the branch networks MLP with multiclass Support Vector Machines (SVM). It also introduces a new probabilistic framework that outputs posterior class probabilities, enabling the model to be applied in different scenarios (e.g. together with Hidden Markov Models). These changes permit the use of a larger number of smaller clusters, which reduce the complexity of the final classifiers. Moreover, the use of binary SVM with probabilistic outputs and a probabilistic decoding scheme permit the use of a pairwise output encoding on the branch networks, which reduces the computational complexity of the training stage. The experimental results show that the proposed model outperforms both the previous model CombNET-II and a single multiclass SVM, while presenting considerably smaller complexity than the latter. It is also confirmed that CombNET-III classification accuracy scales better with the increasing number of clusters, in comparison with CombNET-II.

key words: *large scale classification problems, support vector machines, probabilistic framework, divide-and-conquer*

1. Introduction

Several research fields have to deal with very large classification problems. Some examples are human-computer interface applications (e.g. speech recognition, handwritten character recognition, face detection), bioinformatics (e.g. protein structure prediction, gene expression) and data mining, in which huge amounts of data have to be processed in order to produce useful information. To meet the need of these applications, large scale classification methods have been receiving increasing attention, due to the need of adapting modern but computationally expensive classification methods for their efficient application.

Many authors addressed classification problems that present large number of samples. Jacobs *et al.* [1], [2] introduced the mixture of experts technique, dividing the problem in many small and simpler subtasks by the divide-and-conquer principle. In their approach, the problem is solved by many Multilayer Perceptron (MLP) "expert" classifiers whose outputs are weighted by a "gating" network (trained with the same data) according to their ability to classify each training sample. This principle was further extended to Support Vector Machines (SVM) based experts by Kwok [3] and Rida, Labbi and Pellegrini [4].

The majority of the large-scale classification methods, however, are not appropriate for problems containing large numbers of classes, e.g. classifying thousands of categories. This kind of problem usually also presents a large number of samples and/or features, as in the case of human-computer interface applications. In these cases, training the classifiers with all training samples, as suggested in [1], [2] is unfeasible. For example, MLP based experts would have thousands of output neurons and the SVM based experts would have either a huge number of classifiers or oversized kernel matrices. This is also the case when the splitting is made without any control of the size of each cluster or the balance among them. Iterative methods that constantly reassign the samples among the experts, as proposed by Collobert, Bengio and Bengio [5], [6], were initially designed for binary problems. The reassignments would constantly change the classifiers' structure, requiring restart of the training. Moreover, the initial random splits used in their approach would also generate experts with too many classes and very unbalanced subtasks. From this point, "large scale" will be used to refer to problems with large number of classes, unless stated otherwise.

The CombNET-II model proposed by Hotta *et al.* [7] was one of the first divide-and-conquer based large scale classifiers specifically developed for dealing with classification problems composed by thousands of cate-

Manuscript received October 14, 2005.
Manuscript revised March 23, 2006.

[†]The authors are with the Department of Computer Science & Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Japan.

[††]The author is with the School of Life System Science & Technology, Chukyo University, 101 Tokodachi Kaizu-cho Toyota, 470-0393, Japan.

a) E-mail: mauricio@kugler.com
b) E-mail: bw@nitech.ac.jp
c) E-mail: nugroho@life.chukyo-u.ac.jp
d) E-mail: iwata@nitech.ac.jp

gories. It has presented several good results in Chinese character (Kanji) recognition and some other specific applications. However, as the CombNET-II was originally developed for character recognition tasks, its application in different kinds of problems is not straightforward. Also, the algorithm used in the expert classifiers is the standard MLP, which, though presenting good classification results in previous researches, resulting in large processing time and problems of local minima during the training stage.

Arguing that CombNET-II spends too much time in the training and recognition processes because it uses all the available data in the expert networks training, Arai *et al.* [8]–[10] proposed the HoneycombNET, in which only a few reference vectors representing the data, found by vector quantization (VQ), are used on each expert. The model was further extended in order to reduce recognition time and to permit additional learning. In their ELNET model, Saruta *et al.* [11], [12] eliminated the subspace splitting procedure completely, saying that VQ based clustering methods are slow and, when using averaged vectors for speeding up, the performance of the gating network decreases. In ELNET, each class k has its own MLP expert network, which divides class k (excitation) from the most similar samples (inhibition), found by pattern matching among the samples of other classes.

These models, however, implement many heuristics for reducing processing time that lead them to digress from the basic idea of using the joint probabilities of gating and expert classifiers directly to construct the final answer. This reduces the flexibility of the models and complicates their extension. As to be shown in section 2, CombNET-II follows very closely that concept; thus, it is the most appropriate model for the proposed extensions of this research.

A few other models, based on different principles, have been proposed for solving classification problems with large number of categories. Fritsch and Finke [13] used a hierarchical clustering algorithm called Agglomerative Clustering based on Information Divergence (ACID) to divide the problem in subtasks with small number of classes. However, due to the huge amount of training samples that the upper nodes of the hierarchy had to be trained with, the computational cost was high. Hagihara and Kobatake [14] even proposed the use of large scale networks as the experts of a larger model, in which each expert was trained by a random subset of the classes and the results were combined in the end. Waizumi *et al.* [15] presented a new rough classification network for large scale models based on a hierarchy of Learning Vector Quantization (LVQ) neural networks. However, no definite result from the application of their gating network in a complete large scale model was presented.

The main objectives of this work are the improvement of the CombNET-II performance by the application of more modern pattern recognition algorithms and to develop a generic framework in order to enable its application in different scenarios. In order to accomplish this, a new model is introduced—the CombNET-III. The first objective was achieved by the application of Support Vector Machines (SVM) as the expert classifiers. For the generalization of the model, a new probabilistic framework able to comprise experts with different number of classes has been developed. It has to be noticed that, although intended for large scale problems, the model can also be applied to medium size problems, for instance, one with dozens of classes and a few thousands samples.

The organization of the paper goes as follows: a more detailed revision of CombNET-II is presented in Section 2, and Section 3 introduces the proposed model, its modifications and new characteristics. Section 4 presents experiments with the new model and some comparisons with CombNET-II, and Section 5 concludes the paper with analyses of the results and suggests possible future extensions.

2. Large Scale Classifier CombNET-II

The CombNET-II is a large scale classifier that follows the classic structure of divide-and-conquer methods: a gating network and many experts classifiers, called respectively "stem" network and "branch" networks in the original references [7], [16]. The stem network is a modified VQ based sequential clustering algorithm, called Self Growing Algorithm (SGA), developed to solve the problem of unbalanced clusters generated by the Self-Organizing Map (SOM) used in the original CombNET [16].

Sequential clustering algorithms are fast methods that use each example only a few times, making the method very suitable for large scale applications. Even though the final clusters depend on the order the samples are inputted, this is not so critical for large num-

Make $\nu_1 = \mathbf{x}_1$, $h_1 = 1$ and $R = 1$
for $i \in \{2 \dots \ell\}$
  Find ν_c so that:
    $sim\,(\nu_c, \mathbf{x}_i) = \max\limits_j [sim\,(\nu_j, \mathbf{x}_i)]$
  if $sim\,(\nu_c, \mathbf{x}_i) < \Theta_\mathrm{s}$
    $R = R + 1$, $\nu_R = \mathbf{x}_i$, $h_R = 1$
  else
    $\nu_c = \nu_c \cup \mathbf{x}_i$
    if $h_c > \Theta_\mathrm{p}$
      Divide ν_c in ν'_c and ν_{R+1} so that:
        $|h_c - h_{R+1}| \leq 1$
  end if
end for
do Update the clusters (with necessary divisions)
until No significant changes in any clusters

Fig. 1 Self Growing Algorithm (SGA)

bers of samples. Usually, sequential clustering algorithms have the similarity measurement threshold and the maximal number of clusters as their parameters. The SGA algorithm introduces another threshold to control the maximal inner potential (number of samples) of a cluster. The basic SGA algorithm is described in Figure 1, in which ℓ is the number of samples, R is the current number of clusters, $\mathbf{x}_i$ is the i^{th} sample, ν_j is the j^{th} cluster reference vector, Θ_s is the similarity threshold, Θ_p is the inner potential threshold, h_j is the j^{th} cluster inner potential and $sim\left(\nu_j, \mathbf{x}_i\right)$ represents the similarity measurement between the i^{th} sample and the j^{th} cluster. In its basic form, the CombNET-II uses the average vectors of each class as the training set for the stem network and the normalized dot product (the cosine of the angle between two vectors) as the similarity measurement.

After the stem network process is finished, all the samples belonging to class k will belong to the cluster that contains the reference vector of class k. Therefore, the input space is partitioned in R Voronoi subspaces, which will become the input spaces of the branch networks.

The CombNET-II uses MLP networks trained by gradient descent as the branch networks. These can be trained independently in order to reduce the total processing time. After the branch networks training, the class of an unknown sample $\mathbf{x}$ can be obtained as:

$$y = \omega_k \left| SM_j^\gamma \cdot SB_{jk}^{1-\gamma} = \max_{j'} \left(SM_{j'}^\gamma \cdot \hat{SB}_{j'k'}^{1-\gamma} \right) \right. \quad (1)$$

where:

$$SM_j = sim\left(\nu_j, \mathbf{x}\right) = \frac{\langle \nu_j, \mathbf{x} \rangle}{|\nu_j|\,|\mathbf{x}|} \quad (2)$$

$\hat{SB}_{j'k'}$ is the maximal score among the output neurons of the j^{th} branch network and ω_k is the k^{th} possible category, $k = 1, \ldots, K$. The exponent γ is a weighting parameter ($0 \leq \gamma \leq 1$) that dictates which network (stem or branch) plays the major role in the classification. The basic structure of the CombNET-II is shown in Figure 2.

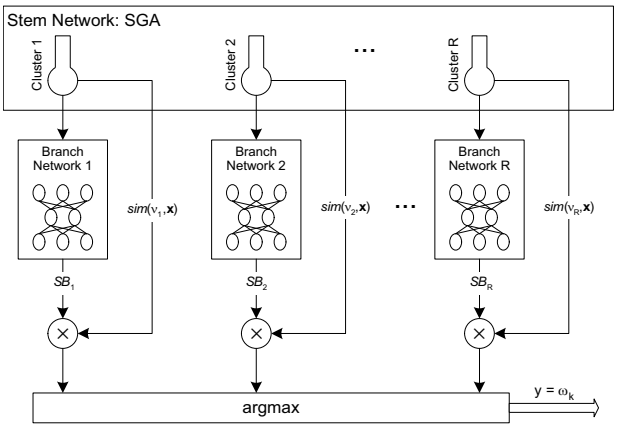

Fig. 2 CombNET-II structure

3. Proposed Model: CombNET-III

The main modification to CombNET-II proposed in this paper is the substitution of the MLP branch networks by multiclass Support Vector Machines based branch networks. Moreover, as mentioned by many authors, a classifier should output posterior class probabilities to allow post processing [17], [18]. This characteristic is required when the classifier is part of another system, for instance, when it is used for the association of HMM states with phonemes in speech recognition, and also facilitates the cascading of classifiers. However, neither CombNET-II nor any of the other large scale models for large number of classes problems commented before (except for the ACID model) present a probabilistic framework. The heuristics for reducing the recognition time in [8]–[12] makes it more difficult to obtain such a kind of outputs.

Support Vector Machine [19], [20] is a structure risk minimization based method that has been successfully applied in many classification tasks with great generalization performance. Due to its high computational and memory cost ($O\left(\ell^3\right)$ and $O\left(\ell^2\right)$, respectively, for ℓ training samples and a naive implementation), the application of SVM in classifications problems with large numbers of samples still remains as a challenge. However, for problems with large numbers of classes in which the number of samples per class is limited, SVM becomes an interesting option as an expert classifier. Therefore, it is selected as the algorithm for the CombNET-III's branch networks. The basic SVM decision function is:

$$f\left(\mathbf{x}\right) = \sum_{n \in SV} y_n \alpha_n K\left(\mathbf{x}_n, \mathbf{x}\right) + b \quad (3)$$

where $\mathbf{x}_n$ is the n^{th} support vector, y_n is the label of the n^{th} support vector, $K\left(\mathbf{x}_n, \mathbf{x}\right)$ is the Kernel function, α_n is the Lagrange multiplier of the n^{th} support vector and b is the bias. The last two terms are found by means of the minimization of a convex quadratic problem.

The application of SVMs as expert classifiers in a divide-and-conquer model, however, is not straightforward. The SVMs unlimited output function of equation (3) and different output ranges among classifiers make the output combination inefficient [18]. Many approaches address the problem of converting the SVM output in a calibrated probability. In this paper, Platt's methodology [17] was used, which consists of the direct conversion of the function values to posterior probabilities by fitting the SVM output with a sigmoidal function. This solution has the desirable property of maintaining the sparseness of the solution. In order to obtain the sigmoid parameters, Platt used a model

trust minimization algorithm in his experiments. In this paper, the Conjugate Gradient (CG) Minimization Method [21] was used. Platt also observed that using the same data for training the SVM and for the sigmoid optimization can sometimes lead to biased fits. However, this problem was not observed in the experiments presented in this work, which is also the case reported in [22].

After the SVMs outputs are moderated, they must be decoded properly, independent of the encoding scheme used. Passerini, Pontil and Frasconi [18] proposed a new decoding procedure for multiclass SVM using error correcting output encodings that outperformed other decoding methods, such as hamming distance and loss based decoding. It also generates a posterior class probability. This method, however, outputs calibrated probabilities that do not directly reflect the classifiers confidence on the overall sample space. Instead, a proportional probability is given. The direct use of this kind of decoding would make the system very dependent on the gating network classification. This is undesirable, as the gating network usually presents a low classification accuracy. This paper introduces a new decoding function in order to obtain adequate measures from the branch networks.

As the classifiers corresponding to one class were trained with the same samples of that class, their output probabilities are not statistically independent. Thus, given a coding matrix $\mathbf{M}^{K\times H}$ in which K is the number of classes and H is the number of classifiers, $m_{k,h} = \{-1, 0, +1\}$ and zero entries are interpreted as "don't care", the probability of class ω_k given an unknown sample $\mathbf{x}$ and a cluster ν_j is defined as the average probability outputted by the classifiers containing that class. The proposed decoding function hence becomes:

$$P(\omega_k|\mathbf{x},\nu_j) = \frac{\sum\limits_{h:m_{k,h}\neq 0} P(y_{j,h} = m_{k,h}|\mathbf{x})}{\sum\limits_{h=1}^{H} |m_{k,h}|} \qquad (4)$$

Fritsch and Finke [13] said that the One-versus-Rest (OvR) encoding is a prerequisite for training neural networks in order to estimate posterior probabilities, which are converted in calibrated posterior probabilities by a *softmax* [23] activation function. The proposed probability decoding eliminates this prerequisite, allowing the use of less time consuming encodings in training, such as the One-versus-One (OvO) scheme [24]. As, in general, the large scale problems with large number of classes do not have such a large number of samples per class, the OvO encoding was used in this work, although any other encoding could have been used.

The stem network uses the average of each class as training data in order to control the number of classes per cluster and avoid unbalanced problems on the branches. However, there is no constrain for each class to belong to only one cluster. If strategies other than the use of averaged data are used, classes belonging to multiple branch networks can occur. Hence, the events related to the class predicted by one branch network are not mutually exclusive, and the probabilities obtained with equation (4) are not calibrated. The final structure of the SVM based branch network is shown diagrammatically in Figure 3.

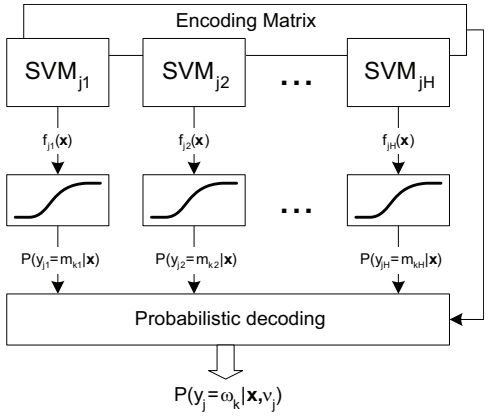

Fig. 3 SVM based Branch Network structure

The events of different clusters, however, are statistically independent, as the stem network generates a "hard" split of the samples and each branch is trained with independent data. Also, the clusters posterior probabilities are calculated from a similarity measurement that considers each cluster individually. Hence, when one cluster gives maximal probability, the probability of other cluster is not null, meaning that they are not mutually exclusive.

The divide-and-conquer probabilistic approaches normally use the total probability theorem for combining the probabilities of the expert networks. However, this theorem considers that the clusters probabilities are mutually exclusive and add up to unity. Furthermore, in the case of unbalanced clusters (i.e. in the case of different number of classes for each cluster), if the total probability theorem is naively used, the branch networks with fewer classes tends to dominate the outlier space. The reason for this is that the branch networks outputs are considered as mutually exclusive, instead of statistically independent. Therefore, this paper proposes a new framework for combining the branch network results.

As a branch network cannot give any information about the categories that it was not trained to recognize, it is assumed that:

$$\omega_k \notin \nu_j \rightarrow P(\omega_k|\mathbf{x},\nu_j) = \frac{1}{2} \qquad (5)$$

The cluster probability $P(\nu_j|\mathbf{x})$ represents the confidence of each branch network output, i.e, it weights between the branch network outputs and 1/2. Hence, the final posterior probability of the class ω_k

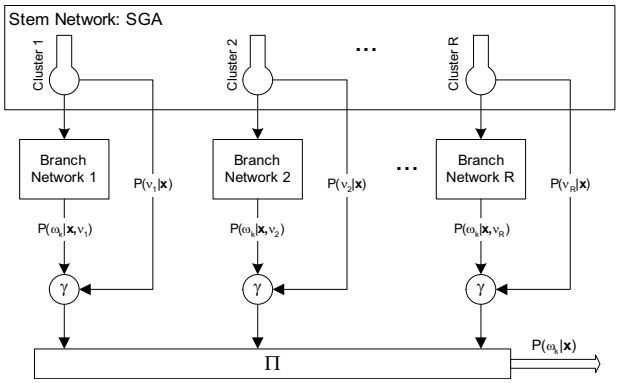

Fig. 4 CombNET-III structure

given an unknown sample $\mathbf{x}$ is calculated as the product of the probability of class ω_k given by each branch network weighted by the respective cluster probability. Finally, the proposed framework final equation can be written as:

$$P(\omega_k|\mathbf{x}) = c\prod_{j=1}^{R}\left[P(\nu_j|\mathbf{x})^{\gamma}P(\omega_k|\mathbf{x},\nu_j)^{1-\gamma} + \right. \\ \left. + \frac{1-P(\nu_j|\mathbf{x})^{\gamma}}{2}\right] \quad (6)$$

where the term c before the product is used to adjust the probabilities scale in order to ensure they are calibrated, summing to unity. Also, as the stem network cluster posterior probability and the branch networks class probabilities are obtained using very different procedures, a weighting factor γ similar to the one used in CombNET-II has to be used. The final structure of the CombNET-III is shown diagrammatically in Figure 4.

When the kernel function of the SVM branch networks is the Gaussian function, the branch networks outputs for an outlier sample tend to zero. Thus, equation (6) tends to generate equiprobable outputs for all classes, as the normalized cosine base stem network also tends to output equiprobable clusters. These are desirable properties, as the interference of one branch network in the other branches sample space tends to be minimized. Also, it is statistically consistent, as the classifier does not have information about the outlier space and should not produce any biased output.

4. Experiments

Two databases were used to illustrate the advantages of the proposed model over previous methods. The *Alphabet* database is not a large scale problem, having few number of categories and a few thousand samples, and can be solved using most standard classification methods. However, as the branch networks parameters can be extensively optimized for each experiment realization, the scaling properties of CombNET-II and CombNET-III with increasing number of clusters can be observed. The *Kanji400* is a much larger database for which standard classifiers starts to present poor performance or large complexity. For this database, the proposed method classification accuracy is compared with other traditional classifiers. All experiments were performed using in-house developed software packages.

Table 1 *Alphabet* database stem network SGA training parameters

Number of Clusters	Similarity Threshold	Inner Potential Threshold
1	-1	30
2	-1	15
3	0.1	14
4	-1	8
5	0.45	8
6	-1	6
7	0.75	6
8	0.7	5

4.1 JEITA-HP *Alphabet* Database

This database consists of the roman alphabet characters subset of the JEITA-HP database † dataset A. The first 200 samples of each character from A to Z were selected for the experiment, with 150 for training (3900 samples) and 50 for testing (1300 samples). The raw characters, which are composed of 64x64 binary values representing black and white dots, were preprocessed by a Local Line Direction (LLD) feature extraction method [25], which generated 256 features. Each sample vector was normalized to a unitary maximal feature value and zero feature mean. This vector normalization improves the normalized dot product similarity measurement efficiency.

The *Alphabet* database was evaluated by the traditional CombNET-II using MLP branch networks, with the evaluation procedure of equation (1), and the proposed CombNET-III model using Gaussian Kernel SVMs as the expert classifiers, under the framework of equation (6). The stem network was trained with several parameters in order to obtain increasing number of clusters, with the best possible balance of number of classes between them and no single-class cluster. For balanced cluster, the non-optimal procedure of using the same set of parameters for all the branches gives acceptable results. The same trained stem networks were used for CombNET-II and CombNET-III evaluation. Table 1 shows the parameters used to train each stem network.

Figures 5 and 6 depict the results for CombNET-II and CombNET-III respectively, showing the variation of the stem (dark circles' dotted line) and branch

†Available under request from http://tsc.jeita.or.jp/ /TSC/COMMS/4_IT/Recog/database/jeitahp/index.html

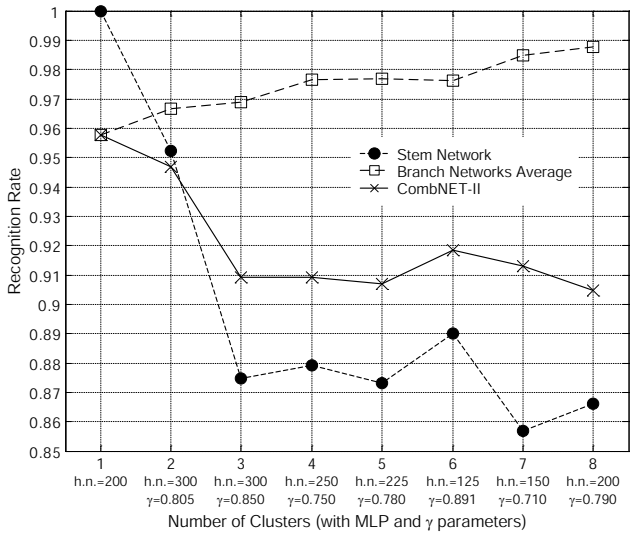

Fig. 5 CombNET-II recognition rate results for the *Alphabet* database

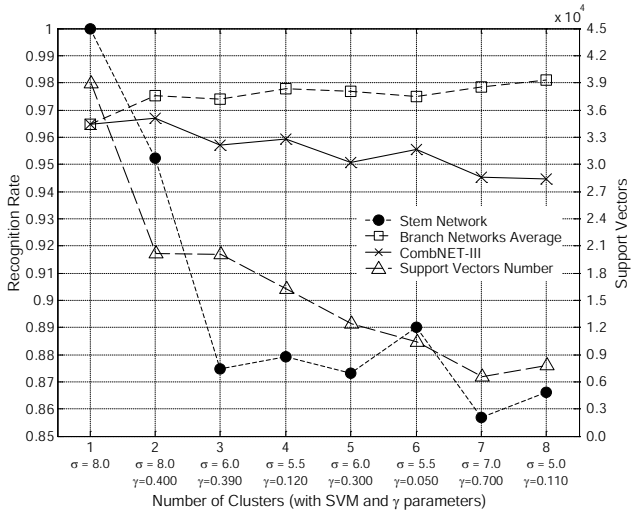

Fig. 6 CombNET-III recognition rate results for the *Alphabet* database

(squares' dashed line) networks and the whole structure (crosses' solid line) recognition rates with the increase of the number of clusters in which the data is divided. Figure 6 also shows the variation of the sum of the number of support vectors in each cluster (diamonds' dashed line). Under the x-axis, the optimized parameters for each number of clusters are shown.

As expected, the CombNET-III performed better than CombNET-II for all cases, specially for large number of clusters, even though the MLP branch networks average classification accuracy is slightly higher than the SVM based branches. Surprisingly, although the *Alphabet* database is small enough for single classifiers, the proposed model with 2 clusters outperformed the single multiclass SVM. The rapid decay of the number of support vectors numbers also shows that CombNET-III can be faster on classification than a single SVM classifier, for instance, the 2 clusters CombNET-III presents around half of the number of support vectors achieved by the single SVM.

4.2 ETL9B *Kanji400* Database

This database consists of a subset of the first 400 categories of the ETL9B database [†]. The performance of the proposed model CombNET-III was compared with the previous model CombNET-II, a single multiclass SVM and the k-NN method. As it is very difficult to obtain a good convergence with a single MLP in a 400 classes problem due to local minima, this comparison was not performed. Moreover, even a single parameter set experiment would be very time consuming.

The ETL9B database contains 3036 categories, 2965 Chinese characters (Kanji) and 71 Japanese Hiragana characters. The first 400 classes were used, each contains 200 samples, from which 150 samples were used as the training set and 50 samples as the test set. The characters were resized by their largest dimension and the peripheral direction contributivity (PDC) feature extraction method [26] was applied. For all classifiers except the k-NN, before the features normalization, each sample vector was independently normalized to a unitary maximal feature value and zero feature mean.

The k-NN method was run for all odd values of k from 1 to 55. The data was normalized to zero mean and unitary standard deviation. For the CombNET-II experiments, the MLP neural networks were trained until the error was smaller than 10^{-4} or the iteration number exceeds 500, with learning rate equal to 0.1, momentum 0.9 and sigmoidal activation function slope 0.1, while the number of hidden neurons and the γ parameter were optimized (by testing several values) for each experiment realization.

In the case of the single SVM and the CombNET-III, the binary SVM classifiers had non-biased output and a Gaussian kernel function, whose parameter σ was optimized for each experiment realization. The soft-margin C parameter was fixed at 200 (as several experimented values did not produce significant changes for the used data). For CombNET-III, each branch network training data was normalized to zero mean and unitary standard deviation.

Both divide-and-conquer models CombNET-II and CombNET-III used the same 12-cluster stem network, which was trained with similarity threshold and inner potential threshold respectively equal to -1 and 53. As these experiments are very time-consuming, specially for CombNET-II branch networks training, no other number of clusters were used. However, this configu-

[†]Available under request from http://www.is.aist.go.jp/etlcdb

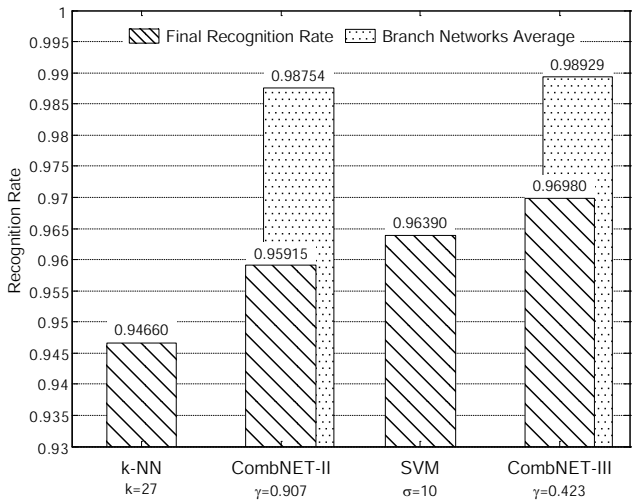

Fig. 7 Recognition rate results comparison for the *Kanji400* database

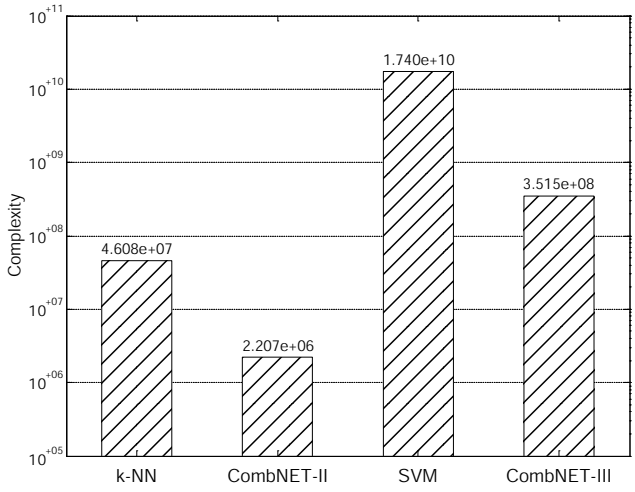

Fig. 8 Final classifiers complexity results comparison for the *Kanji400* database

ration is very appropriate, as the branch networks can perform very well and the stem performance of 78.70% is also acceptable. For these models each branch network parameters were optimized independently.

Figure 7 depicts the classification accuracy results for the proposed method and all compared methods. For the divide-and-conquer methods, it is also shown the branch networks average accuracy. The proposed model outperformed the other methods, reducing the single SVM error rate by around 16% and the previous model CombNET-II by around 26%. As stated before, it is difficult to obtain good convergence for a single MLP with this amount of categories. Therefore, Figure 7 does not include such a result. Figure 8 depicts the complexity for all compared methods, illustrating the amount of memory and calculation required for each model after training. Table 2 describes the complexity definition for each model, in which N is the number of features, ℓ is the number of training samples, R is the number of clusters on the case of divide-and-conquer methods, W is the total number of weights and biases of a MLP and SV is the final number of support vectors in a multiclass SVM. It is to be noticed that the y-axis is in logarithmic scale.

Table 2 Classifiers computational complexity description

Classifier	Complexity Description
k-NN	$N\ell$
CombNET-II	$\sum_{j=1}^{R} W_j$
single SVM	$N \cdot SV$
CombNET-III	$N \sum_{j=1}^{R} SV_j$

The results show that, even the performance of the single multiclass SVM is not so far from the one obtained by CombNET-III, the final classifier's complexity is two orders of magnitude higher. Even changing the kernel parameters, a similar complexity for the single SVM could not be obtained, while the accuracy drops beyond all other methods.

When compared to the previous model CombNET-II, the CombNET-III complexity is higher. However, as the accuracy of CombNET-II is very dependent on the stem network (as the high values of γ under the x-axis of Figure 5 indicate), the performance for the used number of clusters is considerably lower than CombNET-III, even the branch networks average accuracy is nearly the same for both models.

These results confirm the expected advantages of the proposed model CombNET-III on large scale problems classification.

5. Discussion and Conclusions

This paper proposed an extension of the previous large scale classification model CombNET-II. On the development of this new model, named CombNET-III, the following points were addressed: the classification accuracy improvement, the reduction of the large training computational cost of the CombNET-II MLP based branch networks, and the development of a new framework that could output posterior probabilities, enabling it to be used on different applications.

Substituting the MLP branch networks by multiclass SVMs with moderated outputs permitted the first two objectives to be achieved. The local effect of the Gaussian kernel function reduces the interference between the clusters, as the SVM function value tends to be zero for outlier samples. This allows an increase in the importance given to the branch classification result, shown by the small values of γ obtained on the experiments, in comparison with CombNET-II. Also,

although no numerical measurement was presented, the use of the OvO encoding makes the CombNET-III training time to be at least one order of magnitude faster than both CombNET-II and the single multiclass SVM. Finally, the final classification accuracy of CombNET-III outperformed all the compared methods (k-NN, single SVM and CombNET-II), showing that the proposed framework and the use of SVM branch networks are effective.

Future works include the improvement of the stem network, in order to increase its classification accuracy, which will probably result in an improvement of the whole classifier structure. Also, even the CombNET-III complexity is considerably less than the single multiclass SVM, it is still higher than CombNET-II. Techniques such as feature subset selection could be used in order to reduce the classification complexity.

Acknowledgments

The first author is supported by the Ministry of Education, Culture, Sports, Science and Technology, Government of Japan, and also by a grant from the Hori Information Science Promotion Foundation, Japan. The first author also thanks to Hirotaka Okui for his help on the understanding and implementation of the compared methods and Ranniery da Silva Maia for his help on checking the proposed model equations. The research of the third author is partially supported by the Grant-in-Aid for Private University High-Tech Research Center from Ministry of Education, Culture, Sports, Science and Technology, Government of Japan.

References

[1] R.A. Jacobs, M.I. Jordan, G.E. Hinton, and S.J. Nowlan, "Adaptive mixtures of local experts," Neural Computation, vol.3, no.1, pp.79–87, 1991.

[2] M.I. Jordan and R.A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," Neural Computation, vol.6, no.2, pp.181–214, March 1994.

[3] J.T.Y. Kwok, "Support vector mixture for classification and regression problems," Proceedings of the International Conference on Pattern Recognition (ICPR'98), Brisbane, Queensland, Australia, pp.255–258, 1998.

[4] A. Rida, A. Labbi, and C. Pellegrini, "Local experts combination through density decomposition," International Workshop on AI and Staticstics (Uncertainty'99), Morgan Kaufmann, 1999.

[5] R. Collobert, S. Bengio, and Y. Bengio, "A parallel mixture of SVMs for very large scale problems," Neural Computation, vol.14, no.5, pp.1105–1114, May 2002.

[6] R. Collobert, S. Bengio, and Y. Bengio, "Scaling large learning problems with hard parallel mixtures," International Journal on Pattern Recognition and Artificial Intelligence, vol.17, no.3, pp.349–365, 2003.

[7] K. Hotta, A. Iwata, H. Matsuo, and N. Susumura, "Large scale neural network CombNET-II," IEICE Transactions on Information & Systems, vol.J75-D-II, no.3, pp.545–553, March 1992.

[8] M. Arai, J. Wang, K. Okuda, and J. Miyamichi, "Thousands of hand-written kanji recognition by "HoneycombNET"," IEICE Transactions on Information & Systems, vol.J76-D-II, no.11, pp.2316–2323, November 1993.

[9] M. Arai, K. Okuda, and J. Miyamichi, "Thousands of handwritten kanji recognition by "HoneycombNET-II"," IEICE Transactions on Information & Systems, vol.J77-D-II, no.9, pp.1708–1715, September 1994.

[10] M. Arai, K. Okuda, H. Watanabe, and J. Miyamichi, "A large scale neural network "HoneycombNET-III" that has a capability of additional learning," IEICE Transactions on Information & Systems, vol.J80-D-II, no.7, pp.1955–1963, July 1997.

[11] K. Saruta, N. Kato, M. Abe, and Y. Nemoto, "A fine classification method of handwritten character recognition using exclusive learning neural network (ELNET)," IEICE Transactions on Information & Systems, vol.J79-D-II, no.5, pp.851–859, May 1996.

[12] K. Saruta, N. Kato, M. Abe, and Y. Nemoto, "High accuracy recognition of ETL9B using exclusive learning neural network - II (ELNET-II)," IEICE Transactions on Information & Systems, vol.E79-D, no.5, pp.516–522, May 1996.

[13] J. Fritsch and M. Finke, "Applying divide and conquer to large scale pattern recognition tasks," Lecture Notes In Computer Science, (Neural Networks: Tricks of the Trade), vol.1524, London, UK, pp.315–342, Springer-Verlag, 1998. This book is an outgrowth of a 1996 NIPS workshop.

[14] Y. Hagihara and H. Kobatake, "A neural network with multiple large-scale subnetworks and its application to recognition of handwritten characters," IEICE Transactions on Information & Systems, vol.J82-D-II, no.11, pp.1940–1948, November 1999.

[15] Y. Waizumi, N. Kato, K. Saruta, and Y. Nemoto, "High speed and high accuracy rough classification for handwritten characters using hierarchical learning vector quantization," IEICE Transactions on Information & Systems, vol.E83-D, no.6, pp.1282–1290, June 2000.

[16] A. Iwata, T. Touma, H. Matsuo, and N. Suzumura, "Large scale 4 layered neural network "CombNET"," IEICE Transactions on Information & Systems, vol.J73-D-II, no.8, pp.1261–1267, August 1990.

[17] J.C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in Advances in Large Margin Classifiers, ed. A.J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, pp.61–74, MIT Press, Cambridge, MA, March 1999.

[18] A. Passerini, M. Pontil, and P. Frasconi, "New results on error correcting output codes of kernel machines," IEEE Transactions on Neural Networks, vol.15, no.1, pp.45–54, January 2004.

[19] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol.20, no.3, pp.273–297, 1995.

[20] N. Cristianini and J. Shawa-Taylor, An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, Cambridge, 2000.

[21] J.R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," Available at http://www-2.cs.cmu.edu/∼quake-papers/painless-conjugate-gradient.pdf, August 1994.

[22] L.S. Oliveira and R. Sabourin, "Support vector machines for handwritten numerical string recognition," Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9), Tokyo, Japan, pp.39–44, October 2004.

[23] J. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," Neurocomputing: algorithm, ar-

chitectures, and applications, ed. F. Fogelman-Soulie and J. Herault, New York, pp.227–236, Springer-Verlag, 1990.
[24] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," Advances in Neural Information Processing Systems, ed. M.I. Jordan, M.J. Kearns, and S.A. Solla, The MIT Press, 1998.
[25] H. Kawajiri, T. Yoshikawa, J. Tanaka, A.S.Nugroho, and A. Iwata, "Handwritten numeric character recognition for facsimile auto-dialing by large scale neural network CombNET-II," Proceedings of the 4th International Conference on Engineering Application of Neural Networks, Gibraltar, pp.40–46, June 1998.
[26] N. Hagita, S. Naito, and I. Masuda, "Chinese character recognition by peripheral direction contributivity feature," IEICE Transactions on Information & Systems, vol.J66-D, no.10, pp.1185–1192, October 1983.

Mauricio Kugler received the degree in electrical engineering in 2000, and the MSc degree in biomedical engineering in 2003, both from the Federal Center of Technological Education, Brazil. Since April 2003, he is a PhD student in the Department of Computer Science and Engineering of the Nagoya Institute of Technology, Japan. His research interests include machine learning, large scale pattern recognition methods, feature selection methods, biomedical signals processing and hardware programming. He is a member of the Institute of Electrical & Electronics Engineers (IEEE).

Susumu Kuroyanagi received a B.S. in 1991 from the Department of Electrical and Computer Engineering at the Nagoya Institute of Technology. He completed the first half of the doctoral program in 1993 and the second half in 1996, receiving the D.Eng. degree from the same institute. In 1996, he became a research associate in the Department of Electrical and Computer Engineering at the Nagoya Institute of Technology, and, in 2003, a research associate in the Graduate School of Engineering, at the Department of Computer Science and Engineering. Since 2006, he has been an associate professor in this same Graduate School. He is engaged in researches about neural networks and auditory information processing, also being a member of the Acoustic Society of Japan, the Japan Neural Network Society and Japanese Society for Medical and Biological Engineering.

Anto Satriyo Nugroho is a visiting professor in the School of Life System Science and Technology, Chukyo University, Japan. He received his B.Eng degree in 1995, M.Eng degree in 2000, and D.Eng degree in 2003, all in Electrical and Computer Engineering from Nagoya Institute of Technology, Japan. He is also working as a scientist in Agency for the Assessment and Application of Technology (BPPT), of the Republic of Indonesia. His research interest is in the field of pattern recognition, bioinformatics and biomedical engineering. Dr. Anto Satriyo Nugroho is a member of The Institute of Electrical & Electronics Engineers (IEEE), Information Processing Society of Japan (IPSJ) and Japanese Society for Bioinformatics (JSBI).

Akira Iwata received a B.S. in 1973 from the Department of Electrical Engineering, Faculty of Engineering, Nagoya University. He completed the M.E. program in 1975 and became a research associate in the Department of Information, Nagoya Institute of technology. He was a visiting researcher from April 1982 to October 1983 in the research Institute of Medical information, University of Giessen Medical School, Germany. He became an associate professor in the Department of Information, Nagoya Institute of Technology in 1984, and a professor in the Department of Electrical and Computer Engineering in 1993, and vice president in 2002, and has been a professor in the Department of Computer Science and Engineering, Graduate School, since 2004. He is engaged in research on neural networks and internet security, He holds a D.Eng. degree. He received an IEICEJ paper Award in 1993 and an Information Processing Society Best Author Award in 1998. He is a member of the Information Processing Society, JSMEBE, the Japan Electrocardiography Society, the Japan Neural Network Society, and Japan Society for Medical Information Processing. He is an IEEE Senior Member.