Novel Hierarchical Pattern Classification based on Probabilistic Neural Networks

by

Masaru OKAMOTO

Graduate School of Engineering
Hiroshima University

March, 2009

Contents

1 Introduction **1**

1.1 Background 1

1.2 Purpose 4

1.3 Related Works 4

1.3.1 Tree-Based Models 5

1.3.2 Boosting Approach 5

1.3.3 Unsupervised Learning and Estimating the Number of Classes . 6

1.3.4 Elimination of Unexpected Data 7

1.4 Outline of Thesis 8

2 A Tree-based Hierarchical Probabilistic Neural Network **11**

2.1 Introduction 11

2.2 H-LLGMN 11

2.2.1 Construction Hierarchical Tree 12

2.2.2 Construction of Metaclass 13

2.3 Hand Shape Classification Experiment 14

2.4 Concluding Remarks 20

3 Pattern Discrimination using Probabilistic Neural Networks based on Boosting Algorithms **23**

3.1 Introduction 23

3.2 Proposed Pattern Classificatin with the Boosting Approach 24

3.2.1 Structure of the Network 24
3.2.2 Learning of Hierarchical Classifier 25
3.2.3 Construction Network 27
3.3 Experiments 27
3.3.1 Simulation Experiments 28
3.3.2 Pattern Classification of Finger Signal Shapes 30
3.4 Concluding Remarks 36

4 Pattern Discrimination considering Unknown Classes 37
4.1 Introduction 37
4.2 Elimination of Unexpected Data 38
4.2.1 Structure 38
4.2.2 Learning Algorithm 38
4.3 EMG Pattern Classification Method 40
4.4 Experiments 41
4.4.1 Numerical Experiments 41
4.4.2 EMG pattern classifications 44
4.5 Concluding Remarks 45

5 Pattern Discrimination using Unsupervised Hierarchical Neural Network 47
5.1 Introduction 47
5.2 Unsupervised Leaning Algorithm of LLGMN 48
5.3 Hierarchical Clustering 49
5.3.1 Classification 49
5.3.2 Division Validation 50
5.3.3 Pruning Law 51
5.3.4 Unsupervised Learning Algorithm 51
5.3.5 Summary of Algorithm 52

5.4 Experiments 53

5.4.1 Numerical Simulation 53

5.4.2 EMG Pattern Classifications 58

5.5 Concluding Remarks 63

6 Human Interface Applications using Biological Signals **65**

6.1 Introduction 65

6.2 Text input system using EMG signals 65

6.2.1 System Description 65

6.2.2 Experiments 70

6.2.3 System Control Experiments 73

6.3 Application of Hand Shape Classification for Human Interface 79

6.4 Concluding Remarks 81

7 Conclusion **83**

7.1 Results and Contributions 83

7.2 Future Works 85

A Log-Linearized Gaussian Mixture Network [21] **89**

List of Figures

2.1 Example of tree structure constructed by the proposed method. 13
2.2 The 31 pattern of hand shape. 15
2.3 Discrimination results for three subjects. 17
2.4 Constructed tree structure for subject C. 18
2.5 Rader charts of hand gesture patterns of gesture 1 and gesture 31 for subject C. The line indicates the mean value of each channel, and the regions of shade imply $\pm$ 1 S.D. 19
2.6 Rader charts of hand gesture patterns of gesture 8 and gesture 9 for subject C. The line indicates the mean value of each channel, and the regions of shade imply $\pm$ 1 S.D. 19
2.7 Discrimination results of finger motions for subject C. 20

3.1 The structure of the proposed method. 25
3.2 The structure of classifier. 26
3.3 Artificial data used for discrimination experiments. 28
3.4 Classification results. 29
3.5 The number of added LLGMN. 30
3.6 The 31 pattern of hand shape. 31
3.7 Shape sensors attached to fingers. 31
3.8 Discrimination results for three subjects. 32

3.9 Discrimination results of finger motion for subject C. 33

3.10 Rader charts of hand gesture pattern of gesture 1 and gesture 31 for subject C. The line indicates the mean value of each channel. 35

3.11 Rader charts of hand gesture pattern of gesture 8 and gesture 9 for subject C. The line indicates the mean value of each channel. 35

4.1 Artificial data used for classification experiments 42

4.2 Classification results 43

4.3 Comparison of elimination results 44

4.4 EMG classification results for five subjects 45

5.1 Example of artificial data, the parameters of which are shown in Table 5.1 54

5.2 The constructed tree for five classes data. 54

5.3 Effect of the number of learning data on estimate accuracy. 55

5.4 Example of artificial data, the parameters of which are shown in Table 5.3 57

5.5 Measured muscles of cervical and expression 58

5.6 An example of cervical spine injury patient's raw EMG signals, filtered signals used for clustering and the force information. 60

5.7 Rader chart of EMG patterns of /i/ and /u/. The line indicates the mean values of each channel, and the regions of shade imply $\pm$ 1 S.D. . 61

5.8 The constructed tree for the EMG data 61

5.9 Classification accuracy 62

6.1 Overview of the text input system 66

6.2 An example of the character selection screen 68

6.3 Definition of the motion time t' 69

6.4 Discrimination results for four subjects. 71

6.5 Discrimination results for Subject D. 72

6.6 Experimental view of the text input system. 74

6.7 Display of the text input system. 74

6.8 An example of the operation with subject A. 75

6.9 An example of the operation with subject A (from 0 sec to 20 sec). . . . 76

6.10 An example of the operation with subject D. 77

6.11 Input time of words. 78

6.12 Opetation of home electrical appliances using discrimination results of hand shape. 79

6.13 An example of the experimental results during the hand shapes of the electric appliances by the subject. 81

A.1 The structure of LLGMN. 90

List of Tables

2.1 Discrimination accuracies of LLGMN, H-LLGMN, and MLPs. 17

3.1 The number of added classifiers and LLGMNs. 34

5.1 Parameters of each class used in experiments of artificial data 53

5.2 The number of classes estimated by the proposed method, the Terashima's method and BIRCH 56

5.3 Parameters of each class used in experiments of artificial data 56

5.4 The number of classes estimated by the proposed method, the Terashima's method and BIRCH 57

5.5 Clustering results 59

6.1 The number of added classifiers and LLGMNs for subject D. 73

6.2 Example of command allocation for home electric appliances. 80

Acknowledgments

I would like to thank my dissertation advisor, Professor Toshio Tsuji, for his support and encouragement over the past seven years. Also, I express my thanks to Professor Tsuji for receiving me in his great laboratory, where I had my fascinating experience. The experiecnces I have learned in his laboratory have changed my life.

I would like to express my profound acknowledgments to Professor Idaku Ishii and Professor Toshio Mikami, who were my committee members, for their invaluable suggestions and discussions on this dissertation.

I want to express my gratitude to Dr. Nan Bu. Without his support, this dissertation could not have been completed. And I am deeply grateful to Professor Yukihiro Matsubara and Professor Noriyuki Iwane for their kind support.

Also, I am deeply grateful to Mr. Keisuke Shima, Ms. Juri Murakami and Mr. Motonobu Shigeto for their frindliness and their advisable suggestions. They were readly to help me to overcome problems in my research.

Finally, I will forever thank to my former research advisor Professor Jousuke Kuroiwa for introducing me to wonderful world of neural networks.

During my Ph.D. research, I was supported in part by the 21st Century COE (Center of Excellence) Program: Hyper Human Technology toward the 21st Century Industrial Revolution.

Chapter 1

Introduction

The theme of this dissertation is the problem of pattern classification, especially hierarchical classification approach for biological signals. Inspired by combining models such as boosting, tree-based models, and conditional mixture models, novel probabilistic neural networks (PNNs) were developed, not only to improve the accuracy of the classification, but also for better estimation of the structure of the classification model.

1.1 Background

In the field of pattern classification, various methods have been applied for image classification, speech recognition, and data mining. This dissertation focuses primarily on neural networks (NNs), first proposed by McCulloch [1]. The field of NNs has its origins in attempts to find mathematical representations of processing in biological systems [1]-[7], however, from the perspective of practical applications of pattern recognition, it is considered that biological realism can impose unnecessary constraints. Therefore, various NNs without biological properties have been proposed. For example, the multilayer network, a type of NN, has proven to be of great practical value [8]. Although single perceptrons can only express a linear decision surface, multilayer networks trained from given data are capable of expressing a variety of nonlinear decision surfaces. NNs have proven to be a promising classification tool because their learning ability allows them to find optimum non-linear relationships between classes, and feature patterns from training data sets. For example, the back propagation algorithm has proven to be surprisingly successful for practical problems [9], such as learning to recognize handwritten characters and spoken words [10], [11]. Although the approximation properties of feed forward networks have been widely studied and found to be applicable, to effectively use NNs as the classifiers for applications, several problems, such as the choice of network structure, learning convergence, and local minima, should be solved [12]-[14].

The PNNs, which estimate the probability density function (pdf) of patterns, have proven to be an efficient and important method of pattern classification [15]-[21]. For realization of the PNNs, the semi parametric estimation approximates the underlying distribution with mixture models that consist of a number of component functions, usually a Gaussian model, since mixture models have a flexible structure that can represent various distributions, and include a set of parameters to specify particular distributions.

In particular, Tsuji et al. [21] proposed a feed forward PNN, a log-linearized Gaussian mixture network (LLGMN) based on the Gaussian mixture model (GMM) and a log-linear model (see Appendix A for details). Although the weights of the LLGMN correspond to a nonlinear combination of GMM parameters, such as mixture coefficients, mean vectors, and covariance matrices, constraints on the parameters in the statistical model are relieved in the LLGMN. Therefore, a simple back propagation learning algorithm can be derived, and the LLGMN parameters are trained according to a criterion of maximum likelihood (ML) [22]. The LLGMN has been successfully applied to pattern classification of bioelectric signals, e.g., electromyograms (EMG) [23] and electrocardiograms (EEG) [21], [24], and has been used to develop human-interface applications, such as in prosthetic devices and EMG-based pointing devices and so on [25]-[32]. Similarly, other classification methods with high classification performance (over 95%) were proposed, and various human-interface using these proposed methods were developed[33]-[35].

Although the LLGMN has higher classification performance than other NNs, LLGMN and LLGMN-based classification methods [36]-[38] suffer from some inherent limitations when they deal with practical signals, such as bioelectric signals.

- To estimate LLGMN parameters, a Gaussian model number of each class must be fixed beforehand. When the Gaussian model number is fixed at an unsuitable value, the LLGMN training cannot avoid convergence at a local minimum for some initial weights and training data. Therefore, better classification performance requires estimation of an optimum LLGMN structure.

- In a training procedure, it is assumed that all data belong to one of the classes corresponding to PNN outputs (see the following equation).

$$P(c|\boldsymbol{x}) \geq 0, \tag{1.1}$$

$$\sum_{c=1}^{C} P(c|\boldsymbol{x}) = 1, \tag{1.2}$$

  where c is a class that categorizes the data. Therefore, to classify data correctly, training data must consist of data generated from all classes. However, in classification for some practical applications it could be impossible to measure the complete data used as training data, because the number of predefined classes for classification is usually smaller than C and there are some data which belong to hidden classes and cannot be prepared beforehand.

- In general, performing pattern classification requires an understanding of relationships between feature vectors (e.g., biological signals) and corresponding class labels (e.g., motions of a measured subject). However, it is difficult to measure signals in the real world without various noises. In addition, in the case of biological signals, the difference between classes can be ambiguous, such as in biological signals of elderly (or handicapped) people, and reliability of available class labels could be questionable. Consequently, classification accuracy may decrease significantly.

1.2 Purpose

This dissertation aims to improve the performance of the PNNs by combining several PNNs, rather than using a PNN in isolation. This dissertation proposes novel PNNs, the core of which is based on the idea of combining models, such as the boosting approach and tree-based models. Moreover, learning algorithms for some proposed PNNs are proposed, in which ambiguous or unlabeled training data can be successfully discriminated by unsupervised learning. The proposed algorithms can eliminate unexpected input signals, those not belonging to a predefined class corresponding to a PNN output, from the classification process. Some methods are also discussed for applying PNNs to human-machine interfaces, using biological signals for improved system performance.

1.3 Related Works

1.3.1 Tree-Based Models

There are various simple, but widely used models that work by partitioning the input space into cuboid regions, whose edges are aligned with the axes, and then assigning a simple model (e.g., a constant, linear classification) to each region [39]. These can be viewed as a model combination method in which only one model is responsible for making predictions at any given point in the input space. The process of selecting a specific model, given the input data, can be described by a sequential decision making process, corresponding to traversal of a binary tree (one that splits into two branches at each node). Classification and regression trees (CART) [39], ID3 [40] and C4.5 [41] are well-developed techniques, which are the major framework of tree-based models. However, there are some problems with tree-based methods using a simple model at each non-terminal node [42]. One problem is that splits are aligned with the axes or linear splits of the feature space. If the dimension of data is large, separating some classes requires a large number of splits of feature spaces, compared to other splitting methods. Sirat et al. proposed a neural tree (NT) using simple NNs (the perceptron of Rosenblatt) as classification models at each non-terminal node of the tree structure [43]. Although a NT can consist of a few classifiers compared to a simple model, classification performance using perceptrons is not high for the classification of complex data, such as biological data.

1.3.2 Boosting Approach

There has also been a growing interest in a boosting approach for the construction of classification systems with simple classifiers [44]-[47]. The performance of a combined classifier is significantly better than that of any of the base classifiers. Adaptive boosting (AdaBoost) is the most widely used form of boosting algorithm. Boosting can yield good results, even if the base classifiers have a performance that is only slightly better than random[48]. Such base classifiers are called weak classifiers. In addition, this approach eliminates the need for evaluating unnecessary models because the algorithm of addition classifiers determines whether to add a classifier or not.

1.3.3 Unsupervised Learning and Estimating the Number of Classes

Some clustering methods have been proposed to identify groups or clusters of data in input space. The K-means algorithm [49] identifies a partition of the input space that optimizes (usually locally) a given clustering error, such as the sum of the squared distance. The self organizing map (SOM) [50], which is one of the major clustering methods, is a high performance clustering method, and can project data from a high-dimensional input space down to two or three dimensions for visualization. Various examples of clustering signals measured in the real world, using an SOM have been reported to show the practicality of SOM. Some of the hierarchical clustering is also a type of clustering method. There are two types of methods for constructing a hierarchical tree: divisive (top down) and agglomerative (bottom up) clustering. However, a problem accompanying the use of a clustering algorithm is the choice of the number of desired output clusters. For clustering data generated from complicated distributions, many of the aforementioned methods fail to make an interpretable and reasonable partition. These methods either partition data from different classes into one class, or classify data from one class into several different classes adversely, even if the true number of classes is known beforehand. To avoid such limitations, and to perform clustering with a sufficient number of classes for complicated data, a variety of clustering algorithms have been proposed. In these methods, in particular, statistical models (e.g., GMM) are assumed to model clusters, and the parameters of the statistical model for each class are estimated correspondingly [51], [53]. A method proposed in literature [51], prepares more models than necessary beforehand, and modifies them until the number of statistical models is identical with the number of desired classes. On the other hand, a method proposed in literature [53] continues increasing the number of models, and stops when the desired (or suitable) number of classes is reached. However, the more complicated the models are, the more parameters are required to be estimated during the clustering process. Thus, much more training data is required. In addition, other methods using SOM have been proposed by Terashima et al. Balanced interactive reducing and clustering using hierarchies (BIRCH) [54], which is a hierarchical clustering method that constructs a hierarchical classification tree using a linear classification model at each non-terminal node. The number of terminal nodes of the constructed tree corresponds to the number of estimated classes. In these methods, this problem is solved by setting some parameters constant, and the learning concentrates on the remaining parameters. Although these methods succeed with problems where the assumption fits the data characteristics well, clustering results cannot always be satisfying for complicated data when a significant difference exists between the true and assumed distributions.

1.3.4 Elimination of Unexpected Data

When classifying using the posterior probabilities of each class, it is assumed that all data belong to predefined classes in order to calculate the posterior probabilities from Bayes' theorem. However, the data not in those predefined classes may exist in the data from the real world. To deal with this problem, one class classification method using a Support Vector Machine (SVM) [55] to eliminate outlier data, has been proposed [56]. In this method, the radial basis function serves as the SVM kernel. Although this method removes outlying data that are different from the given training data, there is no report on multi-class classification using this approach, and it takes a long time to determine a model's fixed parameters through trial and error.

1.4 Outline of Thesis

This dissertation consists of two parts. The first part focuses on a network architecture based on the hierarchical neural tree, and an automatic construction algorithm for the network structure and the PNN learning algorithm used as a classifier for the network. The second part discusses applications of PNNs in the context of a human-machine interface, using biological signals.

The first part comprises Chapters 2 to 5. Chapter 2 introduces a novel hierarchical classification method, called H-LLGMN, which uses a PNN as a classifier at each non-terminal network node. The proposed method automatically constructs a hierarchical tree by combining PNNs from given data, and can achieve a suitable network structure for network validation to improve the generalization ability. Experiments with biological signals prove the feasibility of the proposed method.

In Chapter 3, a pattern classification method with a boosting approach is proposed to achieve high classification performance using a combined weak classifier based on LLGMNs. The network structure and its decision rules are discussed as well. Then, the learning algorithm of the hierarchical classifier simplified H-LLGMN is proposed in Chapter 3. This method can automatically construct a suitable classification network and each hierarchical classifier is based on a boosting approach from given training data. Simulation experiments are performed to compare the proposed method with other classification methods. Finally, pattern classification experiments for biological signals are conducted. These experiments indicate that the proposed method can successfully construct a suitable network for classification based on artificial data and real biological signals.

Chapter 4 shows an improved learning algorithm for pattern hierarchical classification, based on an unsupervised learning algorithm for clustering. First, an unsupervised learning law for the LLGMN is proposed. The construction algorithm using the unsupervised learning law can estimate the number of terminal nodes corresponding to the number of classes according to statistical information obtained only from the training data. Furthermore, unnecessary splits in the classification tree can be avoided with a pruning rule based on a threshold of the ambiguity of the LLGMN outputs and the amount of training data at each non-terminal node. In this method, the classification tree makes binary splits at each non-terminal node. In numerical simulations, the proposed method proves superior to conventional methods in its estimation of the number of classes. Pattern classification experiments for EMG signals are conducted, and indicate that the proposed method is more effective in classifying data with similar features, compared to a traditional supervised learning algorithm.

Chapter 5 proposes a novel pattern classification method using the prior distribution of training data and PNNs, such as the LLGMN. Prior distribution based on the GMM allows the proposed method to remove unnecessary data, not assumed from the training procedure. In addition, the structure of the prior distribution can be automatically estimated from the training data. This chapter adopts the LLGMN for classification. After elimination, the LLGMN classifies input data into predefined classes. This procedure enables the proposed method to avoid classifying unexpected data. The validity of the proposed method is shown with the classification results of artificial data and EMG signals.

Chapter 6 constitutes the second part of this dissertation. It mainly focuses on the development of a new human-machine interface for text input. It also focuses on improving the selection performance of a control system for electric home appliances called a Bio-Remote.

Finally, Chapter 7 concludes this dissertation and gives some challenges and future works.

Chapter 2

A Tree-based Hierarchical Probabilistic Neural Network

2.1 Introduction

In this chapter, a novel hierarchical probabilistic neural network with a tree structure (Hierarchical Log-Linearized Gaussian Mixture Network, H-LLGMN) is proposed. By using LLGMNs as PNNs in partitions at each non-terminal node, the H-LLGMN is expected to discriminate with consideration to individual user variation and reproducibility uncertainty, signals measured from the human body. The generalization ability of the H-LLGMN can be considered for cross-validation when constructing the metaclasses of the hierarchical tree.

This chapter is organized as follows: Section 2.2 introduces the H-LLGMN. Section 2.3 presents the experiments on shape signals (one of the biological signals) of the proposed method. Finally, the last section summarizes the chapter.

2.2 H-LLGMN

In this section, details on the construction algorithm of a hierarchical tree for classification based upon LLGMNs are explained. The structure of the proposed network is shown in Fig. 2.1. During the construction procedure of the classification tree, metaclasses are created from predefined classes that cannot be accurately classified by single LLGMN. Then, by adding LLGMNs for classification of each metaclasse, accurate classification can be performed.

2.2.1 Construction Hierarchical Tree

The classification network starts from a single LLGMN i.e., the root node and a LLGMN is added at non-terminal nodes corresponding to metaclasses when not satisfying the termination criterion. Finally, the training data is accurately classified by performing hierarchical classification. At each level of the classification tree, LLGMNs are used to achieve classification of metaclasses corresponding to non-terminal nodes. Even for data with complicated distributions, a suitable network structure for classification can be estimated after this procedure. The construction algorithm for the hierarchical tree is summarized as follows.

1. A LLGMN is trained using all data.
2. If the classifcation accuracy for the training and validation data is lower than the threshold, metaclasses are created to integrate similar classes.
3. in order to classify data corresponding to metaclasses, LLGMNs are added and traind by the corresponding data.
4. Steps 2 and 3 are repeated until the termination criterion is satisfied for all classes.

By performing these procedures, a suitable network structure for classification of complicated data can be constructed. Next, the determination algorithm for constructing metaclasses based upon the classification results is proposed.

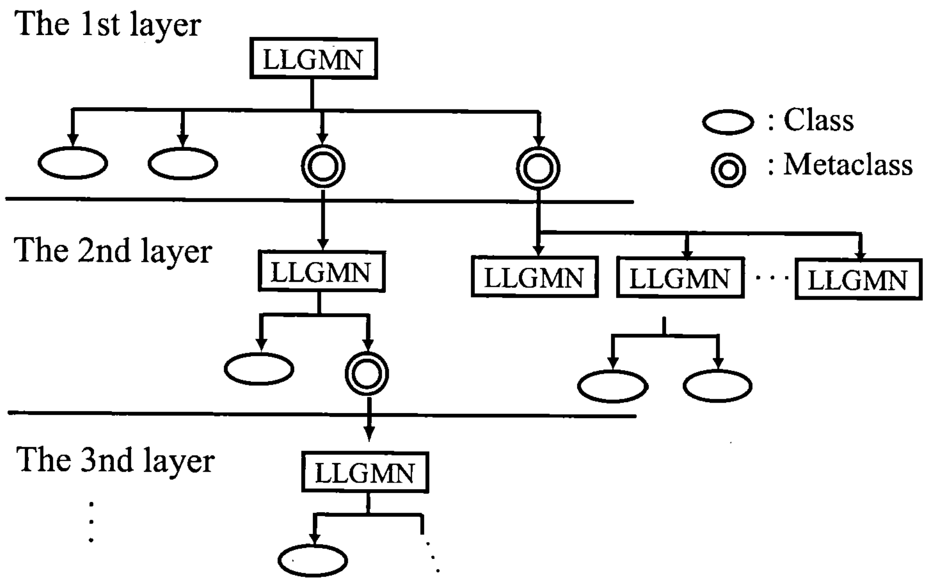


Fig. 2.1: Example of tree structure constructed by the proposed method.

2.2.2 Construction of Metaclass

In order to merge some classes into a metaclass, two criteria using the classification accuracy of the training and validation data are utilized in the proposed method.

First, a set $\mathcal{G}$ is defined as

$$\mathcal{G} = \{C_1, C_2, \ldots, C_K\}, \tag{2.1}$$

where K is the number of classes and $C_i(i = 1, 2, \cdots, K)$ is the set of data belonging to the ith class. A LLGMN is trained using $\mathcal{G}$ as the training data. To evaluate the classification accuracy of the training data, the evaluation function is defined as

$$F_t(i) = \frac{\sum_{i \neq j, j=1}^{K} |D(C_j, C_i)|}{|D(C_i, C_i)|}, \tag{2.2}$$

where $D(C_i)$ is the set of data belonging to class C_i and $D(C_i, C_j)$ is the set of data that belongs to class C_i and is classified into C_j by the LLGMN. $|D(C_j, C_i)|$ is the number of data points for $D(C_j, C_i)$. If $F_t(i)$ is greater than the threshold Th_t, the data classified into C_i is set as the metaclass $MC_i = \cup_{j=1}^{K} D(C_j, C_i)$. This metaclass is added into $\mathcal{G}$ and C_i is removed from $\mathcal{G}$. By merging some classes, which cannnot be accurately classified, into metaclass, a new LLGMN for classifying the metaclass can be added into classification tree.

Although necssary LLGMNs are added to the classification tree based upon inadequate classification of the training data, when the hierarchy of tree grouws too large, there is possibility of learning convergence and local minima because of the decrease in the number of training data at each node. After the classification tree is constructed, the addition of LLGMNs for classification accuracy generalization is conducted by using validation data. An evaluation function that considers the classification accuracy of the validation data is defined as

$$F_v(i, j) = \frac{|D(C_i, C_j)|}{|D(C_i)|}, \tag{2.3}$$

If $F_v(i, j)$ is greater than threshold Th_v, C_i and C_j are merged into a metaclass. This metaclass is then added to $\mathcal{G}$ and C_i and C_j are removed from $\mathcal{G}$. If all $F_v(i, j)$ are smaller than Th_v, this process meets the termination criterion and is stopped.

Through the above criteria, the model construction is performed based on complexity of data and generalization of classification accuracy.

2.3 Hand Shape Classification Experiment

Motion classification experiments using finger-shaped signals were conducted to examine the performance of the proposed method. Three subjects (A, B and C) participated in the experiments.

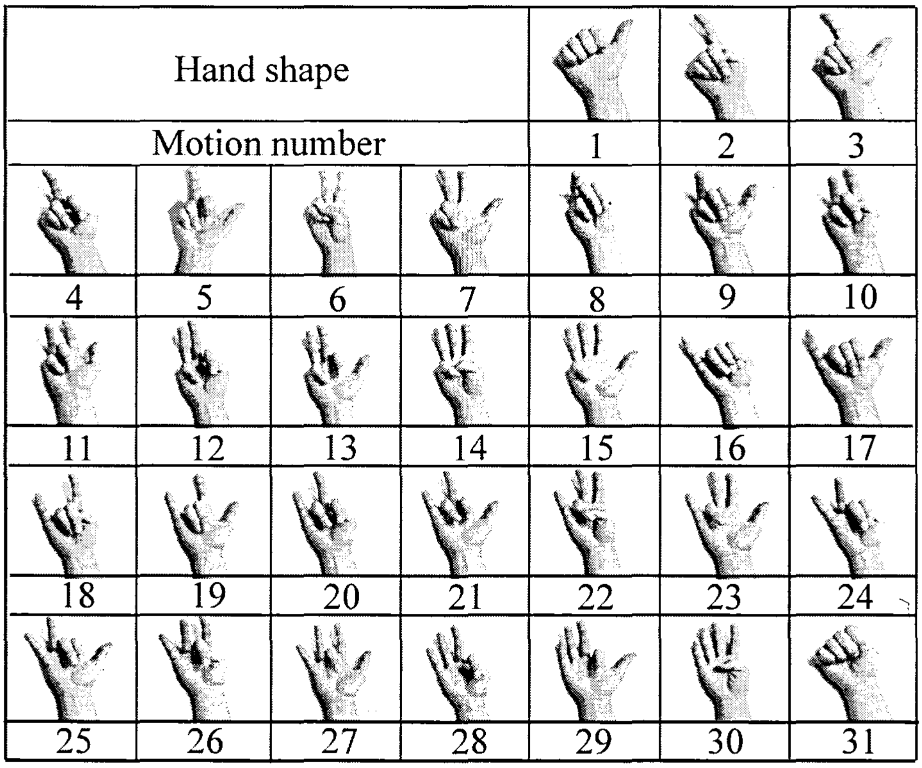


Fig. 2.2: The 31 pattern of hand shape.

The subjects were asked to perform 31 types of motions ($K = 31$) shown in Fig. 2.2.

Five shape signal channels ($L = 5$) were rectified and digitized using an A/D converter (sampling frequency: 167 Hz). Five shape sensors (Measurand Corp.) were attached to each finger of the right hand. These sensors are 1-DOF measuring devices.

One ends of each sensors was fixed to the wrist of the subject, and while the other ends were fixed to the tips of the five fingers. In addition, the sensors were passed through tubes fitted to the fingers for measuring the angles of the fingers. In order to fix the sensors to easy-to-use positions for each subject the exact positions of the sensors were not specified. The measured signals $S_l(n)$ were normalized as follows to obtain a maximum value of 1:

$$N_l(n) = \frac{S_l(n) - S_l^{st}}{S_l^{max} - S_l^{st}}, \tag{2.4}$$

where S_l^{st} is the mean value of $S_d l(n)$ measured when the hand is relaxed, and S_l^{max} is the mean for the maximum value of each channel. The normalized signals were compared with a prefixed threshold M_d to determine whether the subject changed the motion of the hand. In addition, the signals $N_l(n)(l = 1, \cdots, 5)$ are normalized to make the sum of all 5 channels equal to 1 as follows:

$$x_l(n) = \frac{N_l(n)}{\sum_{l=1}^{L} N_l(n)} \tag{2.5}$$

The values of the parameters thresholds for the metaclasses were $Th_t = 1.0$ and $Th_v = 0.5$; each class had 20 training data points and verification data. For each subject, the proposed network is trained by using training data measured from corresponding subject.

The mean values and standard deviations of the classification rates for three independent trials are shown in Fig. 2.3. The number of validation data was 300 samples per class. The number of constructed metaclasses for Subject A, B and C were 0, 2 and 7 respectively.

For the verification of the classification performance of the proposed method, single LLGMN, single MLPs and NT using MLPs based on approach of proposed method were used for the comparison. MLPs had four layers (two hidden layers), the units of which were set as 5, 10, 10 and 31. Table 2.1 shows the classification results by the proposed method and conventional methods. As shown in table, the proposed method (H-LLGMN) can successfully estimate the suitable structure of network and achieves higher discrimination rates than the conventional methods.

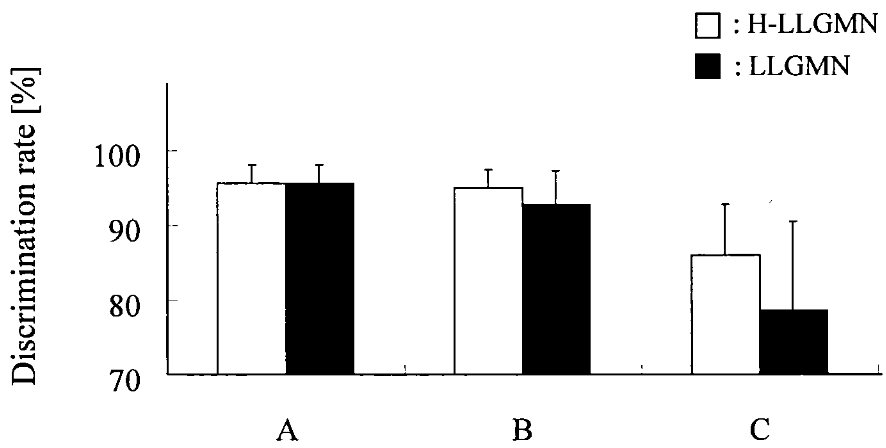


Fig. 2.3: Discrimination results for three subjects.

Table 2.1: Discrimination accuracies of LLGMN, H-LLGMN, and MLPs.

Subjects		A	B	C
Single NN	LLGMN	95.6±2.48	92.7±4.46	78.6±12.0
	MLP	82.1±2.90	52.1±8.36	56.0±9.12
NT	H-LLGMN	95.6±2.48	95.0±2.36	86.0±6.69
	MLP	88.9±3.22	78.6±8.53	73.2±8.17

[%]

Fig. 2.4 illustrates an example of the constructed classification tree for Subject C. Each non-terminal node is labeled according to classes corresponding to metaclasses. In this example, the number of estimated metaclasses is 7 ($\{C_1, C_{31}\}$, $\{C_4, C_5\}$, $\{C_8, C_9\}$, $\{C_{10}, C_{11}\}$, $\{C_{12}, C_{13}\}$, $\{C_{18}, C_{19}\}$ and $\{C_{26}, C_{27}\}$), where C_i is the ith motion. Only the metaclass $\{C_1, C_{31}\}$ is estimated based on the classification results of the training data. The effectiveness of using validation data to construct the classification tree is confirmed from the other metaclasses.

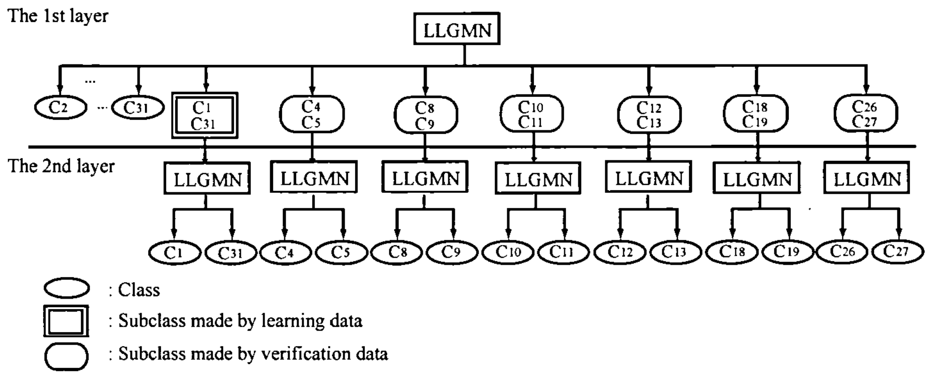


Fig. 2.4: Constructed tree structure for subject C.

Fig. 2.5 shows the mean values and standard deviations of the signal patterns for Motions 1 and 31. This figure shows that the patterns of Motion 1 are similar to those of Motion 31. As a result, a single LLGMN cannot accurately identify the difference between these patterns. However, the proposed method can estimate the distribution of each type of motion accurately. Similarly, the patterns of Motions 8 and 9 overlap (see Fig. 2.6); a more suitable structure can be constructed by the proposed method. Since the distribution of signals belonging to Motion 9 are included to those belonging to Motion 8, a metaclass is constructed due to decreases of classification accuracy of validation data. Fig. 2.7 shows the mean values and standard deviations of the classification rates of each type of motion for Subject C. From this figure, it can be clarified that the classification rate is improved in particular for Motion 5, 13, 19 and 27 by the proposed method.

From these results, it is clear that by adding LLGMNs to a network to estimate the distribution, the proposed method can achieve more accurate classification than a single LLGMN.

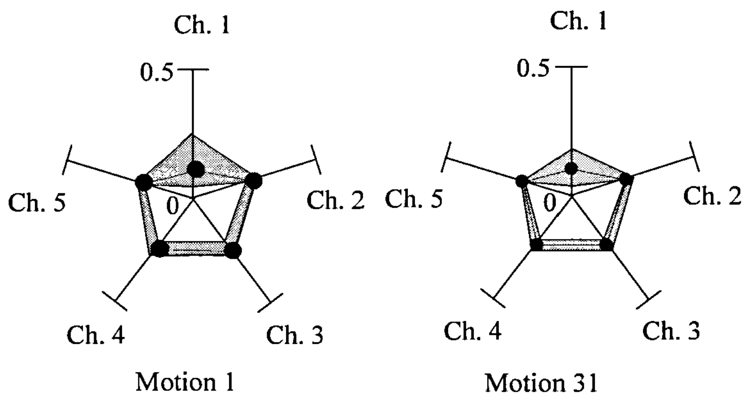


Fig. 2.5: Rader charts of hand gesture patterns of gesture 1 and gesture 31 for subject C. The line indicates the mean value of each channel, and the regions of shade imply ± 1 S.D.

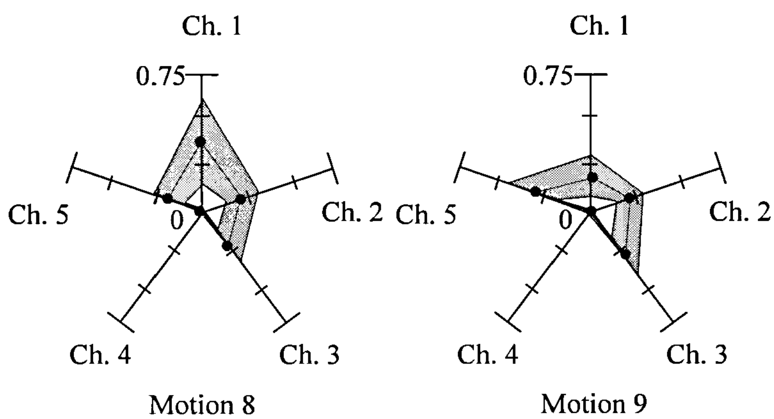


Fig. 2.6: Rader charts of hand gesture patterns of gesture 8 and gesture 9 for subject C. The line indicates the mean value of each channel, and the regions of shade imply ± 1 S.D.

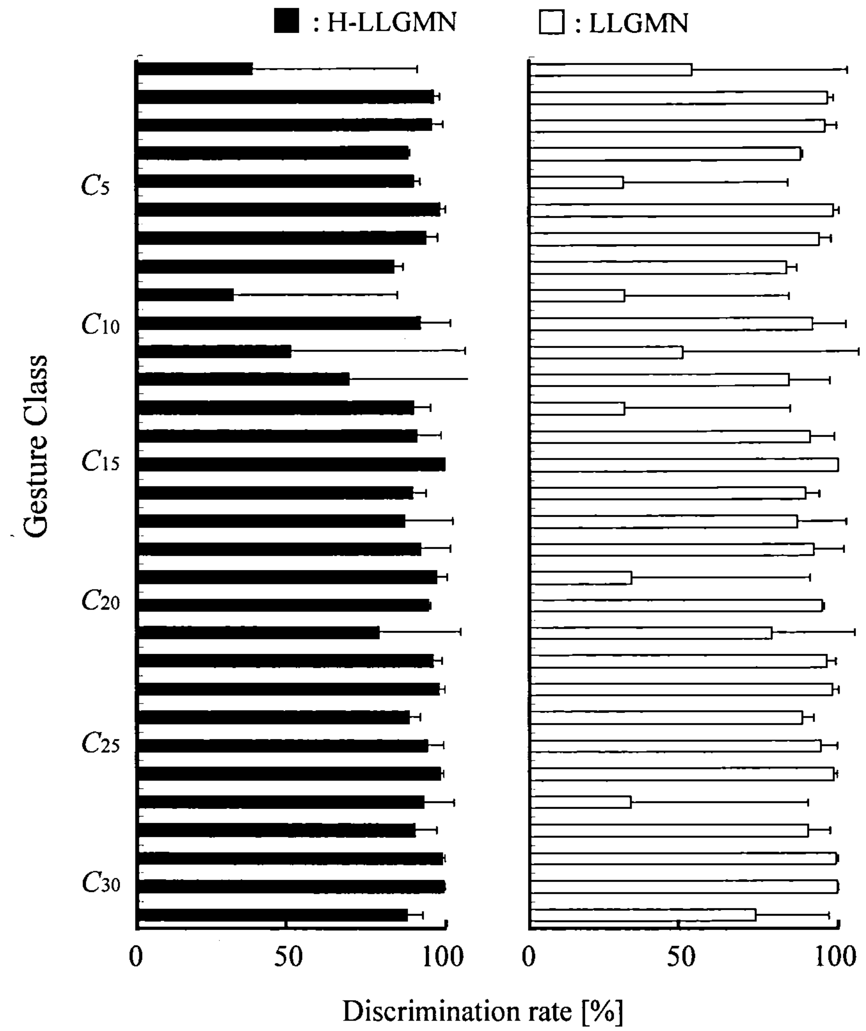


Fig. 2.7: Discrimination results of finger motions for subject C.

2.4 Concluding Remarks

In this chapter, a novel hierarchical probabilistic neural network with a tree structure (H-LLGMN) was proposed in order to enable the discrimination for multiple classes of biological signals. In the proposed method, the structure of the classification network is constructed by adding LLGMNs as classifiers to estimate the distribution of training data. The structure is evaluated based on the classification accuracy of the validation data. Comparison experiments of the proposed medhot and other methods were carried out, and confirmed both the construction of suitable structure and a high classification performance by the proposed method.

Chapter 3

Pattern Discrimination using Probabilistic Neural Networks based on Boosting Algorithms

3.1 Introduction

This chapter proposes a novel hierarchical classification method that can automatically construct classification models through a learning network. In this method, the LL-GMN is utiliszed in order to create a simple and weak classifier. The proposed method can estimate the number of LLGMNs corresponding to the pattern complexity, according to statistical information obtained from the training data.

The next section shows the proposed method for constructing a suitable model using the boosting approach. The results of computer simulation and pattern classification experiments of biological signals are presented in Section 3.3. Finally, the last section concludes this paper.

3.2 Proposed Pattern Classificatin with the Boosting Approach

In the proposed method, the LLGMNs are used in order to create simple classifiers for the classification of input vectors to produce binary splits. Structure of each classifier is a hierarchical tree using LLGMN as each non-terminal node. By combining classifiers based on a boosting approach, the network can discriminate complex data, and calculate a posteriori probability for the training data. The structure of the network and the constructing algorithm are explained below.

3.2.1 Structure of the Network

Initially, the network consists of C classifiers, corresponding to the number of classified classes. C is the number of classes of training data. Each classifier achieves a binary classification to calculate the posteriori probability of the cth class ($c = 1, 2, \cdots, C$). For binary classification, the parameter of LLGMN K is set as 2. $L_c^{(q)}(\boldsymbol{x})(c = 1, \cdots, C, q = 1, \cdots, Q_c)$ is the posteriori probability calculated by classifier, where Q_c is the number of classifiers used for the classification of the cth class added based on boosting approach. Then, the posteriori probability $O_c(\boldsymbol{x})$ is given as

$$O_c(\boldsymbol{x}) = \max_{q=1,\cdots,Q_c} (L_c^{(q)}(\boldsymbol{x})). \tag{3.1}$$

The structure of proposed method is shown in Figure 3.1. The entropy of outputs is also calculated to present the risk of misclassification. The entropy is defined as

$$H(\boldsymbol{x}) = -\sum_{c=1}^{C} O_c(\boldsymbol{x}) \log O_c(\boldsymbol{x}). \tag{3.2}$$

If the entropy $H(\boldsymbol{x})$ is less than the discrimination threshold Te, the class with the largest probability is determined according to Bayes' decision rule (shown in equation

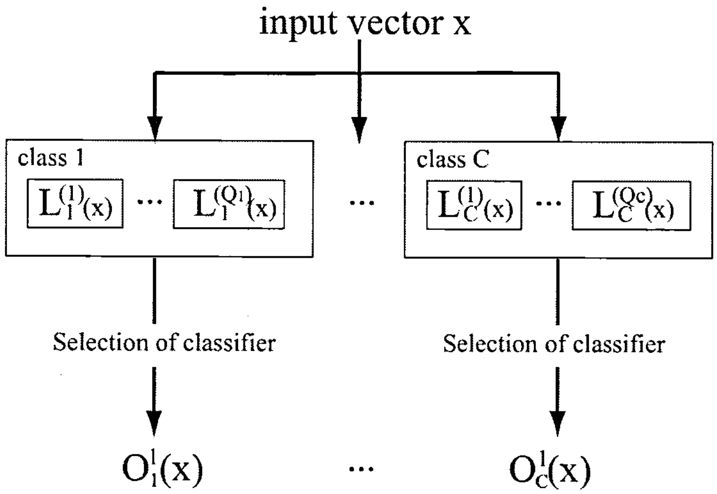


Fig. 3.1: The structure of the proposed method.

3.3). Otherwise, the determination is suspended.

$$Y(\boldsymbol{x}) = \arg\max_c O_c(\boldsymbol{x}). \tag{3.3}$$

3.2.2 Learning of Hierarchical Classifier

Structure of classifier is hierarchical tree using LLGMN. When the learning of the cth class is performed, the training data is divided into two groups, G_c and $G_{\bar{c}}$, where G_c is a set obtained from the training data belonging to class c, and $G_{\bar{c}}$ is the complementary set of G_c. An example of constructed classifier is shown in Fig. 3.2.

Consider a training set $\{\boldsymbol{x}^{(n)}, \boldsymbol{T}^{(n)}\}$ $(n = 1, \cdots, N)$, where $\boldsymbol{T}^{(n)} = (T_1^{(n)}, T_2^{(n)})$. If the input vector $\boldsymbol{x}^{(n)}$ belongs to class c, $T_1^{(n)} = 1$, and $T_2^{(n)} = 0$. An energy function according to the minimum log-likelihood training criterion can be derived as:

$$E = \sum_{n=1}^{N} J^{(n)} = -\sum_{n=1}^{N}\sum_{k=1}^{2} T_k^{(n)} \log^{(3)} O_k. \tag{3.4}$$

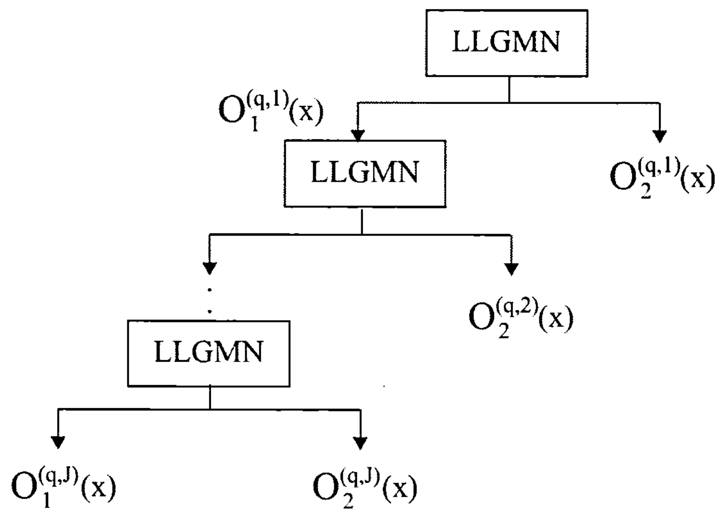


Fig. 3.2: The structure of classifier.

In the training process, modification of the LLGMN's weight $\Delta w_h^{(k,m)}$ is defined as:

$$\Delta w_h^{(k,m)} = -\eta \sum_{n=1}^{N} \frac{\partial J^{(n)}}{\partial w_h^{(k,m)}}, \tag{3.5}$$

$$\begin{aligned} \frac{\partial J^{(n)}}{\partial w_h^{(k,m)}} &= \frac{\partial}{\partial w_h^{(k,m)}}\left(-\sum_{k=1}^{2} T_k^{(n)} \log^{(3)} O_k\right) \\ &= \left({}^{(2)}O_{k,m} - \frac{{}^{(2)}O_{k,m}}{{}^{(3)}O_k} T_k^{(n)}\right) X_h^{(n)}, \end{aligned} \tag{3.6}$$

where $\eta > 0$ is the learning rate.

LLGMNs are added to avoid the misclassification of training data belonging to $G_{\bar{c}}$. To evaluate the misclassification accuracy of training data belonging to $G_{\bar{c}}$, an evaluation function is defined as

$$F' = \frac{|D(\bar{c}, c)|}{|G_{\bar{c}}|}. \tag{3.7}$$

If F' is greater than the threshold Th', more LLGMNs are added hierarchically, and are trained using a two class set $D(c, c)$ and $D(\bar{c}, c)$. Then, the posteriori probability

$L_c^{(q)}(\boldsymbol{x})$, which is calculated by the qth classifier, is defined as,

$$L_c^{(q)}(\boldsymbol{x}) = 1 - \sum_{j=1}^{J_q} \left(\left(\prod_{j'=0}^{j-1} {}^{(3)}O_1^{(q,j')}(\boldsymbol{x}) \right) {}^{(3)}O_2^{(q,j)}(\boldsymbol{x}) \right), \tag{3.8}$$

where J_q is the number of LLGMNs added to the qth classifier, $O_1^{(q,j)}(\boldsymbol{x})$ is the posteriori probability calculated by the jth LLGMN in the qth classifier and $O_1^{(q,0)}(\boldsymbol{x})$ is set to 1. By combining the LLGMN hierarchically to construct a network, the misclassification of data belonging to class c' can be avoided.

3.2.3 Construction Network

In the proposed method, the addition and learning of the classifier is repeated for each class. A classifier is initially trained to classify the training data into G_c and $G_{\bar{c}}$. If $O_1(\boldsymbol{x}) > O_2(\boldsymbol{x})$, it is considered that x is classified into class c. Then, $D(c, \bar{c})$ is the data set belonging to G_c, and is classified into $G_{\bar{c}}$. An evaluation function that considers the training accuracy is defined as

$$F = \frac{|G_c| - |D(c, c)|}{|G_c|}. \tag{3.9}$$

If F is greater than the threshold Th, a classifier is added for accurate discrimination. To train newly added classifier, training data $D(c, \bar{c})$ and $G_{\bar{c}}$ are used. Repeating the addition of classifiers until the evaluation function is less than the threshold Th allows model construction and classifier learning to take place simultaneously.

Through the above training, the model construction and training of the classifier are performed based on a boosting approach.

3.3 Experiments

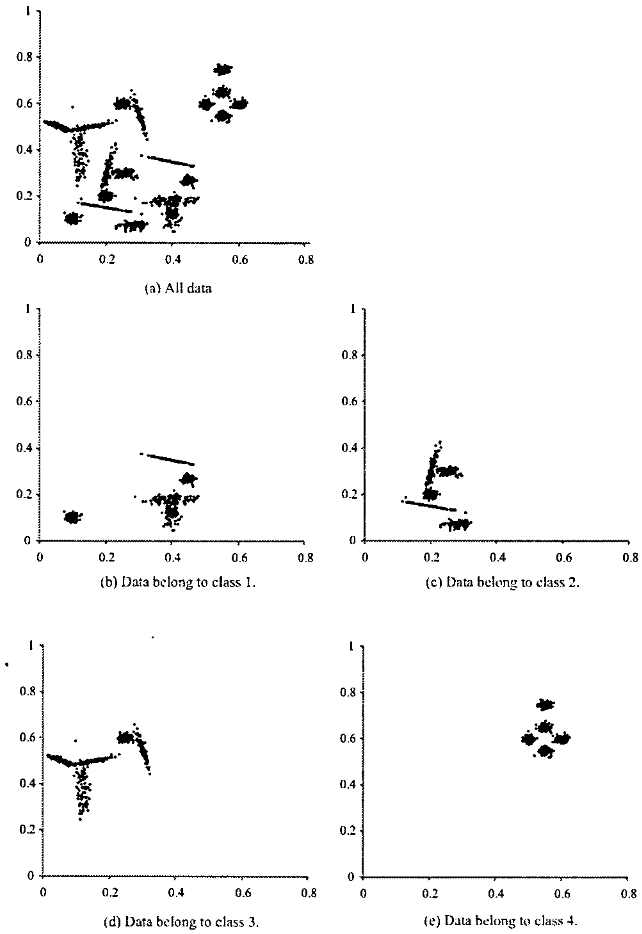


Fig. 3.3: Artificial data used for discrimination experiments.

3.3.1 Simulation Experiments

First, pattern classification experiments on artificial data were conducted for evaluating the performance of the proposed method. A two-dimensional input space consisted of six classes ($C = 6$) ; each class consisted of five Gaussian sources. Examples of the data are shown in Fig. 3.3. For each class, we generated 200 samples to train each LLGMN ($M_k = 1, K = 2$), and then validated the trained network using test data (500 samples/class). The values of the parameters Te, Th and Th' were set as 0.8.

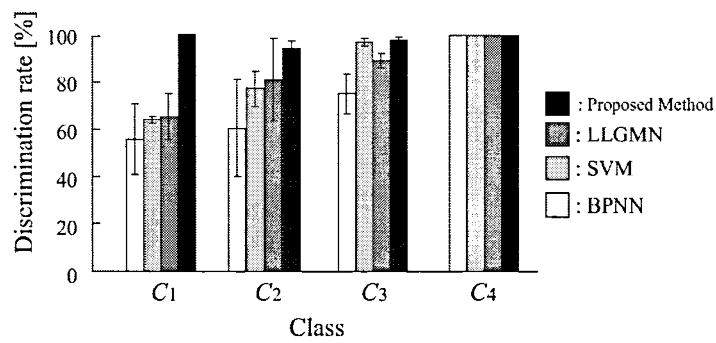


Fig. 3.4: Classification results.

For the verification of the classification performance of the proposed method, single LLGMN, SVM and BPNN classifiers were used for the comparison. BPNN had four layers (two hidden layers), the units of which were set as 2, 10, 10 and 4. Also, a SVM having second-order polynomial kernel was used to perform a two-class classification. By combining two-class classifiers, multi-class classification using SVMs was achieved.

Fig. 3.4 shows the classification results by the proposed method and conventional methods for ten independent trials (the initial weights and training data were chosen at random). The results clearly indicate that the proposed method achieved the best classification rate among all the four methods. The mean values and standard deviations of the number of added LLGMNs for each class are shown in Fig. 3.5. For estimating a simple distribution such as a class six, a single LLGMN was used. On the other hand, many LLGMNs were added to the network for the estimation of complex distributions. These results indicate that the proposed method can estimate successfully the suitable class number of each class, and has the advantage that no unnecessary LLGMNs need to be added while evaluating the discrimination accuracy for determining the network structure.

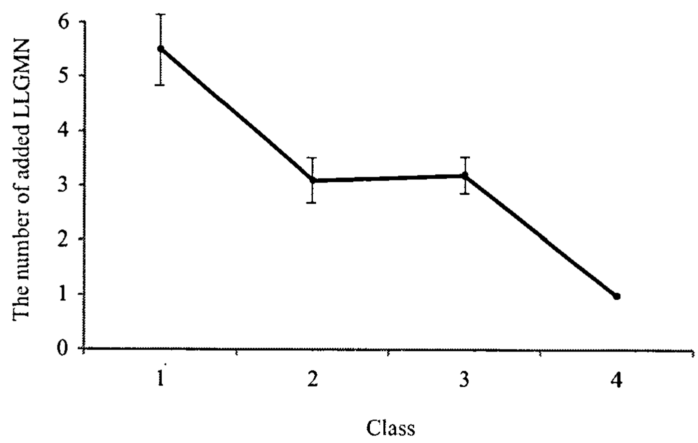


Fig. 3.5: The number of added LLGMN.

3.3.2 Pattern Classification of Finger Signal Shapes

Next, motion classification experiments using finger-shaped signals were conducted for examining the performance of the proposed method. Three subjects (A, B and C) participated in the experiments.

Experimental Conditions

The subjects were asked to perform 31 types of motions ($C = 31$). The motions are shown in Fig. 3.6. Five shape signal channels ($D = 5$) were rectified and digitised using an A/D converter (sampling frequency: 167 Hz). Five shape sensors (Measurand Corp.) were attached to each finger of the right hand. These sensors are 1-DOF measuring devices. The attached sensors are shown in Fig. 3.7. One ends of the sensors were fixed to the wrist of the subject, and the other ends were fixed to the corresponding tips of fingers. Also, for measuring the angle of the finger, the sensors were passed through the tubes that were fitted to the fingers (see Fig. 3.7). In order to fix the sensors to easy-to-use positions for each subject the exact positions of sensors were not specified.

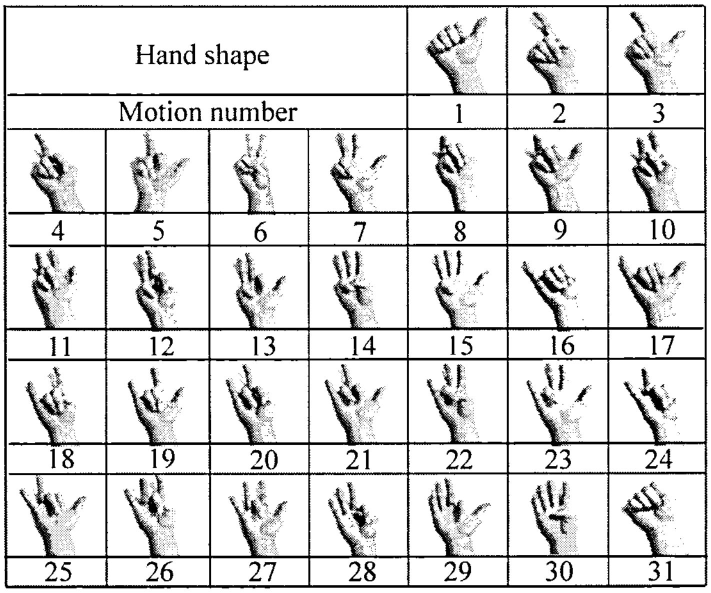


Fig. 3.6: The 31 pattern of hand shape.

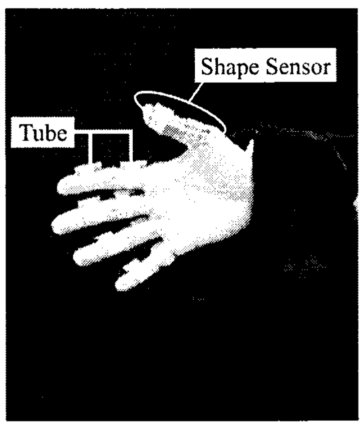


Fig. 3.7: Shape sensors attached to fingers.

The measured signals $S_d(n)$ were normalized as follows for obtaining a maximum value of 1:

$$N_d(n) = \frac{S_d(n) - S_d^{st}}{S_d^{max} - S_d^{st}}, \tag{3.10}$$

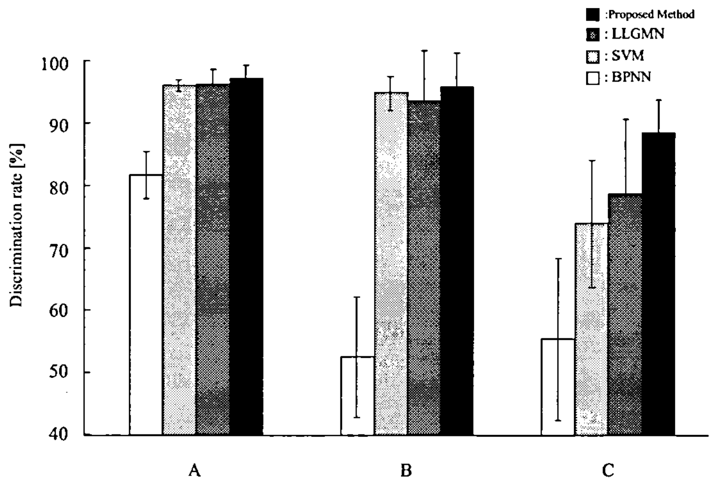


Fig. 3.8: Discrimination results for three subjects.

where S_d^{st} is the mean value of $S_d(n)$ measured when the hand is relaxed, and S_d^{max} is the mean of the maximum value of each channel. The normalized signals were compared with a prefixed threshold M_d to determine whether the subject changed the motion of the hand. In addition, signals $N_d(n)(d = 1, \cdots, 5)$ are normalized to make the sum of all D channels equal to 1 as follows:

$$s_d(n) = \frac{N_d(n)}{\sum_{d=1}^{D} N_d(n)}. \tag{3.11}$$

The values of the parameters Th and Th' were set as 0.8 and M_d was set as 0.5.

In this experiment, the shape signals measured beforehand were selected using our proposed method. For each subject, the proposed network is trained by using training data measured from corresponding subject.

Pattern Classification Results

The mean values and standard deviations of the classification rates are shown in Fig. 3.8. BPNN had four layers (two hidden layers), the units which were set as 5, 10, 10

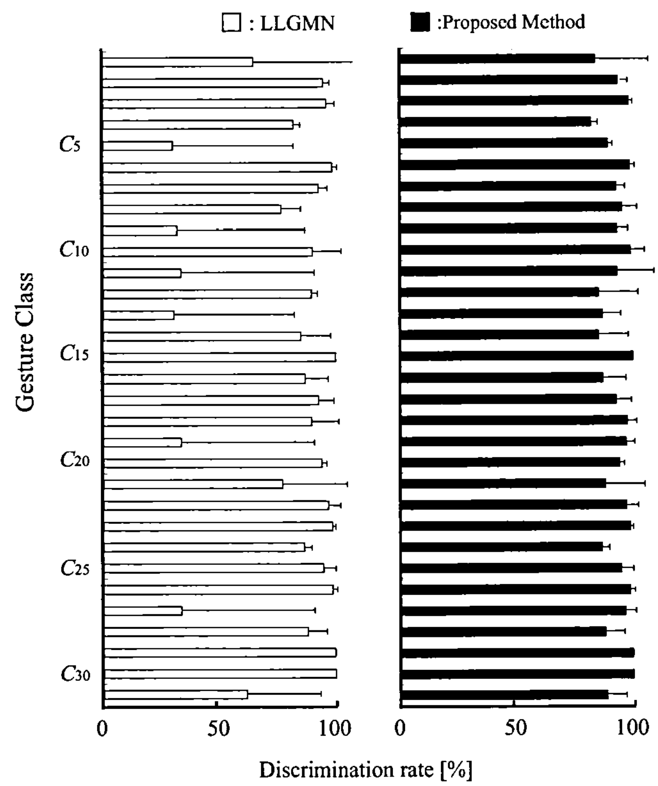


Fig. 3.9: Discrimination results of finger motion for subject C.

and 32. Moreover, 32 SVMs were used for the classification. As shown in the figure, the classification results of the proposed method are similar to those of SVM and single LLGMN for the case of Subjects A and B. In the case of Subject C, however, the classification results of other methods degrade more than that of the proposed method.

Table 3.1 shows an example of the number of added classifiers and LLGMNs in the network of Subject C. Here, we infer that a better classification is achieved by adding the classifiers and LLGMNs.

Fig. 3.9 shows the mean values and standard deviations of the classification rates of each type of motion for Subject C. From this figure, it can be clarified that the classification rate has improved overall by the proposed method. For example, Fig. 3.10 shows the mean values and standard deviations of the signal patterns of Motions

Table 3.1: The number of added classifiers and LLGMNs.

class	classifiers	LLGMNs
1	2	5
2	1	1
3	1	1
4	3	3
5	2	2
6	1	1
7	1	1
8	4	4
9	2	6
10	3	3
11	2	4
12	2	2
13	2	3
14	1	1
15	1	1
16	1	1
17	1	1
18	1	1
19	2	6
20	1	1
21	1	1
22	1	1
23	1	1
24	1	1
25	1	1
26	2	4
27	3	4
28	1	1
29	1	1
30	1	1
31	2	7

1 and 31. This figure shows that the patterns of Motion 1 are similar to those of Motion 31. As a result, a single LLGMN cannot accurately identify the difference between these patterns. However, the proposed method can estimate the distribution

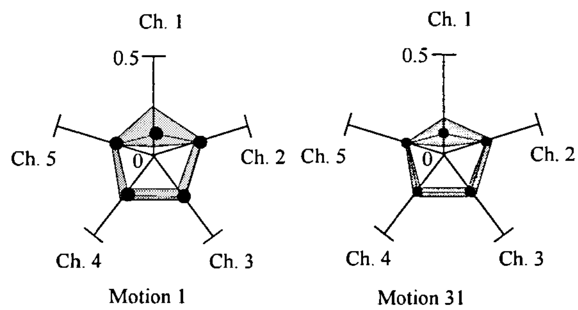


Fig. 3.10: Rader charts of hand gesture pattern of gesture 1 and gesture 31 for subject C. The line indicates the mean value of each channel.

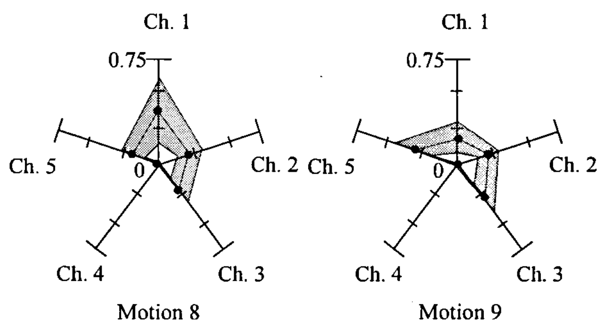


Fig. 3.11: Rader charts of hand gesture pattern of gesture 8 and gesture 9 for subject C. The line indicates the mean value of each channel.

of each type of motion accurately using more than one LLGMN. For example, the patterns of Motions 8 and 9 overlap (see Fig. 3.11); a more suitable structure can be constructed by the proposed method by combining the LLGMNs. From these results, it is clear that by adding LLGMNs to a network for the estimation of the distribution, the proposed method can achieve a more accurate classification than a single LLGMN.

3.4 Concluding Remarks

In this chapter, a novel hierarchical probabilistic neural network based on a boosting approach is proposed.

In the proposed method, the structure of the classification network is constructed by adding LLGMNs as classifiers to estimate the distribution of training data. By evaluating the structure based on classification accuracy, the addition of unnecessary LLGMNs can be avoided.

Experimental results on the artificial dataset and hand shape signals prove the feasibility of the proposed method. Comparison experiments of the proposed method and single LLGMN were conducted, and the high classification performance of the proposed method was confirmed. It has been shown that the proposed method is suitable for classification of complex data, since the required classifiers will automatically be added in the network in order to perform an accurate classification.

Chapter 4

Pattern Discrimination considering Unknown Classes

4.1 Introduction

This chapter proposes a new pattern classification method using prior probability of EMG signals. In this method, estimated prior probability based on GMM is utilized for elimination of unexpected data. Moreover, the structure of prior distribution for data can be automatically estimated through a training procedure. After elimination, LLGMN can classify data into predefined classes. This procedure enables the proposed method to avoid the classification of unexpected data.

The rest of this chapter is organized as follows. Section 4.2 proposed the details of the method of elimination of unexpected data and learning algorithm of the proposed structure. In Section 4.3, the EMG pattern classification method using LLGMN is provided. The results of computer simulation and phoneme pattern classification experiments of EMG signals are presented in Section 4.4. Finally, the last section concludes this chapter.

4.2 Elimination of Unexpected Data

In the proposed method, the GMM are used in order to remove unnecessary data not belonging to predefined classes. The structure of GMM consists of some Gaussian distribution component. By estimating the number of components automatically, suitable structure for elimination can be constructed. The structure of the network and the constructing algorithm are explained below.

4.2.1 Structure

The prior probability $F(\boldsymbol{x})$ is given as

$$F(\boldsymbol{x}) = \sum_{m=1}^{M} \alpha_m g(\boldsymbol{x}, \boldsymbol{\mu}_m, \delta_m^2 \boldsymbol{E}), \tag{4.1}$$

where $\boldsymbol{x}$ is inputted data and M denotes the number of components, α_m is the mixture coefficient for component m, and $g(\boldsymbol{x}, \boldsymbol{\mu}_m, \delta_m^2 \boldsymbol{E})$ is a Gaussian distribution with mean vector $\boldsymbol{\mu}_m$and covariance matrix $\delta_m^2 \boldsymbol{E}$. $\boldsymbol{E}$ is the identity matrix. If $F(\boldsymbol{x})$ is greater than the threshold Tp, the data is classified into predefined classes by LLGMN. Otherwise, the classification is suspended.

4.2.2 Learning Algorithm

The proposed method can automatically estimate the suitable number of components corresponding to the complexity of training data. In the training procedure, a set of vectors $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N)$ are utilized. The details of the proposed training scheme is as follows:

Step 1 Initialization:

1. Set the number of components M as 1 and the termination threshold $\delta^{(p)}$ as any given real number.

2. Initialize the mean vector $\boldsymbol{\mu}_1$ with randomized values, and then set δ_1^2 as the maximum value of the ii element of Σ calculated from the following equation and α_1 as 1.

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu}_1)(\boldsymbol{x}_n - \boldsymbol{\mu}_1)^T \tag{4.2}$$

Step 2 Update the mean vector:

1. Set the training iteration t as 1.

2. Update the mean vectors according to the following equations for all training data [50].

$$\Delta\mu_{m'}(\boldsymbol{x}_n) = (1 - \frac{t}{T})(\boldsymbol{x}_n - \boldsymbol{\mu}_{m'}) \tag{4.3}$$

$$m' = \arg\max_{m} g(\boldsymbol{x}_n, \boldsymbol{\mu}_m, \delta_m^2 \boldsymbol{E}) \tag{4.4}$$

where T is the predefined maximum iteration number and $m = 1, \cdots, M$.

3. Classify training data into M groups decided from equation 4.4.

4. Compute δ_m^2 and the mixture coefficient α_m according to equations 4.5 and 4.6 [57].

$$\delta_m^2 = \max_i \delta_{ii}^{(m)}, \tag{4.5}$$

$$\alpha_m = \frac{||G_m||}{N}, \tag{4.6}$$

where $\delta_{ij}^{(m)}$ is the ij element of Σ_m (see equation 4.7), G_m is the set of data clasified into group m and $||G_m||$ is the number of data belonging to group G_m.

$$\Sigma_m = \frac{\sum_{\boldsymbol{x}_n \in G_m}(\boldsymbol{x}_n - \boldsymbol{\mu}_m)(\boldsymbol{x}_n - \boldsymbol{\mu}_m)^T}{||G_m||} \tag{4.7}$$

5. This step of training repeats, until t reaches a predefined number T.

Step 3 Addition of component:

1. Stop the training procedure, if $\delta_m^2 > \delta^{(p)} - \epsilon$ and $M > 1$, where ϵ is a small positive number.

2. Otherwise, a component is added. $\delta^{(p)}$ is set as the maxmum variance δ_m^2 for next validation. Then, the mean vectors of the added component is initialized randomly and δ_m^2 and α_m of all components are calculated according to Equations (5) and (6). Then, go to Step 2.

4.3 EMG Pattern Classification Method

The proposed EMG pattern classification consists of three parts: (1) EMG feature extraction, (2) elimination of unexpected EMG signals and (3) classification network (LLGMN).

L channels of EMG signals are recorded using surface electrodes attached to muscles. The EMG signals are measured with a sampling frequency $f = 1000$Hz, then rectified and filtered by a Butterworth filter (cutoff frequency: 1Hz). Each sampled EMG pattern, defined as $\boldsymbol{EMG}(t)$ was normalized to make the sum of L channels equal to 1 using the following equation,

$$x_l^{(t)} = \frac{EMG_l(t) - EMG_l^{st}}{\sum_{l'=1}^{L}(EMG_{l'}(t) - EMG_{l'}^{st})}, \tag{4.8}$$

where EMG_l^{st} is the mean value of $EMG_l(t)$ measured while relaxing the muscles. The feature vectors $\boldsymbol{x}(t) = [x_1(t), x_2(t), \cdots, x_L(t)]$ are inputted into classification network. A power level is estimated from the EMG signals as

$$power(t) = \frac{1}{L}\sum_{l=1}^{L}\frac{EMG_l(t) - EMG_l^{st}}{EMG_l^{max} - EMG_l^{st}}, \tag{4.9}$$

where EMG_l^{max} is the mean value of $EMG_l(t)$ measured under the maximum voluntary contraction. The power level is compared with a prefixed threshold M_d to determine whether the motion actually happened.

For elimination of unexpected EMG signals, a prior probability calculated from GMM described in Section 4.2 is employed. EMG signals corresponding to predefined classes are classified by LLGMN. The output of the LLGMN corresponds to the posterior probability $P(k|\boldsymbol{x}(t))$ of class k given the input vector $\boldsymbol{x}(t)$. The entropy of outputs is also calculated to prevent risk of misclassification. The entropy is defined as

$$H(\boldsymbol{x}(t)) = -\sum_{k=1}^{K} {}^{(3)}O_k(t) \log^{(3)} O_k(t). \tag{4.10}$$

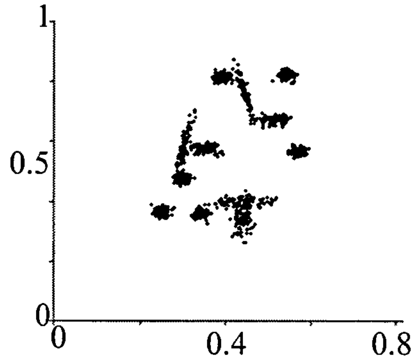
If the entropy H($\boldsymbol{x}$(t)) is less than the classification threshold Te, the specific motion with the largest probability is determined according to the Bayes ' decision rule. Otherwise, the determination is suspended.

$$Y(\boldsymbol{x}(t)) = \arg\max_k {}^{(3)}O_k(t). \tag{4.11}$$

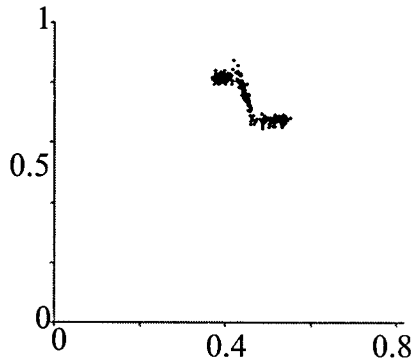
4.4 Experiments

4.4.1 Numerical Experiments

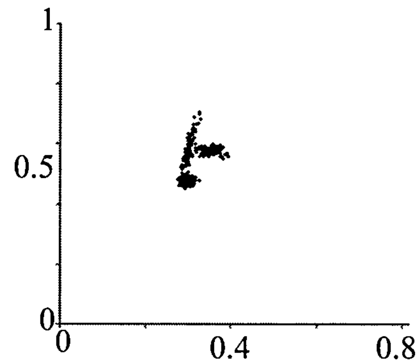
First, pattern classification experiments on artificial data were conducted for evaluating the performance of the proposed method. A two-dimensional input space consisted of three classes ($K = 3$) and predefined class, each class and predefined class consisting of three components. Examples of the data are shown in Fig. 4.1. For each class, 200 samples were generated to train, and then the trained network was validated using test



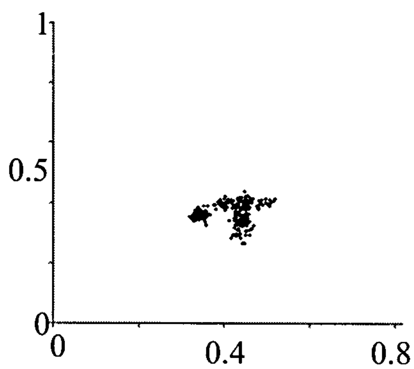
(a) All data



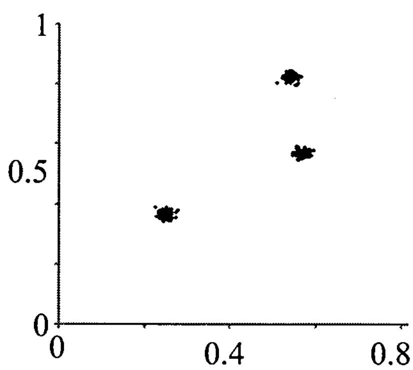
(b) Data belonging to class 1



(c) Data belonging to class 2



(d) Data belonging to class 3



(e) Data not belonging to predefined classes.

Fig. 4.1: Artificial data used for classification experiments

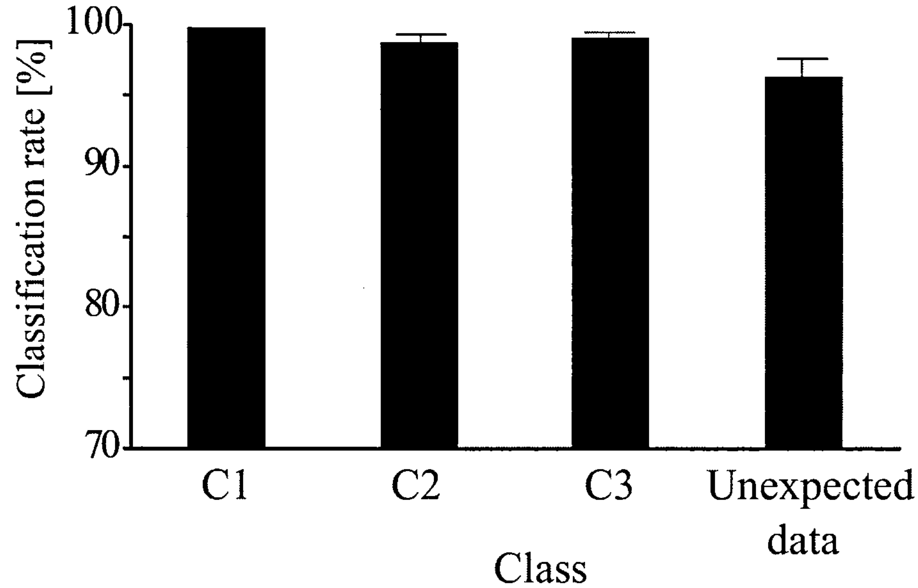


Fig. 4.2: Classification results

data (500 samples/class). The values of the parameters were $Tp = 0.01, \epsilon = 0.01$ and $te = 0.8$.

Fig. 4.2 shows the classification results by the proposed method for 10 independent trials (the initial mean vectors and weights of LLGMN were chosen at random). The results indicate that the proposed method achieved the elimination of unexpected data and high classification performance.

In order to confirm the effectiveness of automatic addition of components, GMMs that fixed the number of components were used for comparison. In this experiments, test data for validation were divided into the data belonging to predefined classes and the unexpected data by the proposed method and the traditional GMMs (the number of components are 5, 10, 15). Fig. 4.3 shows the elimination results by the proposed method and the other methods using fixed GMMs. The mean value of the number of components for GMM was 17.6 ± 3.0. These results indicate that the proposed method can estimate successfully the number of components even for unexpected data

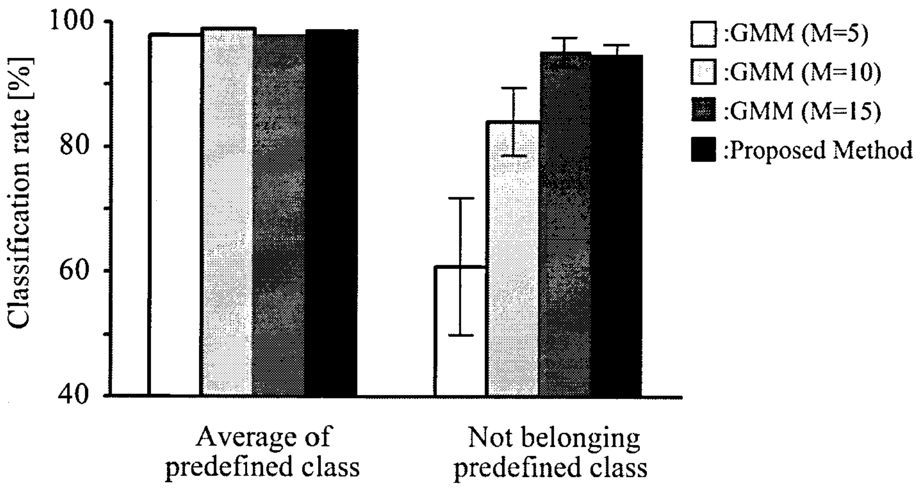


Fig. 4.3: Comparison of elimination results

not belonging to the predefined classes.

4.4.2 EMG pattern classifications

Phoneme classification based on EMG signals was conducted to examine performance of the proposed method. In the experiments, EMG signals measured from mimetic and cervical muscle were used to classify six Japanese phonemes (/a/, /i/, /u/, /e/, /o/, /n/). In this experiment, classes corresponding to five phonemes (/a/, /i/, /u/, /o/, /n/) were used as predefined classes and EMG signals belonging to utterance /e/ were set as unexpected EMG signals. Five subjects (A, B, C, D and E) participated in the experiments.

Five pairs of Ag/AgCl electrodes (NT-511G: NIHON KOHDEN Corp.) were attached to the subject' s face (Depressor Anguli Oris, Zygomaticus Major, Masseter, Digastric, Depressor Labii Inferioris; a pair of electrodes on each muscle) with conductive paste. The EMG signals from five muscles were recorded (sampling frequency: 1kHz). The values of the parameters were $Tp = 0.01, \epsilon = 0.01, te = 0.8$ and $M_d = 0.25$.

Five sets of randomly chosen initial mean vectors and weights were used to train

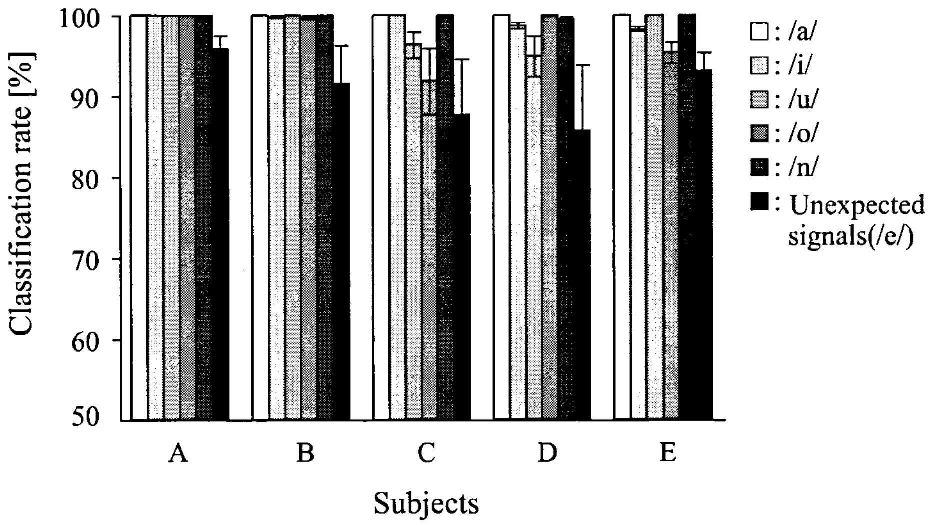


Fig. 4.4: EMG classification results for five subjects

each sample data. For each subject, the proposed network is trained by using training data measured from corresponding subject. The mean values and standard deviations of the classification rates are shown in Fig. 4.4. From this figure, it can be seen that the elimination of unexpected EMG signals and the classification accuracy of EMG signals belonging to predefined classes are achieved by using the prior probability of EMG signals and LLGMN for classification.

4.5 Concluding Remarks

In this Chapter, in order to deal with the classification problem with ambiguous teacher signals, a hierarchical clustering has been proposed. In this method, a LLGMN which a probabilistic NN derived from the GMM, is used as non-terminal node in the classification tree.

Entropy of the LLGMN ' s outputs and the data number at each node are used as stopping and pruning indices in the proposed method, and unnecessary splits in the structure of classification tree can be avoided, so that the proposed method can make a interpretable and reasonable partition of the training data according solely to its statistical characteristics.

In the numerical simulations, the proposed method shows superior to the conventional method in the estimation of the number of classes. And from the results of EMG pattern classification experiments, it is considered that the proposed method is more effective in classification data with the similar features, comparing with a traditional method where the LLGMN is trained using a supervised training algorithm.

Chapter 5

Pattern Discrimination using Unsupervised Hierarchical Neural Network

5.1 Introduction

In this chapter, a novel hierarchical clustering method is proposed. In this method, the GMM is used to model the statistical characteristics of feature vectors with no restriction assumed on its parameters, and a probabilistic NN derived from the GMM, called log-linearized Gaussian mixture network (LLGMN), is utilized for partition at each non-terminal node. In addition, this paper proposes an unsupervised learning law for the LLGMN. The proposed method can estimate the number of terminal nodes corresponding to the number of classes according to statistical information obtained solely from the training data. Furthermore, unnecessary splits in the classification tree can be avoided with a pruning rule based on a threshold of the ambiguity of the LLGMN ' s outputs and the number of data at each non-terminal node. In this paper, the classification tree makes binary splits at each non-terminal node.

In the following section, we propose an unsupervised learning law for the LLGMN. In Section 6.3, the algorithm for constructing the classification tree is introduced.

Section 6.4 presents experiments on artificial data and the EMG signals to examine the validity of the proposed method. Finally, Section 6.5 gives a summary of the chapter.

5.2 Unsupervised Leaning Algorithm of LLGMN

Although the LLGMN can be trained with a supervised learning algorithm as mentioned, its classification performance degrades significantly, when the teacher signals are not reliable. However, there have been no unsupervised learning algorithms developed for LLGMN so far. In this chapter, an unsupervised learning algorithm based on the entropy criterion is introduced. Given the number of classes C, the entropy is defined as:

$$J_{SO} = -\sum_{n=1}^{n}\sum_{c=1}^{C} {}^{(3)}O_c^{(n)} \log {}^{(3)}O_c^{(n)}. \tag{5.1}$$

The proposed unsupervised learning algorithm seeks to find proper parameters of LLGMN by minimizing Equation (9). The weight ' s modification $\Delta w_h^{(c,m)}$ is defined as:

$$\begin{aligned}
\Delta w_h^{(c,m)} &= -\eta \frac{\partial J_{SO}^{(n)}}{\partial w_h^{(c,m)}}, \qquad (5.2)\\
\frac{\partial J_{SO}^{(n)}}{\partial w_h^{(c,m)}} &= \frac{\partial}{\partial w_h^{(c,m)}}\left(-\sum_{n=1}^{N}\sum_{c=1}^{C} {}^{(3)}O_c^{(n)} log {}^{(3)}O_c^{(n)}\right)\\
&= -\sum_{n=1}^{N}\sum_{c'=1}^{C} \frac{\partial}{\partial {}^{(3)}O_{c'}^{(n)}} {}^{(3)}O_{c'}^{(n)} log {}^{(3)}O_{c'}^{(n)}\\
&\quad \times \sum_{m'=1}^{M_{c'}} \frac{\partial {}^{(3)}O_{c'}^{(n)}}{\partial {}^{(2)}O_{c',m'}^{(n)}} \frac{\partial {}^{(2)}O_{c',m'}^{(n)}}{\partial {}^{(2)}I_{c,m}^{(n)}} \frac{\partial {}^{(2)}I_{c,m}^{(n)}}{\partial w_h^{(c,m)}}\\
&= -(J_{SO} - log {}^{(3)}O_c^{(n)} {}^{(2)}O_{c,m}^{(n)} X_h^{(n)}. \qquad (5.3)
\end{aligned}$$

After the LLGMN is well trained, a reasonable C-class partition can be performed on the training data. However, for some ill-posed initial weights, the LLGMN may be trained to cluster all training data into one class and the energy function J_{SO}, converge to some local minimum. Some methods, such as the k-means clustering, can avoid this problem by setting mean vectors derived from the training data as initial weights. But this method can not be applied to the LLGMN, since parameters such as the mean vectors and variances of each cluster (Gaussian distribution) are not directly used as the weights of LLGMN. Since the proposed learning algorithm needs prior information about the number of classes C, the unsupervised learning law based on equation 5.1-5.3 is not practical.

To deal with these problems, we propose a hierarchical clustering method using LLGMN at each non-terminal node of a classification tree. In this method, the LLGMN classifies data into two-subclasses at each non-terminal node; after a sequence of binary splits, the training data would be eventually partitioned into classes of any desired number, corresponding to the number of terminal nodes. In the next section, a tree construction algorithm and an unsupervised learning law to train the LLGMN are explained.

5.3 Hierarchical Clustering

In this section, details of the construction strategy for a binary classification tree based on LLGMNs are explained. During the construction process, split or pruning is determined according to the statistical properties of the training data. Adopting a LLGMN at a non-terminal nodes of the classification tree complements the unsupervised learning algorithm of the LLGMN introduced in Section 5.2.

5.3.1 Classification

The divisive clustering starts from a single cluster i.e., the root node and terminates when satisfying the termination criterion. The training data is then divided into the objective number of clusters. At each level of the classification tree, LLGMNs are used to achieve binary splits, and each non-terminal node is divided into two sub-clusters if significant statistical differences exist between the two parts. Even for data with complicated distributions, interpretable clustering can be perform after a nested series of binary splits.

5.3.2 Division Validation

By using the proposed method, a classification tree can be constructed. However, excessive splitting may occur when the hierarchy of the tree becomes too deep. To counteract this issue, cross-validation is adopted and the posterior probabilities of the validation data are utilized to determine whether or not to split a node. First, the validation data is prepared and the entropy $H(\boldsymbol{x})$ is defined as:

$$H(\boldsymbol{x}) = -\sum_{c=1}^{C} {}^{(3)}O_c^{(n)} log {}^{(3)}O_c^{(n)}. \tag{5.4}$$

Then, the assembled average value H_E of $H(\boldsymbol{x})$ is utilized as the termination criterion of splitting.

$$H_E = -\frac{1}{|N_c|} \sum_{\boldsymbol{x}^{(n)} \in N_c} H(\boldsymbol{x}^{(n)}), \tag{5.5}$$

where N_c is the set of validation data belonging to the node in consideration, and $|N_c|$ is the number of validation data in N_c. If H_E is higher than a threshold H_T, the splitting of the corresponding node is terminated. As a result, excessive splits can be avoided. On the other hand, if all validation data for the node in consideration is clustered into one class, outliers may exist in the training data and the division of this node must be terminated. Also, for occasions when there is only one training data in a node, further splits of this node must be terminated, as division is impossible.

With this method, the classification tree can be constructed based on the statistical properties of the training data, and can cluster complicated data into an interpretable number of classes.

5.3.3 Pruning Law

In the proposed method, outliers are always classified into some terminal nodes (clusters) separated from other major clusters. In particular, when the hierarchy of the tree grows too large, the influence of outliers becomes prominent because of the decrease of the number of training data at each node. After the classification tree is constructed, pruning is conducted to improve the clustering efficiency. The number of training data left in each terminal node is utilized as a decision index for pruning. If the ratio of the number of training data in a terminal node to the total training data number is lower than a threshold α_T, this node and its counter are merged into their father node. With this pruning law, excessive splits may be prevented, and the number of clusters may not increase corresponding to the number of outlier data.

5.3.4 Unsupervised Learning Algorithm

As mentioned in Section 5.2, correct clustering is not available if ill-posed initial weights of LLGMN are used. Here, we design the unsupervised learning rule used in the proposed method, where the number of clusters is restricted to two. In this rule, the initialization of the weights is made with two data selected from the total training data set, noted as $\boldsymbol{A}$. During the training process, the rest data is gradually added into the data set $\boldsymbol{B}$ used for training to proceed clustering. In this way, clustering starts with a data set of simple distribution, with the increment of the data number used for training, clustering result turns to be complicated.

Let us consider that LLGMN clusters data into two classes: C_1 and C_2. First, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are chosen for the initialization of weights from the set $\boldsymbol{A}$ according to the following equation,

$$(\boldsymbol{x}_1, \boldsymbol{x}_2) = \arg\max_{\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \in \boldsymbol{A}} (||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}||). \tag{5.6}$$

Assuming that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are labeled with C_1 and C_2, respectively. The training of the LLGMN is performed using the supervised learning algotithm. Here, $T^{(\boldsymbol{x}_1)}$ and $T^{(\boldsymbol{x}_2)}$ are given as virtual teacher vectors to $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The initialization of weights is carried out to prevent the LLGMN to convergence to a local minimum. With the initialized weights, unsupervised learning of the LLGMN is then performed. The mean values of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are calculated using the training data clustered into C_1 and C_2, respectively. One data $\boldsymbol{x} \in \boldsymbol{A} - \boldsymbol{B}$ is selected, and it is added into the set $\boldsymbol{B}$ labeled with either of C_1 and C_2, from whose central the Euclid distance of x is minimum. Training of the LLGMN according to Equation 5.1 is performed using data in the set $\boldsymbol{B}$. After training with a pre-defined number of times, another training data is selected from the set $\boldsymbol{A} - \boldsymbol{B}$ and added into the set $\boldsymbol{B}$. This training step repeats, until all of the training data is added into the $\boldsymbol{B}$, i.e. $\boldsymbol{B} = \boldsymbol{A}$.

5.3.5 Summary of Algorithm

The construction algorithm for the classification tree is summarized as follows:

1. The training data is presented to the root node.

2. Training data in a terminal node is divided into two subclasses using the LLGMN until the termination criterion is satisfied.

3. Whether or not to split a node is determined using the posterior probabilities of the validation data.

4. Steps 2 and 3 are repeated until the termination criterion is satisfied at all terminal nodes.

5. The terminal nodes corresponding to outliers are merged according to the pruning law.

Table 5.1: Parameters of each class used in experiments of artificial data

class	μ_x	μ_y	δ_x	δ_y	δ_{xy}
C1	0.4	0.2	0.08	0.04	-0.8
C2	0.2	0.7	0.05	0.05	0.
C3	0.7	0.6	0.03	0.03	0.
C4	0.6	0.8	0.03	0.03	0.
C5	0.8	0.8	0.03	0.03	0.

By performing the construction of the classification tree and the training of the LL-GMN, clustering with the desired number of classes based on the statistical information can be attained.

5.4 Experiments

Numerical simulations were carried out in order to verify the effectiveness ot the proposed method compared to conventional methods. In addition, pattern clustering and classification experiments of the EMG signals were carried out.

5.4.1 Numerical Simulation

The feature data is illustrated in Fig. 5.1: There are two features $\boldsymbol{x} = (x, y)$, and five classes, $C_i (i = 1, 2, 3, 4, 5)$. Each class consists of one normal distribution, parameters of which are shown in Table 5.1. The number of training data for each class is 100 , and the number of validation data for each class is 200. The LLGMN includes seven units in the first layer, two units in the second layer corresponding to the total number of components, and two units in the third layer. To construct the classification tree, the threshold of entropy H_T was set as 0.2, the threshold of pruning α_t as 0.01, the learning rate η as 0.01, and the training times for each addition of training data as 100. An example of the constructed classification tree is illustrated in Fig. 5.2, where the circles and squares indicate non-terminal and terminal nodes respectively. The

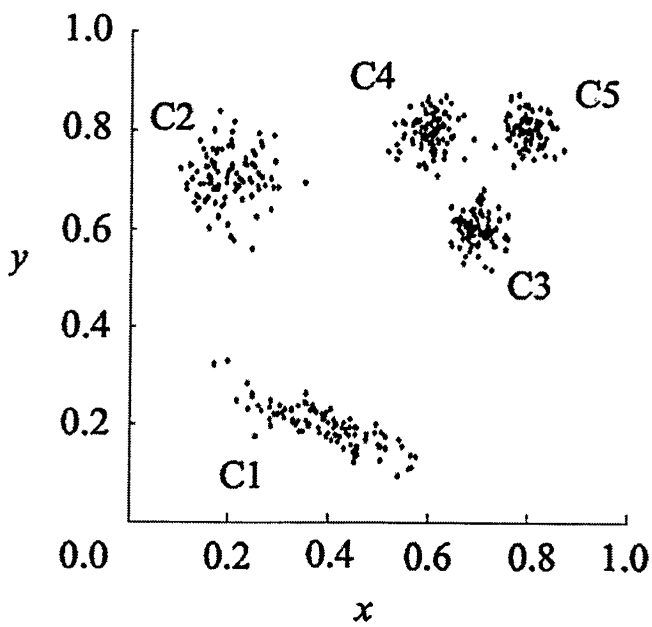


Fig. 5.1: Example of artificial data, the parameters of which are shown in Table 5.1

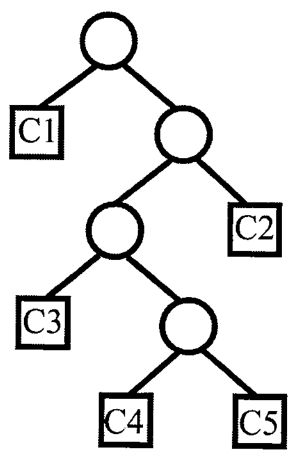


Fig. 5.2: The constructed tree for five classes data.

number of terminal nodes corresponds to the number of classes. Each terminal is a node labeled according to the class that the data belongs to. The classification tree

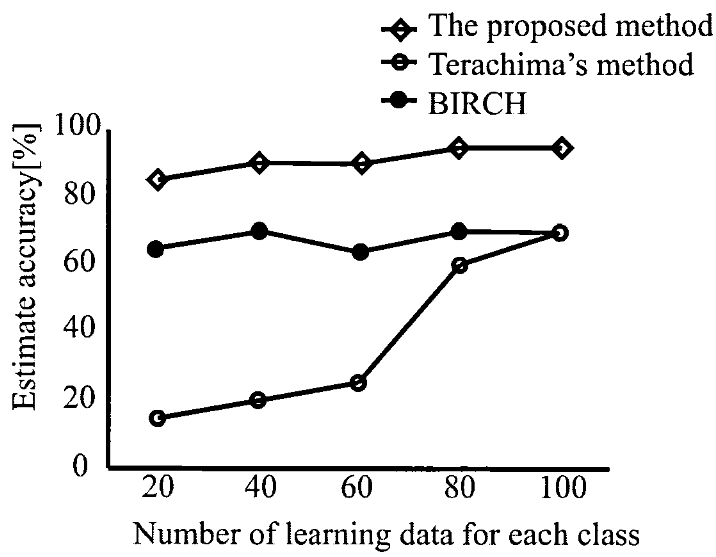


Fig. 5.3: Effect of the number of learning data on estimate accuracy.

starts from the root node, where the training data is divided into two nodes: one for $C1$ and the other for all other classes. Data clustered into the latter node is then further split into node $C2$ and the rest. Finally, the constructed classification tree partitions feature data into five classes. To validate the generalization ability, 300 samples for each class not used during the training process were clustered, and the mean value and S.D. of classification rate for 20 independent trials is 98.5 ± 0.64.

Next, comparison experiments were carried out with conventional method proposed in [52] (Terashima's method) and BIRCH.

The numbers of training data for each class was changed from 20 to 100. For each condition of training data number, we constructed classification tree 20 times. The number of validation data is twice as much as the number of training data. The ratio of times that five classes were correctly estimated is indicated in Fig. 5.3. As shown in this figure, even if the number of training data decreases, the estimation accuracy

Table 5.2: The number of classes estimated by the proposed method, the Terashima's method and BIRCH

The number of classes	4	5	6	7
The proposed method	0	9	1	0
The Terashima's method ($N_U = 50$)	2	7	1	0
The Terashima's method ($N_U = 40$)	5	5	0	0
The Terashima's method ($N_U = 30$)	7	3	0	0
BIRCH	0	0	3	7

Table 5.3: Parameters of each class used in experiments of artificial data

class	μ_x	μ_y	δ_x	δ_y	δ_{xy}
C1	0.4	0.2	0.08	0.1	-0.8
C2	0.2	0.7	0.08	0.08	0.
C3	0.7	0.6	0.05	0.05	0.
C4	0.6	0.8	0.05	0.05	0.
C5	0.8	0.8	0.05	0.05	0.

rate of the proposed method keeps in a high level compared to the other methods.

Table 5.2 indicates the number of trials that training data was classified into each number of classes. As shown in this table, the proposed method can estimate the numver of classes more accurately compared with other methods. When data for several classes are gathered in the input space, BIRCH classifies the data into one class. Such a problem is frequently found in traditional clustering medhods. The proposed methods avoids this problem by constructing a hierarchical tree for classification. Consequently, the proposed method achieves higher classification performance than the conventional method.

Next, pattern classification of synthetic data different from those in Table 5.1 was conducted. An example of the feature data is indicated in Fig. 5.4, and the parameters for each class are shown in Table 5.3. The number of training and validation data was 100 and 200 respectively. Ten different sets of initial weights, training data and validation data were used to construct the classification tree. Table 5.4 depicts the

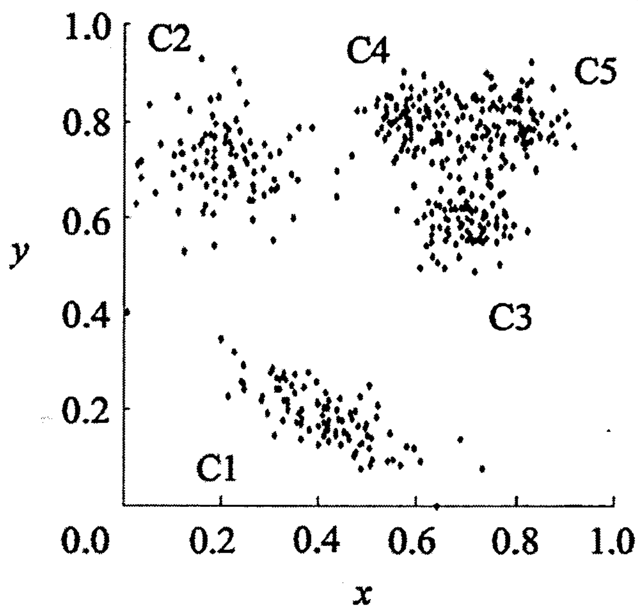


Fig. 5.4: Example of artificial data, the parameters of which are shown in Table 5.3

Table 5.4: The number of classes estimated by the proposed method, the Terashima's method and BIRCH

The number of classes	3	4	5	6	7
The proposed method	1	0	9	1	0
The Terashima's method ($N_U = 50$)	9	1	0	0	0
The Terashima's method ($N_U = 40$)	9	1	0	0	0
The Terashima's method ($N_U = 30$)	10	0	0	0	0
BIRCH	0	0	2	2	6

number of trials that the training data was split with each number of classes.

The proposed method can cluster data successfully in the overlapped region according to its posterior probability. The mean value and S.D. of the classification rate for ten trials are 89.5 ± 3.271. Although the classification rate decreases compared to the data shown in Fig. 5.1, it is shown that the proposed method can cluster data with overlaps between the clusters.

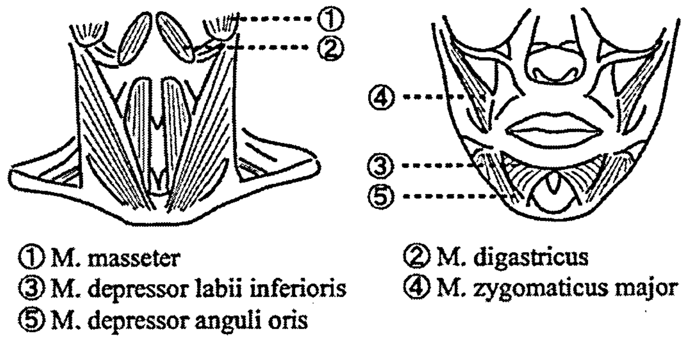


Fig. 5.5: Measured muscles of cervical and expression

As described above, the proposed method can carry out better estimations of the number of classes than conventional methods even when the cluster data overlaps.

5.4.2 EMG Pattern Classifications

Experiments for the pattern classification of EMG signals were carried out. The EMG signals used were five-channel data; they were measured from five mimetic and cervical muscles of a patient with a cervical spine injury (see Fig. 5.5) for six phonemes, i.e. /a/, /i/, /u/, /e/, /o/ and /n/. In the experiments, the patient produced the six phonemes in the order. However, since no acutual voice was uttered, the patient contrasts the muscles relevant to utterance. A reliable label of teacher signals is not available.

First, the EMG signals are digitized by an A/D converter (sampling frequency: 1.0kHz) after being amplified, rectified and filtered through a digital second order Butterworth filter (cut-off frequency: 1.0Hz). These sampled signals are represented as $E_l(t)$. To recognize the beginning and ending of utterance, the force information $F_{EMG}(t)$ is calculated from EMG signals as

$$F_{EMG}(t) = \frac{1}{L}\sum_{l=1}^{L}\frac{E_l(t) - E_l^{st}}{E_l^{max} - E_l^{st}}, \tag{5.7}$$

Table 5.5: Clustering results

Node number	/a/	/i/	/u/	/e/	/o/	/n/
1	34	1	0	0	0	0
2	0	23	4	0	0	0
3	2	2	2	0	0	0
4	2	12	8	0	0	0
5	0	0	21	0	0	0
6	0	0	0	0	0	34
7	0	0	0	9	0	0
8	0	0	0	26	0	1
9	2	2	5	5	4	3
10	0	0	0	0	36	3

where E_l^{st} is the mean value of $E_l(t)$ measured while relaxing the muscles, and E_l^{max} is the mean value of $E_l(t)$ measured under the maximum voluntary contraction. For the phoneme classification $x(t)$ is normalized to make the sum of five channels equal 1:

$$x_l(t) = \frac{E_l(t) - E_l^{st}}{\sum_{l=1}^{L}(E_l(t) - E_l^{st})}. \tag{5.8}$$

Fig. 5.6 shows an example of the raw EMG signals, filtered EMG signals used for classification and the force information. Fig. 5.7 depicts the average value and S.D. of normalized patterns belonging to /i/ and /u/. The patterns of /i/ and /u/ are quite similar to each other. To construct the classification tree, the threshold of entropy H_T is set as 0.2, the threshold of pruning α_T as 0.01, the learning rate η as 0.01 and the training times for each addition of training data is set as 500. The number of learning data for each phoneme was 40, and the number of verification data for each vowel soud was 80. The LLGMN includes 21 units in the first layer, six in the second layer that corresponds to the total component number, and six in the third layer. Fig. 5.8 illustrates an example of the constructed classification tree. Table 5.5 indicates the distribution of data of each phoneme among the terminal nodes.

Most data for /a/ were classified into one terminal node, as well as the data for /o/

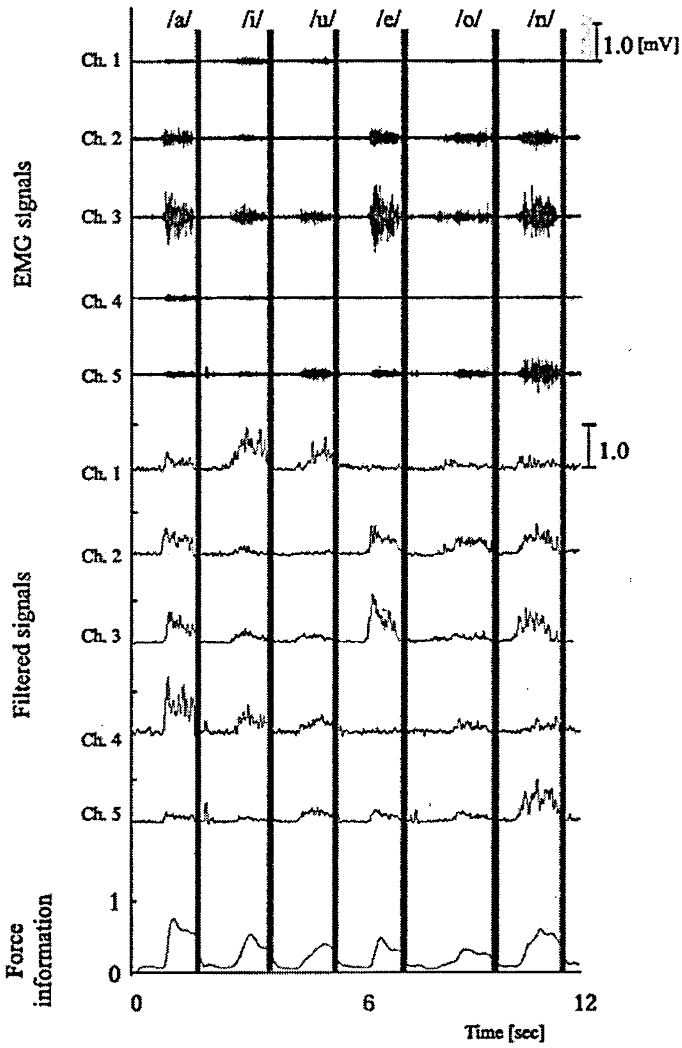


Fig. 5.6: An example of cervical spine injury patient's raw EMG signals, filtered signals used for clustering and the force information.

and /n/. These patterns can be correctly classified. The data of /e/ were classified into two terminal nodes. Since the pattern for /e/ varies when uttered, the data can be classified into two clusters. Some data belonging to /i/ and /u/ were classified into one class because they are similar.

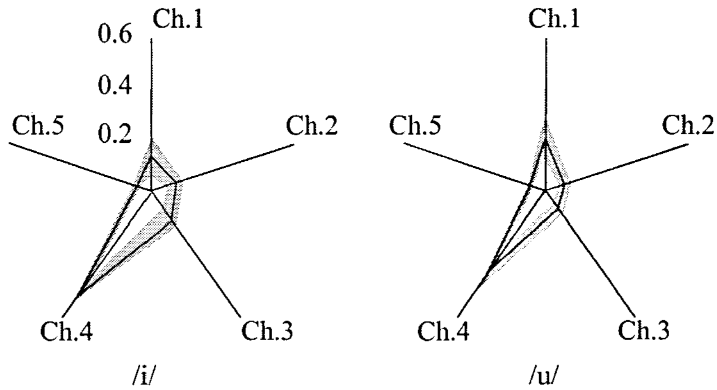


Fig. 5.7: Rader chart of EMG patterns of /i/ and /u/. The line indicates the mean values of each channel, and the regions of shade imply ± 1 S.D.

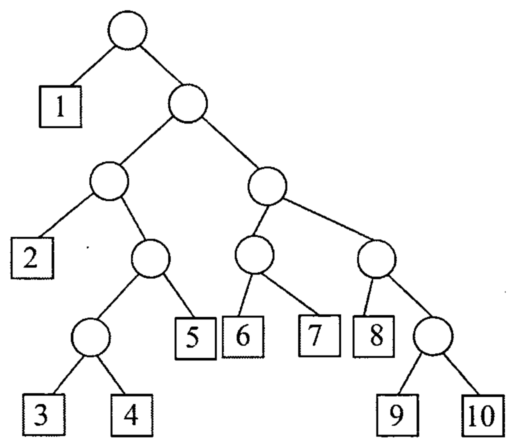


Fig. 5.8: The constructed tree for the EMG data

On the other hand the data classified into node 9 includes all phonemes. The shaded parts in Fig. 5.6 indicate data classified into node 9. The data classified into class 9 were uttered just before the end of the utterance. Such data is ambiguous data and classified into one cluster. As described above, considering the features of data the proposed method can classify EMG signals successfully. In order to examine the validity of the proposed method, classification experiments were carried out. The

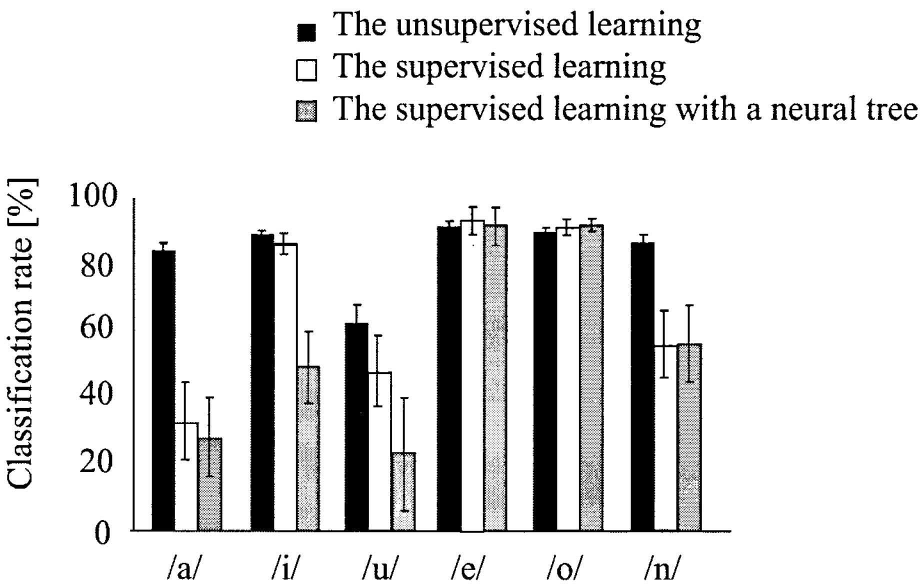


Fig. 5.9: Classification accuracy

classification ability of the proposed method was compared to a supervised trained single LLGMN and a neural tree [58] using the LLGMN as a non-terminal node. To evaluate the discrimination accuracy, labeling for the node was performed. It assumes that each node was corresponded to utterance that most of data classified into the correct node. Although utterance /e/ corresponds to terminal node 7 and 8, it is assumed that all data classified into these terminal nodes are classified into utterance /e/ by constructed tree. After construction of the classification tree, 600 data for each class were prepared for classification. Fig. 5.9 shows the classification rate in case of considering data classified into class 9 as no utterance and suspending classification. The mean values and S.D. of classification rate of all utterance using the proposed method for 20 times is 92.6 ± 0.86, using the LLGMN traditional recognition is 85 ± 2.12 and using Neural tree is 61.4 ± 4.3. The proposed method suspending classification achieved high classification rate. As described above, it is considered that this method is more effective in discrimination of data with the ambiguous feature between each class than other methods.

5.5 Concluding Remarks

In this chapter, in order to deal with the classification problem of ambiguous teacher signals, a hierarchical clustering is proposed. In this method, LLGMNs is used as classifier at non-terminal nodes in the classification tree.

Entropy of the LLGMN outputs and the number of data at each node are used as stopping and pruning indices in the proposed method to avoid unnecessary splits in the structure of classification tree; this allows the proposed method to make interpretable and reasonable partitions of the training data according solely to the statistical characteristics.

In numerical simulations, the proposed method shows superior results to the conventional methods when estimatiing of the number of classes. From the results of the EMG pattern classification experiments, it is considered that the proposed method is more effective in classification data with the similar features compared to traditional methods where the LLGMN is trained using a supervised training algorithm and neural tree.

Chapter 6

Human Interface Applications using Biological Signals

6.1 Introduction

In this chapter, two humanmachine interfaces using biological signals are proposed. By improving traditional interfaces with the proposed classification methods, it is expected that people with disabilities, who cannot utilize traditional systems, can be assisted by these systems.

The chapter is organized as follows: Section 6.2 describes the proposed text input system using EMG signals. Section 6.3 introduces human-machine interface to control electrical appliances. Finally, the last section summarize the chapter.

6.2 Text input system using EMG signals

6.2.1 System Description

The structure of the Japanese text input system is shown in Fig. 6.1. This system can be divided into four parts: (1) EMG signal acquisition and feature extraction, (2) phoneme classification, (3) character selection and (4) word estimation.

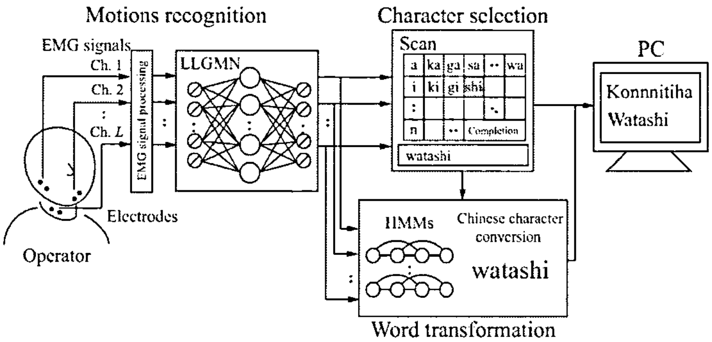


Fig. 6.1: Overview of the text input system

EMG Signal Acquisition and Feature Extraction

EMG signals from the L channels are recorded using surface electrodes attached to muscles. The signals are measured with a sampling frequency of f_s Hz and then rectified and filtered by a Butterworth filter (cutoff frequency: f_cHz). Each sampled EMG pattern, defined as $E_l(t)(l = 1, 2, \cdots, L)$, was normalized to make the sum of five channels equal to 1 using the following equation;

$$x_l^{(t)} = \frac{E_l(t) - E_l^{st}}{\sum_{l'=1}^{L}(E_{l'}(t) - E_{l'}^{st})}, \tag{6.1}$$

where E_l^{st} is the mean value of $E_l(t)$ measured while rthe muscles are relaxed. The feature vectors $\boldsymbol{x}(t) = [x_1(t), x_2(t), \cdots, x_L(t)]$ are input into the classification network. The power level is estimated from the EMG signals as

$$\alpha(t) = \frac{1}{L}\sum_{l=1}^{L}\frac{E_l(t) - E_l^{st}}{E_l^{max} - E_l^{st}}, \tag{6.2}$$

where E_l^{max} is the mean value of $E_l(t)$ measured under the maximum voluntary contraction. The power level is compared with a predetermined threshold to determine whether motion actually occurred.

Pattern Classification

Although LLGMN is generally employed for pattern classification, other methods such as classification methods described in Chapters 2, 3 and 4 can be used. Using samples labeled with their corresponding motions, the network structure is built, and the network learns the non-linear mapping between the EMG patterns and motions simultaneously.

The entropy of the output is also calculated to prevent the risk of misclassification. The entropy is defined as Equation 15.

$$H(\boldsymbol{x}(t)) = -\sum_{c=1}^{C} O_c(\boldsymbol{x}(t)) \log O_c(\boldsymbol{x}(t)), \tag{6.3}$$

where $O_c(\boldsymbol{x}(t))$ corresponds to the posterior probability of motion number c. If the entropy $H(\boldsymbol{x}(t))$ is less than the discrimination threshold Te, the specific motion with the largest probability is determined according to Bayes' decision rule. Otherwise, determination is suspended.

$$Y(\boldsymbol{x}) = \arg\max_{c} O_c(\boldsymbol{x}) \tag{6.4}$$

Character Selection

In this part, character selection is performed continuously to produce a sequence of characters using classified motions.

Fig. 6.2 shows an example of the character selection screen. The shaded area indicates a cursor, which can translate automatically; the character at the cursor is selected by a classified specific motion corresponding to the determination command. In this system, not only Japanese characters but also other keys (delete key, space key and so on) are arranged on the screen (as shown in Fig. 6.2).

In this system, three control modes are available because the number of motions that can be classified correctly is different for different people. In the first control

Direction of cursor movement →

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
a	ka	ga	sa	za	ta		da	na	ha	ba	pa	ma	ya	ya	ra	wa
i	ki	gi	shi	zi	ti		di	ni	hi	bi	pi	mi			ri	
u	ku	gu	su	zu	tu	tu	du	nu	hu	bu	pu	mu	yu	yu	ru	
e	ke	ge	se	ze	te		de	ne	he	be	pe	me			re	
o	ko	go	so	zo	te		do	no	ho	bo	po	mo	yo	yo	ro	wo
n	,	.	Back Space	All Delete	Change	Kanji	Enter	①	②	③	④	⑤	⑥	⑦	⑧	Comp-letion

Fig. 6.2: An example of the character selection screen

mode, the cursor moves from the left grid to the right grid in the first column. If the cursor moves to the far right side of the grids, it moves to the far left in the next movement. By classified motion corresponding to the determination command, the cursor movement is shifted to the vertical direction, and cursor moves along the grid continuously. When the corresponding motion is classified again, the character at the cursor is selected.

If the phoneme classification part can classify the corresponding EMG signals into two motions, using the second control mode, motion can be matched to the command for movement between column. Another motion also corresponds to the determination command. Using more than three classified motions, various commands (e.g, delete, enter and so on) can be added arbitraily for the user ' s convenience.

Furthermore, an input algorithm using six phonemes (/a/, /i/, /u/, /e/, /o/ and /n/) for classification is as follows: Calculate the duration of the classified specific motion t' (see Fig. 6.3). If $t' \geq T'$, classified character is selected. If not, the cursor moves to the column, corresponding to the classified phoneme, from the left edge to the right edge. By the classified phoneme same as one previously classified, the character

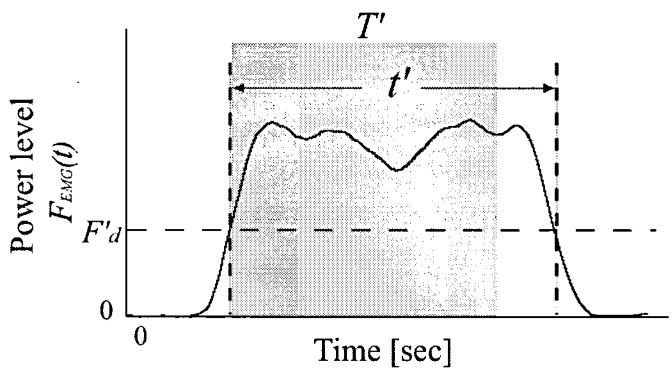


Fig. 6.3: Definition of the motion time t'

at the cursor is selected. If other phonemes are classified, the column for movement is changed to the corresponding column. Thus, the system's method of operation can be changed according to the user's ability.

Also, using the force information estimated from the EMG signals, the transition time of the cursor is changed arbitrarily. Table 1 shows the relationships between force information and transition time. To coordinate the cursor's transition time, user can arbitrarily set the threshold of the force classifying phoneme.

Word Estimation

In this part, the sequence of characters generated by the character selection part is converted into the corresponding kanji, and the complete word is predicted by matching the input character sequence with possible words. For kanji translation, this system uses a database of relationships between kanji and different character sequences.

HMM, which has been developed successfully, especially in the field of speech recognition, is applied for word recognition. One HMM is prepared for each word. The posterior probabilities of each word are normalized as

$$O_i = \frac{P(i|S_1S_2\cdots S_N)}{\sum_{i=1}^{M} P(i|S_1S_2\cdots S_N)} \tag{6.5}$$

where $S_1S_2\cdots S_N$ is the input character sequence, $P(i|S_1S_2\cdots S_N)$ is the posterior probabilities estimated by the ith HMM, and M is the number of HMMs.

For HMM training, data consisting of words and corresponding character strings, is prepared. In this system, character strings consisting of vowels are included in the training data because users can input vowels directly. Since HMMs approximate the probabilistic characteristics of time series through learning, robust recognition can be achieved for words with varying temporal characteristics. Candidate words estimated in this part are displayed in the grid on the screen (see Fig. 6.2). Since the user can also select these words from the screen, words based on the same character string can be entered easily.

Finally, the generated text is input into various application on the PC.

6.2.2 Experiments

To examine the performance of the proposed system, control experiments were performed. The EMG signals were measured from five electrodes attached to the operator ' s face ($L = 5$: Depressor Anguli Oris, Zygomaticus Major, Masseter, Digastric, Depressor Labii Inferioris; a pair of electrodes was placed on each muscle).

EMG Pattern Classification

Five sets of randomly chosen initial weights were used to train each set of sample data. Experiments of the EMG classification using LLGMN and the classification method proposed in Chapter 3 were performed. To verify the performance of the proposed method, a single LLGMN, a support vector machine (SVM) and back-propagation

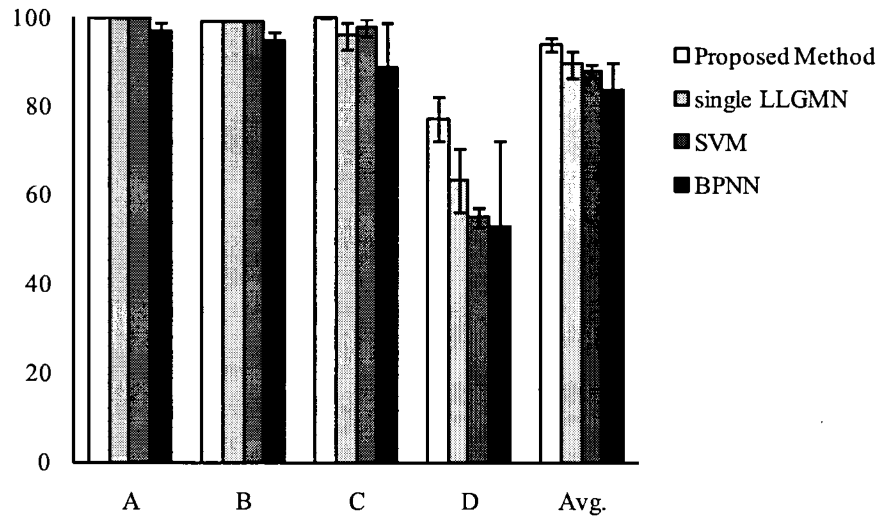


Fig. 6.4: Discrimination results for four subjects.

neural network (BPNN) classifiers were used for the comparison. BPNN had four layers (including two hidden layers), the units of which were set as 2, 10, 10 and 4. Also, an SVM having a second-order polynomial kernel was used to perform a two-class classification. By combining two-class classifiers, multiclass classification was achieved using SVMs. The experiments were performed for four subjects (A, B, C: healthy; D: a patient with a cervical spine injury). For each subject, the proposed network is trained by using training data measured from corresponding subject. The mean values and standard deviations of the classification rates using the LLGMN, the method proposed in Chapter 3 and other methods are shown in Fig. 6.4. The classification results of the proposed method are similar to those of a single LLGMN and SVM in the case of subjects A and B. Comparing the classification rates of subject C, it can be seen that the method using the proposed EMG pattern classification method outperformed the use of a single LLGMN and BPNN. In addition, in the case of subject D, the

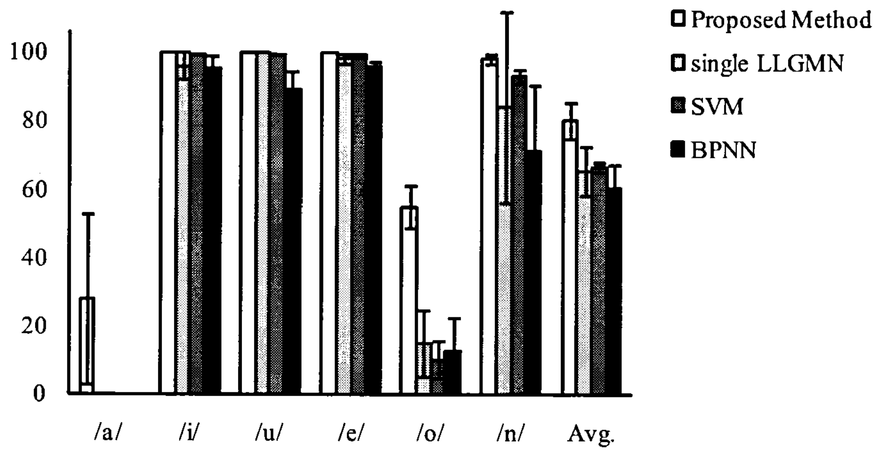


Fig. 6.5: Discrimination results for Subject D.

classification results of other methods degraded significantly more than that using the method proposed in Chapter 3.

Fig. 6.5 shows the mean values and standard deviations of the classification rates of each type of utterance for subject D.

From this figure, it can be seen that the classification accuracies of utterance /o/ and /n/ were improved by the proposed method. On the other hand, in the classification of utterance /a/, similar misclassifications occurred when using the proposed method and other methods. It considered that these results were caused by the ambiguous EMG pattern found with the utterance /a/.

Table 6.1 shows an example of the number of added classifiers and LLGMNs in the network of subject D. In contrast, the number of classifiers and LLGMNs for subjects A and B were set as 1. Here, we infer that classification is improved by adding the classifiers and LLGMNs for the estimating the distribution of EMG patterns. It is

Table 6.1: The number of added classifiers and LLGMNs for subject D.

phoneme	classifiers	LLGMNs
/a/	13	33
/i/	4	7
/u/	2	3
/e/	4	9
/o/	16	24
/n/	5	11

clear that by adding LLGMNs to a network for the estimating the distribution of EMG signals, the proposed method can achieve the most accurate classification of all the methods considered.

6.2.3 System Control Experiments

The experiments were performed for four subjects (A, B, C: healthy; D: a patient with a cervical spine injury). For each subject, the proposed network is trained by using training data measured from corresponding subject. The results in Fig. 6.4 show that single LLGMN can be used as a classifier, and when subjects A, B or C use this system, six phonemes can be utilized for input signals. On the other hand, in the case of subject D, only three phonemes (/i/, /u/ and /e/) can be used as input signals, because the classification rates of /o/ and /n/ were too low for use as input signals.

In this system, the number of selected character strings is set as 22, corresponding to the number of HMMs. In these experiments, the Baum-Welth algorithm was used for HMM learning.

The experimental view and display of the proposed system are shown in Fig. 6.6 and 6.7, respectively. Figure 6.7, shows that the number symbols are arranged in the column corresponding to /n/. By selecting these number symbols, the user can chose the kanji and the recognized word estimated by HMMs (see Fig. 6.7 (A) and (B)).

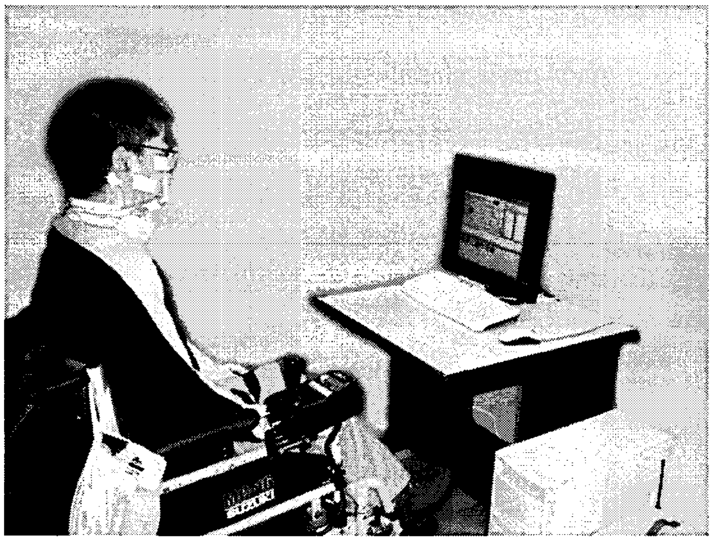


Fig. 6.6: Experimental view of the text input system.

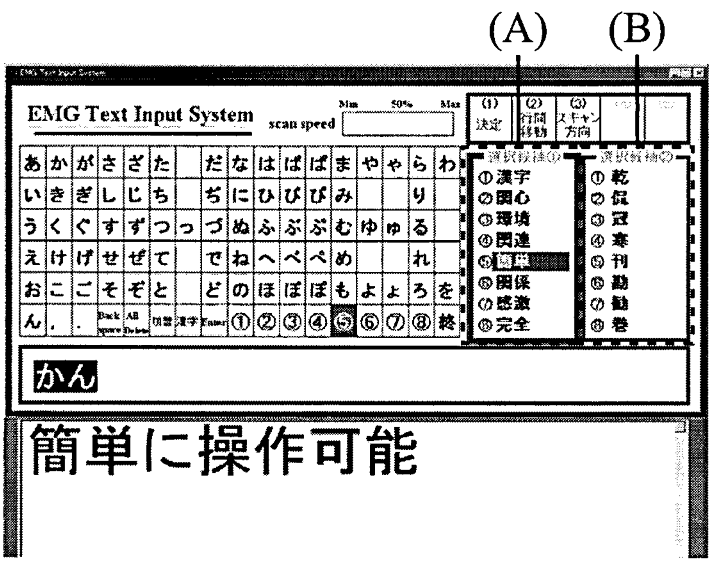


Fig. 6.7: Display of the text input system.

Also, categories (A) and (B) are changed by selecting the corresponding character on the screen.

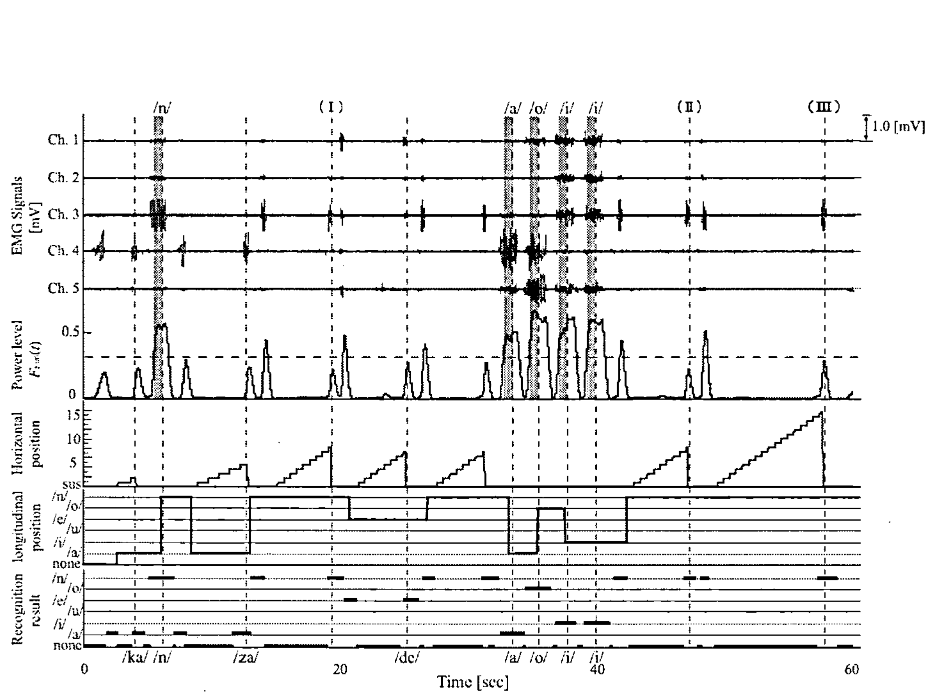


Fig. 6.8: An example of the operation with subject A.

The detailed time history of subject A is shown in Fig. 6.8. In this experiment, the character string "Kantan de Tanosii" is input in Japanese. In this figure, the EMG signals, force level, horizontal position, longitudinal position and recognition results are plotted. Gray areas indicate the threshold T ' . It can be seen that with character selection using classified phonemes and word recognition, subject A can input text successfully. Furthermore, misguided selection (corresponding to /za/) is modified (from /kanza/ to /kantan/) by word recognition using HMMs (see Fig. 6.8 (I)). In time (II), the input sequence /aoii/ is changed to /tanosiii/ by HMM recognition. Finally, the character string is input to a text editor (see Fig. 6.8 (III)).

Part of the time history corresponding to Fig. 6.8 is shown in Fig. 6.9. In this figure, each gray area indicates the length of transition time, which is determined by the power level. It is confirmed that the transition time is adjusted based on the power level of the EMG signals. These results shows that the user can select a suitable

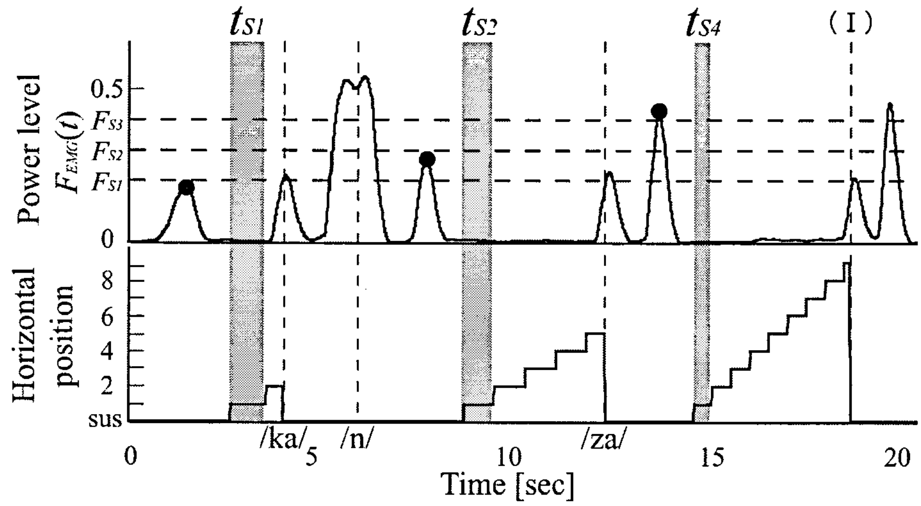


Fig. 6.9: An example of the operation with subject A (from 0 sec to 20 sec).

transition time successfully.

Next, a detailed time history (from 90s to 115s) of subject D is shown in Fig. 6.10. Subject D input /tanosii/ using three utterances (/i/, /u/ and /e/). To input arbitrary characters using only three utterances, the utterances correspond to "determination command", "movement of column" and "invert the direction of cursor movement" respectively. This figure shows that subject D used word recognition to estimate the character string "tanosii"from the input string "tano". Using word recognition, input took 136.46 s. To input the same string, subjects A, B and C took 70.37, 79.70 and 96.13s respectively. In these experiments, since only subject D had no experience operating the proposed system, it took longer for that subject than for others. The operating time is expected decrease gradually with training in using the system.

To validate the effectiveness of word estimation, input times were compared. In these experiments, subjects A, B and C used six utterances to control this system. The input words are "watasi", "tanosii", "ohayou" and "hijouni" and these wordS are

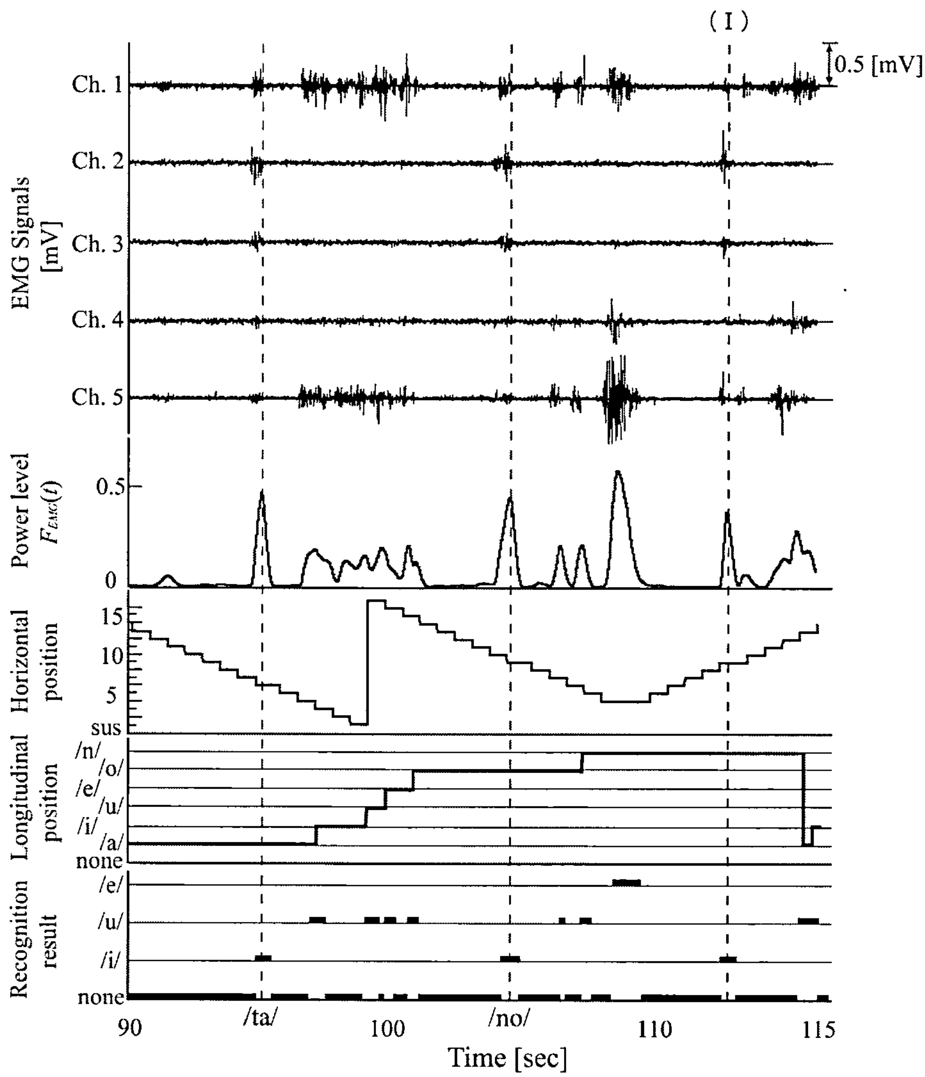


Fig. 6.10: An example of the operation with subject D.

transferred into Chinese kanji. The transition time of the cursor is set as 0.67 s.

The comparison results for five independent trials are shown in Fig. 5.11. In these experiments, three transfer methods were used: (1) kanji conversion from all characters, (2) kanji conversion from parts of characters (e.g, "wata", "tano", "oha" and "hijo")

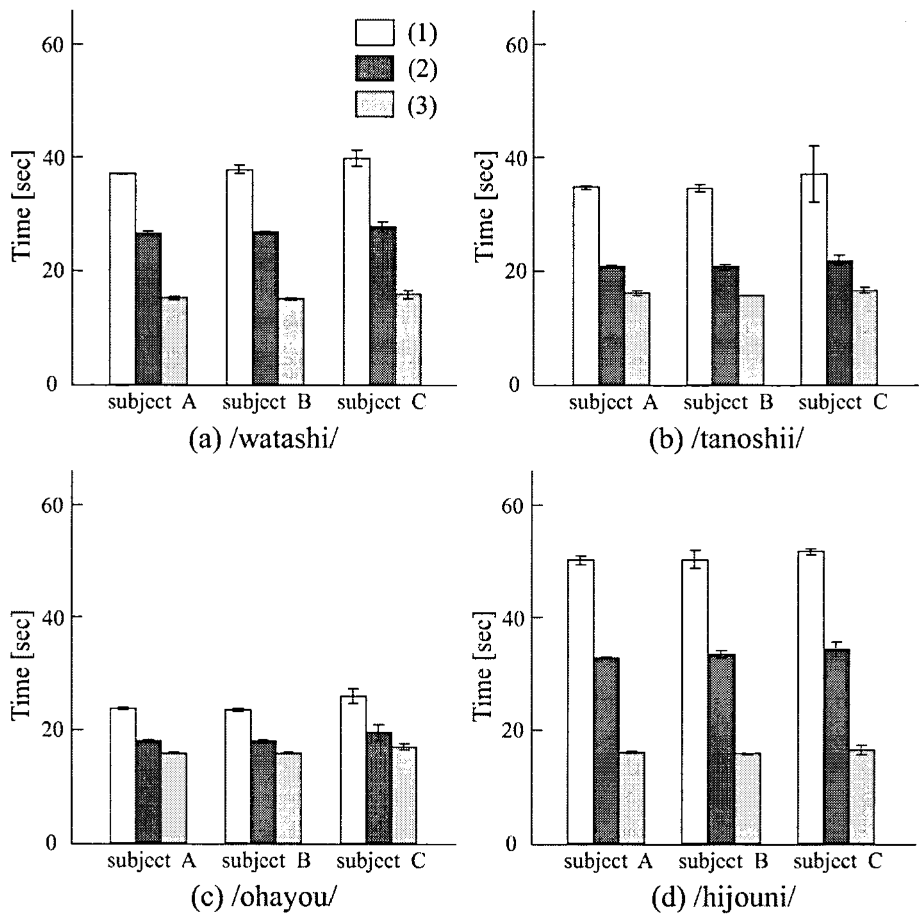
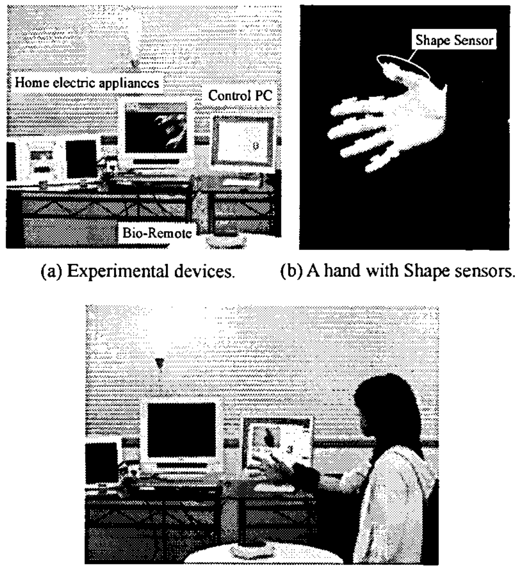


Fig. 6.11: Input time of words.

and (3) kanji conversion from characters consisting of only vowels. These results show that text input using method (3) is the fastest of all methods tested. However, in the case of inputting a word not expected by the HMMs, it is impossible to perform input using methods (2) and (3). Also increasing the words for recognition interrupts effective recognition of words using HMMs. In such cases, input using method (1) is needed.



(a) Experimental devices. (b) A hand with Shape sensors.

(c) An operation scene for home electric appliances.

Fig. 6.12: Opetation of home electrical appliances using discrimination results of hand shape.

6.3 Application of Hand Shape Classification for Human Interface

In this section, improved Bio-Remote system to control electrical appliances using classification method proposed in Chapter 2 and 3 is dveloped. This system was manipulated according to the user's intention determined from the biological signals.

In general, it is difficult to discriminate user's intentions from biological signals. Therefore, if necessary, the user can manipulate various applications with residual functions that combine input channels using this system. The method proposed in Chapter 2 and 3 can discriminate various hand shapes from biological signals. The function of the control system for electrical home appliances using hand shapes is shown in Fig. 6.12. [61]. In this system, the discrimination results are sent to the main unit and the infrared signals corresponding to the electric home appliances are transmitted directly from the infrared LED of the main unit to the appliances.

Examples of the operations corresponding to discriminated motion are shown in Table 6.2. [61]. In a typical Bio-Remote, various operations can be performed by

Table 6.2: Example of command allocation for home electric appliances.

Motion number	Object	Command
1	Light	On
2		Off
3	TV	Switch
4		CH up
5		CH down
6	CD player	Power on/off
7		Play
8		Stop
9		Volume up
10		Volume down
⋮	⋮	⋮

repeating a command selection. However, from this figure, it can be inferred that each operation corresponds to a single hand shape motion.

An experiment was conducted, using a healthy person as the subject, to verify the validity of the proposed method. In this experiment, operations corresponding to the user's motions were executed until the same discrimination occured 150 times. An example of the subject's operation is shown in Fig. 6.13. In this figure, five channels of the normalised signals, discrimination results and control commands are plotted. Gray areas indicate that the Bio-Remote is not operated.

From these experimental results, it can be inferred that the subject could operate electrical home appliances by changing the position of his/her fingers. It should be noted that there was no malfunction, and that the appliances could be operated according to the subject's intent, which confirms that by the use of the proposed system,

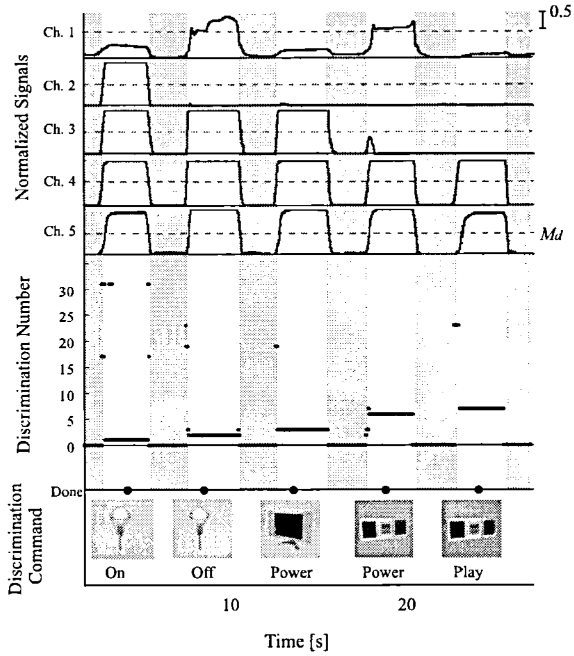


Fig. 6.13: An example of the experimental results during the hand shapes of the electric appliances by the subject.

the subject can control various electric appliances simply by moving his or her fingers.

6.4 Concluding Remarks

In this chapter, two devices using biological signals are proposed. First, a novel text input system using EMG signals are descrived. In this system, motions estimated from EMG signals with PNNs are used as control commands. Based on the number of estimated motions corresponding to commands, the proposed system can apply the control method that is suitable for the number of motions. In addition, using HMMs corresponding to words, pre-defined words (character strings) are recognized from a partial character string. This recognition method enables users to perform effective input in less time. To validate the availability of EMG signals as control commands, EMG classification experiments were performed. These results show that EMG classification was achieved with a high success rate using a single LLGMN or the classification method proposed in Chapter 3. To further examine the realization of the proposed system, text input experiments were performed. These experiments showed that the proposed system can help not only healthy peple but also handicapped people to perform text input using only their utterances.

The second device is a human interface for controlling electrical home appliances using hand shapes. From experimental results, it can be inferred that the operator could control the home electric appliances using hand shapes according to his or her intentions. Furthermore, assignment of the hand shapes directly to the operation command of a electrical home appliances confirmed the feasibility of direct operation of electrical home appliances using the Bio-Remote.

Chapter 7

Conclusion

7.1 Results and Contributions

Chapters 2 and 3 discussed the hierarchical pattern classification method. In Chapter 2, a construction algorithm for a hierarchical classification network based on validation was proposed. Using validation data to evaluate the constructed network, it is expected that the network can achieve generalization accuracy higher than other classification methods trained only training data. On the other hand, in Chapter 3, a hierarchical probabilistic neural network based on a boosting approach was proposed. By connecting weak classifiers consisting of LLGMNs with a boosting approach, proposed method can outperform any of the classifiers. In the learning procedure proposed in Chapters 2 and 3, since network construction and learning of the LLGMNs (network classifiers) are performed simultaneously, there is no need to set the network structure beforehand.

In Chapter 4, a classification method was proposed using the prior probability of data estimated from GMM to eliminate data not belonging to predefined classes. Although the case dealt with in Chapter 4 was not emphasized in previous research in probabilistic pattern classification, this case is a potential problem in classifying patterns in real-world data.

On the other hand, with the unsupervised learning algorithm for the LLGMNs, a hierarchical pattern classification method that can estimate the suitable number of classes was proposed in Chapter 5. In general, there is a possibility that the network cannot be trained to classify data into specified classes because of overlapping data or complex distributions of data. In this case, using the method proposed in Chapter 5, the trained network presents the appropriate splits of training data and the number of classes estimated through learning. Based on numerical simulation and EMG pattern classification experiments, the proposed method is more effective than traditional methods in classifying data that overlap each other. It is expected that the presented information, such as the number of estimated classes and the differeces between predefined classes and estimated classes, can be used not only for pattern classification but also to improve signals measurement and class definition.

Although the proposed methods in Chapters 2, 3, 4 and 5 were developed in order to overcome problems that are actually confronted in pattern classification of data measured from the real world, such as biological signals, it is expected that they can also be used as classifiers for other types of complicated data. Using these methods, users need not decide the parameters and the structure of a network by trial and error. Also, this advantage helps researchers develop effective human-machine interfaces with better classification accuracy and automatic learning function.

In Chapter 6, two human-machine interfaces using biological signals were proposed. Our research group previously proposed the EMG-based Japanese speech synthesizer system using LLGMN and HMMs [60]. This system can recognize words and text based on six Japanese phonemes classified from the user's EMG signals. Although a user who cannot speak, performs vocalization using this system, the user's EMG signals must be correctly classified into six Japanese phonemes in order to use the system. However, in the proposed text input system, the number of control commands corresponding to the given classes can be changed in accordance with user's classification ability. Therefore, many users, including those with greater disability, can input text into a text editor on PC and express their intentions whenever using this system. Also, the Bio-Remote [61], an environmental control system for handicapped persons using biological signals, was proposed by our research group. In this system, a user can select various operations by repeating a command selection. The feasibility of convenient direct operation by the mildly handicapped using hand shapes to directly indicate operation commands to the Bio-Remote is confirmed. As stated above, by developing the various operating methods, it is expected that all persons with disabilities can be assisted by these systems in their daily activities.

7.2 Future Works

In this dissertation, the problems of pattern classification methods, in particular PNNs, are introduced. However each proposed method can overcome only one problem. For practical application, these proposed networks should be integrated into isolated pattern classification methods having PNNs as classifiers.

In future research on the proposed methods, many more theoretical aspects should be studied. The proposed methods should also be used for pattern classification of other data, such as image recognition. In this dissertation, although the LLGMN was used as the classifier, some of the proposed methods (such as those proposed in Chapters 2, 3 and 4) can apply other PNNs as network classifier. To confirm the generalization of the proposed methods, other PNNs should be applied using these methods.

Finally, some problems for future researches are discussed. The pattern classification methods proposed in this dissertation extended the traditional PNNs by constructing the hierarchical tree for classification. The proposed methods have been shown to have high performance for classification of complex data. However, the pattern classification using PNNs should be further improved with studies which are not included in the present researches of this dissertation. Here, two of important issues related to this dissertation will be discussed.

In Chapter 2, 3 and 5, the pattern classification methods using hierarchical tree have been proposed. Although these methods can construct hierarchical tree for pattern classification and adjust parameters of LLGMNs at each non-terminal node, it is impossible to reconstruct structure of hierarchical tree through online learning. The structure of classification networks need to be update regularly. In general it is difficult to train parameters or structure of a network using new data without destroying the old patterns and forgetting previously learned information. In recent years, incremental learning algorithms using unsupervised learning are proposed [62], [63]. Although these algorithms are applied for specific networks such as SOM and k-means which are based on linear classification, it is difficult to use these algorithms as learning algorithms for other NNs based on nonlinear classification. On the other hand, in proposed methods, it is expected that change of subtree doesn't influence classification by other tree structure. Therefore, there is possibility that by performing reconstruction of subtree during long-term daily use, the network can adapt to the changes in patterns.

Pattern classification is frequently confronted with high-dimensional feature data in practical applications [64]. In order to realize the generalization ability, suitable l-dimensional feature should be selected from original features (d-dimension). Although such feature extraction methods based on principal component analysis (PCA) and linear discriminant analysis (LDA) were proposed [22], there is possibility that suitable subspaces for classification of each class are different. Also in proposed methods, suitable projected compact features for classification at root node may be different that for classification of subclasses at non-terminal nodes. In order to deal with these problems, we would like to improve the construction method for network structure which can estimate not only the structure but also suitable dimension of feature data.

Publications concerning this dissertation are listed in the bibliography [65]-[70].

Appendix A

Log-Linearized Gaussian Mixture Network [21]

LLGMN is based on a log-linear model and a Gaussian mixture model (GMM). It calculates posteriori probability for the training data. In this dissertation, LLGMN is utilized for partition at the non-terminal node of the hierarchical tree in Chapter 2, 3 and 5.

The structure of LLGMN is shown in Fig. A.1. In order to represent a normalized distribution corresponding to each component of GMM as weight coefficients of NN, the input vector $\boldsymbol{x}(\in \Re^D)$ is converted into the modified input vector $\boldsymbol{X}$ as follows:

$$\boldsymbol{X} = \{1, \boldsymbol{x}^T, x_1^2, x_1x_2, \cdots, x_2^2, \cdots, x_2x_D, \cdots, x_D^2\}^T \qquad \text{(A.1)}$$

The first layer of LLGMN consists $H = 1 + D(D+3)/2$ units, which correspond to the dimension of the input vector $\boldsymbol{X}$, and the identity function is used for the activation function of each unit. The outputs of the first layer multiplied by weight $w_h^{(k,m)}$ are transmitted to the second layer. Where $w_h^{(K,M_K)} = 0$, K and M_K denote the number of classes (patterns) and components belonging to class M, respectively. In this layer, LLGMN calculates the posteriori probability of each Gaussian component $\{k, m\}$. The

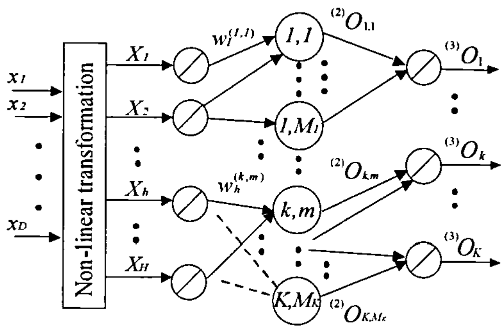


Fig. A.1: The structure of LLGMN.

unit k in the third layer integrates the outputs of M_k units in the second layer.

$$^{(2)}I_{k,m} = \sum_{k=1}^{H} {}^{(1)}O_h w_h^{(k,m)} \tag{A.2}$$

$$^{(2)}O_{k,m} = \frac{\exp({}^{(2)}I_{k,m})}{\sum_{k'=1}^{K}\sum_{m'=1}^{M_{k'}} \exp({}^{(2)}I_{k',m'})} \tag{A.3}$$

The relationship between the input $^{(3)}I_k$ and the output O_k in the third layer is

$$^{(3)}I_k = \sum_{m=1}^{M_k} {}^{(2)}O_{k,m} \tag{A.4}$$

$$^{(3)}O_k = {}^{(3)}I_k \tag{A.5}$$

The output of the third layer $^{(3)}O_k$ corresponds to the posterior probability $P(k|\boldsymbol{x})$ of class k given the input vector $\boldsymbol{x}$, and the former can be used to evaluate the ambiguity of a classification result.

This network has the ability of adaptive learning for statistical properties of data. It can discriminate data with complex distributed structure, and in comparison to the conventional method [59] using normal distribution restricted the parameter.

Bibliography

[1] W. S. McCulloch and W.Pitts:"A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, Vol. 5, pp.115–113, 1943.

[2] B. Widrow and M. E. Hoff: " Adaptive switching circuits," *In IRE WESCON Convention Record*, Vol. 4, pp.96–104, 1960.

[3] F. Rosenblatt: " Principles of Neurodynamics," *Perceptrons and the Theory of Brain Mechanixms*, Spartan, 1962.

[4] D. E. Rumelhart, G. E. Hinton and R. J. Williams: "Parallel Distributed Processing," *ṖExplorations in the Microstructure of Cognition*, Vol. 1: Foundations, MIT Press, 1986.

[5] P. S. Churchland and T. J. Sejnowski: *The computational brain*, The MIT Press, 1992.

[6] S. F. Zornetzer, J. L. Davis and C. Lau: *An introduction to neural and electronic networks*, Academic Press, 1994.

[7] M. Gabriel and J. Moore: *Learning and computational neuroscience: Foundations of adaptive networks*, MIT Press, 1990.

[8] S. Amari: " Theory of adaptive pattern classifiers," *IEEE Trans. Electron. Computers*, EC-16, pp.299–307, 1967.

[9] D. E. Rumelhart, G. E. Hinton and R. J. Williams: "Learning internal representations by error propagation," *Parallel Distributed Processing*, Vol. 1, pp.318–362, 1986.

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jakcel: "Backpropageation applied to handwritten zip code recgnition," *Neural Computation*, Vol. 1, No. 4, 1989.

[11] K. J. Lang, A. H. Waibel and G. E. Hinton: " A time-delay neural network architecture for islated word recgnition," *Neural Networks*, Vol. 3, pp.33–43, 1990.

[12] J. A. S. Freeman and D. Saad: "Online learning in radial basis function networks," *Neural Computation*, Vol. 9, pp.1601–1622, 1997.

[13] R. S. Sutton: " Two problems with backpropagation and other steepest-descent leaning procedures for networks," *in Proc. of Eighth Annual Coference of the Cognitive Science Society*, pp.823–831, 1986.

[14] J. E. Vitela and J. Reifman: " premature saturation in backpropagation networks: mechanism and necessary conditions," *Neural Networks*, Vol. 10, pp.721–735, 1997.

[15] D. F. Specht: "Probabilistic neural network," *Neural Networks*, Vol. 3, pp.109–118, 1990.

[16] M. D. Richard and R. P. Lippmann: " Neural network classifiers estimate bayesian a osteriori probabilities," *Neural Computation*, Vol. 3, pp.461–483, 1991.

[17] G. E. Hinton and D. van Camp: " Keeping neural network simple by minimizing the description length of the weights," *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pp.5–13, 1993.

[18] D. Barber and C. M. Bishop: " Ensemble Learning in Bayesian neural networks, " *Generalization in Neural Networks and Machine Learning*, pp.215–237, 1998. , " *Neural Networks*, Vol. 3, pp.109–118, 1990.

[19] D. J. C. MacKay: "A practical Bayesian framework for back-propagation networks, " *Neural Computation*, Vol. 4, No. 3, pp.448–472, 1992.

[20] D. J. C. MacKay: " The evidence framework applied to classification networks, " *Neural Computation*, Vol. 4, No. 5, pp.720–736, 1992.

[21] T. Tsuji, O. Fukuda, H. Ichinobe and M. Kaneko: " A Log-Linearrized Gaussian Mixture Network and Its Applilcation to EEG Pattern Classification, " *IEEE Trans. on Systems, Man and Cybernetics Part C*, Vol. 29, pp.60–72, 1999.

[22] C. M. Bishop: " Neural Networks for Pattern Recognition, " Oxford University Press, 1995.

[23] T. Tsuji, H. Shigeyoshi, O. Fukuda and M. Kaneko: " Bio-mimetic Control of an Externally Powered Prosthetic Forearm Based on EMG Signals, " *Transactions of the Japan Society of Mechanical Engineers*, Vol. 66, pp.2764–2771, 2000. (In Japanese)

[24] O.Fukuda, T. Tsuji and M. Kaneko: "Pattern Classification of EEG Signals Using a Log-Linearized Gaussian Mixture Neural Network, " *Proc. of IEEE Int. Conf. on Neural Networks*, pp.2470-2484, 1995.

[25] O. Fukuda, T. Tsuji and M. Kaneko: " An EMG Controlled Robotic Manipulator Using Neural Network, " *Proc. of IEEE Int. Workshop on Robot and Human Communication*, pp.442–447, 1997.

[26] O. Fukuda, T. Tsuji and M. Kaneko: "An EMG-Controlled Pointing Device Using a Neural Network," *Proc. of the 1999 IEEE Int. Conf. on System, Man, and Cybernetics*, Vol. 4, pp.63–68, 1999.

[27] O. Fukuda, T. Tsuji, M. Kaneko and A. Ohtsuka: " A Human-Assisting Manipulator Teleoperated by EMG Signals and Arm Motions," *IEEE Trans. on Robotics and Automation*, Vol. 19, pp.210–222, 2003.

[28] O. Fukuda, J. Arita, and T. Tsuji: " An EMG-controlled omnidirectional pointing device using a HMM-based neural network," *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 3195-3200, 2003.

[29] N. Bu, T. Hamamoto, T. Tsuji and O. Fukuda: " FPGA Implementation of a Probabilistic Neural Network for a Bioelectric Human Interface," *The 2004 IEEE International Midwest Symposium on Circuits and Systems*, Vol. III, pp.29-32, 2004.

[30] K. Shima, N. Bu, M. Okamoto, and T. Tsuji: " A Universal Interface for Video Game Machines using Biological Signals," *Proceedings of 4th International Conference on Entertainment Computing - ICEC 2005*, pp. 88-98, 2005.

[31] N. Bu, T. Tsuji, J. Arita, and M. Ohga: "Phoneme Classification for Speech Synthesiser using Differential EMG Signals between Muscles," *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, Paper-ID 788, 2005.

[32] K. Shima, R. Eguchi, K. Shiba, and T. Tsuji: "CHRIS: Cybernetic Human-Robot Interface Systems," *Proceedings of 36th International Symposium on Robotics*, WE1C3, 2005.

[33] R. Kato, H. Yokoi and A. Tamio: " Real-time learning method for adaptable motion-discrimination using surface EMG signal, " *Proceedings of the 2006 IEEE/RSJ International conference on Intelligent Robots and Systems (IROS2006)*, pp. 2127–2132, 2006.

[34] D. Nisikawa, W. Yu, H. Yokoi and Y. Kakazu: " On-line learning method for EMG prosthetic hand control, " *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, Vol. 84, No. 10, pp. 35–46, 2001.

[35] K. A. Farry, I. D. Walker and R. G. Baraniuk: " Myoelectric teleoperation of a complex robotic hand, " *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 5, pp. 775–788, 1996.

[36] T. Tsuji, N. Bu, O. Fukuda and M. Kaneko: "A Recurrent Log-Linearized Caussian Mixture Network, " *IEEE Trans. on Neural Networks,* Vol. 14, No. 2, pp.304–316, 2003.

[37] N. Bu, T. Tsuji and O. Fukuda: " MMI-based Training for a Probabilistic Neural Network, " *Proc. of the 2003 Int. Joint Conf. on Neural Networks,* pp.2661–2666, 2003.

[38] N. Bu and T. Tsuji: " Maltivariate Pattern Classification based on Local Discriminant Component Analysis, " *Proc. of the 2004 IEEE Int. Conf. on Robotics and Biomimetics,* Paper-IE 290, 2004.

[39] L. Breiman, J. H. Friedman, R. A. OLshen and P. J. Stone: *Classification and regression tree*, Wadsworth International Group, 1984.

[40] J. R. Quinlan: " Induction of decision trees, " *Machine Learning*, Vol. 1, No. 1, pp. 81-106, 1986.

[41] J. R. Quinlan: "C4.5," *Programs for Machine Learning*, Morgan Kaufmann, 1993.

[42] T. R. Hastie and J. Friedman: *The Elements of Statistical Learning*, Springer, 2001.

[43] J. A. Sirat and J-P. Nadal: "Neural trees: A new tool for classification," *Network: Computation in Neural Systems,* Vol. 1, pp.423–448, 1990.

[44] R. E. Schapire: " The strength of weak learnability, " *Machine Learning,* Vol. 5, No. 2, pp.197–227, 1990.

[45] Y. Freund: " Boosting a weak learning algorithm by majority, " *Information and Computation,* Vol. 121, No. 2, pp.256–285, 1995.

[46] C. Domingo,T. Tsukiji and O. Watanabe: "Partial Occam's razor its applications," *Information Processing Letters,* Vol. 64, pp.179–185, 1997.

[47] J. H. Friedman: " Greedy function approximation: a gradient boosting machine, " *Annals of Statistics,* Vol. 29, No. 5, pp.1189–1232, 2001.

[48] J. H. Friedman, T. Hastie and R. Tibshirani: " Additive logistic regression: a statisical view of boosting, " *Annals of Statistics,* Vol. 28, pp.337–407, 2000.

[49] S. P. Lloyd: " Least squares quantization in PCM, " *IEEE Trans. on Information Theory,* Vol. 28, No. 2, pp.129–137, 1982.

[50] T. Kohonen: *Self-organizing maps,* Springer, 1995.

[51] S. Suzuki and N. Ueda: " Adaptive Clustering Method using Modular Leaning Architecture, " *IEICE Trans. on Information and Systems,* PT. 2, Vol. J83, No. 6, pp.1529–1538, 2000. (in Japanese)

[52] M. Terashima, F. Shiratani and K. Yamamoto: " Unsupervised Cluster Segmentation Method Using Data Density Histgram on Self-Organizing Feature Map, " *IEICE Trans. on Information and Systems,* PT. 2, Vol. J79, No. 7, pp.1280–1290, 1996. (in Japanese)

[53] B. Fritzke: " A growing Neural Gas Network Learns Topologies, " *Advances in Neural Information Processing Systems,* Vol. 7, pp.625–632, 1994.

[54] T. Zhang, R. Ramakrishnan and M. Livny: " An Efficient Data Clustering Method for Very Large Databases, " *Proc. 1996 ACM SGMOD int. Conf. on Management of Data,* pp.103–114, 1996.

[55] N. Cristianini and J. Shawe-Taylor: *An Introduction to support vector machines,* Cambridge University Press, 2000.

[56] M. J. Tax and P. W. Duin: " Support vector domain description, " *Pattern Recognition Letters,* Vol. 20, pp.1191–1199, 1999.

[57] J. R. Magnus and H. Neudecker: *Matrix differential calculus tieh application in statistics and economietrics,* Wiley, 1999.

[58] G. L. Foresti and C. Micheloni: " Generalized Neural Trees for Pattern Classification, " *IEEE Transaction on Neural Networks,* Vol. 13, No. 6, pp.1540–1547, 2002.

[59] L. I. Perlovsky and M. M. McManus: " Maximum likelihood neural networks fr sensor fusion and adaptive classification, " *Neural Networks,* Vol. 4, pp.89–102, 1991.

[60] O. Fukuda, S. Fujita and T. Tsuji: " A substitute vocalization system based on emg signals, " *The IEICE Trans. on Information and Systems,* Vol J88-D-II, No. 1, pp.105–112, 2005. (in Japanese)

[61] Toshio Tsuji, Keisuke Shima, Akihiko Funabiki, Satoru Shitamori, Kenji Shiba, Osamu Fukuda, and Akira Otsuka:"A New manipulation Method for Environment Control Systems," *The Society of Life Support Technology,* Vol. 18, No. 4, pp.5–12, 2006. (in Japanese)

[62] S. Furao, T. Ogura and O Hasegawa: " An enhanced self-organizing incremental neural network for online unsupervised learning, " *Neural Networks,* Vol. 20, pp.893–903, 2007.

[63] S. Furao and O Hasegawa: " A fast nearest neighbor classifier based on self-organizing incremental neural network, " *Neural Networks,* Vol. 21, pp.1537–1547, 2008.

[64] Anil K. Jain, Robert P. W. Duin and J. Mao: " Statistical Pattern Analysis, " *Muscle & Nerve,* Vol. 23, pp.18–26, 2000.

[65] M. Okamoto, N. Bu and T. Tsuji: " Unsupervised Learning for Hierarchical Clustering Using Statistical Infomation, " *Proceedings of the International Symposium on Neural Networks, Part I,* pp. 834–839, 2004.

[66] M. Okamoto, N. Bu and T. Tsuji: " Hierarchical Clustering with Tree Structure Based on Probabilistic Neural Networks, " *Trans. of the Society of Instrument and Control Engineers,* Vol. 41, No. 3, pp. 283–290, 2005. (in Japanese)

[67] M. Okamoto, J. Murakami, K. Shiba and T. Tsuji: " A Tree-based Hierarchical Probabilistic Neural Network and Its Application to Hand Shape Discrimination,"

Trans. of the Society of Instrument and Control Engineers, Vol. 43, No. 2, pp. 128–134, 2007. (in Japanese)

[68] T. Tsuji, M. Shigeto, M. Okamoto, T. Sugiyama and K. Shiba: " A Text Input System manipulated by EMG Signals," *Journal of Human Interface Society,* Vol. 9, No. 2, pp. 27–34, 2007. (in Japanese)

[69] M. Okamoto, K. Shima, Y. Matsubara and T. Tsuji: " Pattern discrimination method with a boosting approach using hierarchical neural trees, " *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* Vol. 222, No. 7, pp.701–710, 2008.

[70] M. Okamoto, Y. Matsubara, N. Bu and T. Tsuji: " Neuro-based Pattern Classification Method for EMG Signals not Belonging to Predefined Classes, " *Proceedings of the 11th International Conference on Humans and Computers (HC2008),* pp.163–168, 2008.