Unsupervised Learning for Hierarchical Clustering Using Statistical Information

Masaru Okamoto, Nan Bu, and Toshio Tsuji

Department of Artificial Complex System Engineering
Hiroshima University
Kagamiyama 1-4-1, Higashi-Hiroshima, Hiroshima, 739-8527 JAPAN
{okamoto, bu, tsuji}@bsys.hiroshima-u.ac.jp
http://www.bsys.hiroshima-u.ac.jp

Abstract. This paper proposes a novel hierarchical clustering method that can classify given data without specified knowledge of the number of classes. In this method, at each node of a hierarchical classification tree, log-linearized Gaussian mixture networks [2] are utilized as classifiers to divide data into two subclasses based on statistical information, which are then classified into secondary subclasses and so on. Also, unnecessary structure of the tree can be avoided by training in a cross-validation manner. Validity of the proposed method is demonstrated with classification experiments on artificial data.

1 Introduction

Recently, there have been growing interests in using bioelectric signals such as electromyogram (EMG) to conduct man-machine interface. In order to discriminate an operater's intentions from bioelectric signals efficiently, several attempts have been made so far [1], [2]. Generally, such pattern discrimination is performed by estimating the relationship between the bioelectric signals as feature vectors and the corresponding intentions as class labels. However, difference between classes in the bioelectric signals of elderly or handicapped people is ambiguous, and this relates to poor reliability of the class labels available. To overcome this problem, clustering analysis has been widely adopted, in which a collection of patterns is organized into clusters based on similarity.

The previously proposed clustering analysis techniques can be dichotomized as either k-means algorithm or hierarchical clustering. The k-means algorithm identifies a partition of the input space. On the other hand, the hierarchical clustering performs a nested series of partitions and finally performs a grouping with a suitable number of classes. Also, in order to determine the number of class automatically, a clustering algorithm using self organizing maps (SOM) [5] has been proposed [6]. In this method, estimation of the number of classes is carried out based on the number of the data belonging to each node of SOM. However, when parameters in this method were not set up appropriately, such method may fail to perform satisfying clustering for complicated data.

F. Yin, J. Wang, and C. Guo (Eds.): ISNN 2004, LNCS 3173, pp. 834–839, 2004.

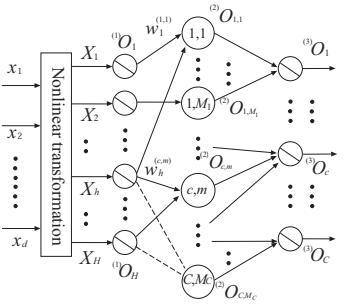

Fig. 1. Structure of LLGMN

In this paper, a novel hierarchical clustering method is proposed. In this method, a probabilistic NN that is derived from the Gaussian mixture model (GMM), called a log-linearized Gaussian mixture network (LLGMN) [2], is utilized for partition at each non-terminal node. The proposed method can estimate the number of terminal nodes corresponding to the number of classes according to the statistical information obtained solely from the training data.

2 LLGMN

2.1 Structure of LLGMN

The structure of LLGMN is shown in Fig.1. First, an input vector $\boldsymbol{x} \in \Re^D$ is converted into a modified vector $\boldsymbol{X}$ as follows:

$$\boldsymbol{X} = [1, \boldsymbol{x}^T, x_1^2, x_1x_2, \cdots, x_1x_D\ , x_2^2, x_2x_3, \cdots, x_2x_D, \cdots, x_D^2]^T\ . \tag{1}$$

The first layer consists of H units corresponding to the dimension of $\boldsymbol{X}$, and an identity function is used for an activation function of each unit. The variable ${}^{(1)}O_h\ (h = 1, \cdots, H)$ denotes the output of the hth unit in the first layer. Each unit in the second layer receives the output ${}^{(1)}O_h$ weighted by coefficients $w_h^{(c,m)}(c = 1, \cdots, C; m = 1, \cdots, M_c)$. C denotes the number of classes and M_c is the number of components belonging to class c. The relationship between the input ${}^{(2)}I_{c,m}$ and the output ${}^{(2)}O_{c,m}$ of unit $\{c, m\}$ in the second layer can be defined as

$${}^{(2)}I_{c,m} = \sum_{h=1}^{H} {}^{(1)}O_h w_h^{(c,m)}\ , \tag{2}$$

$${}^{(2)}O_{c,m} = \frac{\exp[{}^{(2)}I_{c,m}]}{\sum_{c'=1}^{C}\sum_{m'=1}^{M_{c'}}\exp[{}^{(2)}I_{c',m'}]}\ , \tag{3}$$

where $w_h^{(C,M_C)} = 0$. The relationship between the input ${}^{(3)}I_c$ and the output is described as,

$$ {}^{(3)}O_c = {}^{(3)}I_c = \sum_{m=1}^{M_c} {}^{(2)}O_{c,m} \ . \tag{4} $$

The output of the third layer ${}^{(3)}O_c$ corresponds to the posterior probability $P(c|\boldsymbol{x})$ of class c.

2.2 Supervised Learning Algorithm [2]

Consider a training set $\{\boldsymbol{x}^{(n)}, \boldsymbol{T}^{(n)}\}$ $(n = 1, \cdots, N)$, where $\boldsymbol{T}^{(n)} = \{T_1^{(n)}, \cdots, T_C^{(n)}\}$. If the input vector $\boldsymbol{x}^{(n)}$ belongs to class c, $T_c^{(n)} = 1$, and $T_{c'}^{(n)} = 0$ for all of the other class c'. An energy function according to the minimum log-likelihood training criterion can be derived as:

$$ J_{SV} = -\sum_{n=1}^{N}\sum_{c=1}^{C} T_c^{(n)} \log {}^{(3)}O_c^{(n)} \ . \tag{5} $$

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

$$ \Delta w_h^{(c,m)} = -\eta \sum_{n=1}^{N} \frac{\partial J_{SV}^n}{\partial w_h^{(c,m)}} \ , \tag{6} $$

$$ \frac{\partial J_{SV}^n}{\partial w_h^{(c,m)}} = ({}^{(2)}O_{c,m}^{(n)} - \frac{{}^{(2)}O_{c,m}^{(n)}}{{}^{(3)}O_c^{(n)}} T_c^{(n)}) X_h^{(n)} \ , \tag{7} $$

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

3 Hierarchical Clustering

The divisive clustering starts from a single cluster, and terminates when a termination criterion has been satisfied, so that the training data are divided into the appropriate number of clusters. At each non-terminal node, LLGMN is used to achieve binary splits. Even for data of complicated distributions, interpretable clustering can be made after a nested series of binary splits. In this section, after the description of the proposed unsupervised learning algorithm of LLGMN, division validation according to the statistical properties of the training data and pruning law are explained.

3.1 Unsupervised Learning Algorithm

Given the number of classes, C, the entropy used as cost function is defined as:

$$ J_{SO} = -\sum_{n=1}^{N}\sum_{c=1}^{C} {}^{(3)}O_c^{(n)} \log {}^{(3)}O_c^{(n)} \ , \tag{8} $$

where N is the number of total data. The proposed unsupervised learning algorithm modifies weights by minimizing Eq. (8), However, for some initial weights, the LLGMN may be trained to cluster all training data into one class, and cost function, J_{SO}, may converge to such a local minimum. Therefore, in the proposed method, the initialization of the weights is carried out to prevent the LLGMN to converge to one of the local minima, and the number of classes is restricted to two.

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 total training data set $\boldsymbol{A}$ according to the following equation,

$$(\boldsymbol{x}_1, \boldsymbol{x}_2) = \underset{\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \in \boldsymbol{A}}{\operatorname{argmax}} \left(||\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}|| \right) . \tag{9}$$

Then the set $\boldsymbol{B}$, which means the set of the utilized data, is set as $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$. Assuming that $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are labeled with C_1 and C_2, respectively. Training of LLGMN is performed using the supervised learning rule [2] in order to classify $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ into C_1 and C_2 respectively. Then, with the initialized weights, unsupervised learning of the LLGMN is performed using the set $\boldsymbol{B}$. The mean values of $\bar{\boldsymbol{x}}_{C_1}$ and $\bar{\boldsymbol{x}}_{C_2}$ are calculated using the training data clustered into C_1 and C_2, respectively. One datum $\boldsymbol{x} \in \boldsymbol{A} - \boldsymbol{B}$, from which the distance to either $\bar{\boldsymbol{x}}_{C_1}$ or $\bar{\boldsymbol{x}}_{C_2}$ is the smallest, is added into the set $\boldsymbol{B}$. Then, modification of the weight $\Delta w_h^{(c,m)}$ is defined as:

$$\Delta w_h^{(c,m)} = -\eta \frac{\partial J_{SO}^{(n)}}{\partial w_h^{(c,m)}} , \tag{10}$$

$$\frac{\partial J_{SO}^{(n)}}{\partial w_h^{(c,m)}} = -(J_{SO} - \log {}^{(3)}O_c^{(n)})^{(2)}O_{c,m}^{(n)} X_h^{(n)} . \tag{11}$$

After training with a pre-defined number of times, another training datum is selected from the set $\boldsymbol{A} - \boldsymbol{B}$ and added into the set $\boldsymbol{B}$. This step of training repeats, until all the training data is added into $\boldsymbol{B}$, that is to say, $\boldsymbol{B} = \boldsymbol{A}$.

3.2 Division Validation

With the proposed method, unnecessary splits may occur when the hierarchy of the tree becomes too deep. In this method, cross-validation is adopted and the posterior probabilities of the validation data is utilized to determine whether to split a node or not. First, the validation data is prepared and the entropy $H(\boldsymbol{x})$ is defined as:

$$H(\boldsymbol{x}) = -\sum_{c=1}^{C} {}^{(3)}O_c^{(n)} \log {}^{(3)}O_c^{(n)} . \tag{12}$$

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

$$H_E = \frac{1}{|\boldsymbol{N}_c|} \sum_{\boldsymbol{x}^{(n)} \in \boldsymbol{N}_c} H(\boldsymbol{x}^{(n)}) , \tag{13}$$

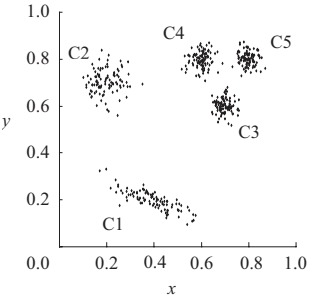

Fig. 2. Examples of artificial data

where N_c stands for the set of validation data belonging to the node under consideration, and $|N_c|$ is the number of validation data in N_c. If H_E is higher than a threshold H_T, splitting of the corresponding node is terminated. On the other hand, if all validation data of the node in consideration are 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 split of this node must be terminated, since division is impossible. With this validation, a classification tree can be constructed based on the statistical properties of the training data, and can cluster complicated data into a proper number of classes.

3.3 Pruning Law

In the proposed method, outliers are always classified into some terminal nodes (clusters) separated from other major clusters. Especially, when the hierarchy of the tree grows too large, the influence of outliers becomes prominent because of a decrease of the number of training data in 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 of pruning. If the ratio of the number of training data in a terminal node to the total training data number is lower than 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 clustering may not increase corresponding to the number of outlier data.

3.4 Experiments

Numerical simulations were carried out in order to verify the proposed method. The feature data is illustrated in Fig. 2: There are 2-dimensional data $\boldsymbol{x} \in \Re^2$, and generated from five classes, C_i $(i = 1, 2, \cdots, 5)$. Each class consists of one normal distribution. 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, threshold of entropy H_T is set as 0.2, threshold of pruning α_t as 0.01, learning rate η as 0.01, and training times in each addition of training data as 100. The classification tree starts from the root node, where training data are divided into two nodes at each non-terminal node, and finally, a hierarchical tree is constructed from five terminal nodes. To validate the generalization ability, 300 samples for each class that are not used in training process were clusterd, and the discrimination rate for 20 independent trials was $98.5 \pm 0.64\%$. It can be found that the proposed method can estimate the number of classes and achieve high classfication rate.

4 Conclusion

In this paper, to deal with the discrimination problem of ambiguous teacher signals, a hierarchical clustering method was proposed. In this method, entoropy of the LLGMN's outputs at each node are used as the termination criterion, and unnecessary splits in the structure of the classification tree can be avoided, so that the proposed method can make an interpretable and reasonable partition of the training data according solely to its statistical characteristics.

In future works, we would like to carry out discrimination experiments on various data and to examine the influence of the parameters to the clustering result. Furthermore, we would like to establish an improved method that determines the value of theresholds such as α_T automatically.

References

1. Hiraiwa, A., Shimohara, K., Tokunaga, Y.: EMG Pattern Analysis and Classification by Neural Network. IEEE International Conference on Syst., Man and Cybern., (1989) 1113–1115
2. Tsuji, T., Fukuda, O., Ichinobe, H., Kaneko, M.: A Log-linearized Gaussian Mixture Network and its Application to EEG Pattern Classification. IEEE Trans. on System, Man and Cybernetics-Part C: Applications and Reviews, Vol. 29., No. 1. (1999) 60–72
3. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1974)
4. Ward, J.H.: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association, Vol. 58., No. 301. (1963) 235–244
5. Kohonen, T.: Self-organization and Associative Memory. Third Edition, Springer-Verlag, Berlin (1994)
6. Terashima, M., Shiratani, F., Yamamoto, K.: Unsupervised Cluster Segmentation Method Using Data Density Histogram on Self-organizing Feature Map. IEICE Transactions on Information and Systems, PT. 2, Vol. J79., No. 7., (1996) 1280–1290 (in Japanese)