International Journal of Fuzzy Logic and Intelligent Systems, vol. 6, no. 1, March 2006 pp. 10-14

An Optimal Clustering using Hybrid Self Organizing Map

Sung-Hae Jun

Department of Bioinformatics & Statistics, Cheongju University
360-764, Chungbuk, Korea

Abstract

Many clustering methods have been studied. For the most part of these methods may be needed to determine the number of clusters. But, there are few methods for determining the number of population clusters objectively. It is difficult to determine the cluster size. In general, the number of clusters is decided by subjectively prior knowledge. Because the results of clustering depend on the number of clusters, it must be determined seriously. In this paper, we propose an efficient method for determining the number of clusters using hybrid self organizing map and new criterion for evaluating the clustering result. In the experiment, we verify our model to compare other clustering methods using the data sets from UCI machine learning repository.

Key Words : Optimal Clustering, Hybrid Self Organizing Map, CCVP

1. Introduction

Automatic determination of the number of population clusters is needed in the clustering like K-means algorithm, hierarchical clustering method, etc. Usually we have determined the number of clusters subjectively. In this paper, we propose a method for automatic determination of the number of clusters using Hybrid Self Organizing Map(HSOM) based fuzzy clustering. That is, the SOM, Bayesian learning, and fuzzy set logic are used in proposed algorithm for decision of optimal number of clusters. The existing methods have had an uncertainty because they have determined the number of clusters with subjectivity. One of the gates to eliminate the uncertainty is through the fuzzy set theory[13]. If X is a collection of objects denoted generally by x, then a fuzzy set A in X consists of a set of x and its membership function. The membership function can be expressed by values from 0 to 1 as the degree of truth that maps X to A. However it is difficult to choose a suitable form for the membership function. Nowadays, it is common to determine the membership function subjectively. Then this may make the problems more ambiguous in the machine learning, which should resolve the uncertainty. In this paper, we propose an objective method to select the membership function for determining the number of clusters by heuristic approach using HSOM. For illustration of our method, we consider examples with comparing other clustering methods which are the SOM, K-means algorithm and statistical clustering methods using the data sets from UCI machine learning repository.

2. Optimal Clustering

2.1 Determining the number of clusters

The cluster is a set of adjacent objects in training data. Objects in the same cluster have close similarity and objects in other clusters have dissimilarity. We use distance as a measure of similarity between objects. The first problem to consider in clustering is to determine the number of clusters. K-means method requires an initial number of clusters and hierarchical clustering technique also requires an optimal number of clusters for stopping clustering process[4]. But it is hard to find any objective algorithm to determine the initial cluster size, and most of them are determined subjectively. So we propose a HSOM for fuzzy clustering algorithm to determine the optimal cluster size.

2.2 Fuzzy Clustering

Let X is a nonempty set and x is an element of X. A fuzzy set A is defined as the following[14].

$$A = \{ (x, \mu_A(x)) | x \in X \} \tag{1}$$

where $\mu_A(x)$ is a membership function that expresses a degree of inclusion of x into A. In this paper, fuzzy set is used to determine the cluster size in clustering[2]. These membership functions of fuzzy set for clustering are computed repeatedly by HSOM from given training data. That is, X becomes a set of all possible clusters and A becomes a fuzzy set of appropriate cluster size. $\mu_A(x)$ is a membership function for each possible cluster size. Therefore, we decide the element with the largest membership function in fuzzy set A to optimal cluster size.

3. A Hybrid Self Organizing Map

3.1 Self Organizing Map

The Kohonen's networks have two models. These are SOM and LVQ(learning vector quantization)[7]. We use SOM in this paper because SOM is a neural network model for unsupervised learning[5]. Though SOM requires a size of feature maps, it may not need the number of clusters. The

Manuscript received Feb. 14, 2006; revised Mar. 10, 2006.

SOM algorithm can be expressed in the following steps[5].

Step1: Initialization

Choose random values for the initial weights.

Step2: Winner finding

Find the winner neuron *j** at time k, using the minimum distance criterion

$$j^* = \arg\neg A_j \| x(k) - w_j \| , \quad j=1,\ldots,N^2$$

where $x(k)$ represents the kth input pattern and $\| . \|$ is the Euclidean norm.

Step3: Weight updating

Adjust the weights of the winner and its neighbors, using the following rule,

$$w_j(k+1) = \begin{cases} w_j(k) + \eta(k)(x(k) - w_j(k)) & \text{if } j \in N_j(k) \\ w_j(k) & o.w. \end{cases}$$

where $\eta(k)$ is a positive constant and $Nj*(x)$ is the neighborhood set of the winner neuron $j*$ at time k.

Repeat: until given conditions satisfaction.

3.2 Hybrid SOM

We use Bayesian learning approach for proposed algorithm[6]. The Bayesian approach assigns a degree of plausibility to any proposition, hypothesis, or model. The Bayesian learning starts from the following Bayes' rule.

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \tag{2}$$

where *M* is a parameterized model and *D* represents a data set. The use of priors is the strength of Bayesian approach, since it allows incorporating prior knowledge and constraints into the modeling process. Using the Bayes' rule with a chosen probability model means that the data, *D* affect the posterior inference only through *P*(*D*|*M*) which is called the likelihood function. Bayes' rule can now be used to combine the information in the data with the prior probability. In particular, the interest is likely to focus on the posterior probability. To make a decision about new data, often called predictive inference, we follow a similar logic. Before the data, *D* has a following form.

$$P(D) = \int P(D, M)dM = \int P(M)P(D|M)dM \tag{3}$$

This is the prior predictive distribution. The distribution of $\mathcal{D}$ is called the posterior predictive distribution, posterior because it is conditional on the training data, *D* and predictive because it is a predictive for a new data $\mathcal{D}$.

$$P(\mathcal{D}|D) = \int P(\mathcal{D}|M)P(D|M)\, d\,M \tag{4}$$

This displays the posterior predictive distribution as an average of conditional predictions over the posterior distribution of *M*. In conclusion, this bayesian learning consists of three components which are prior probability distribution, likelihood function and posterior probability distribution. The prior probability distribution has the information of past data or initial knowledge of model. We use it to represent past training results. Current training data are appeared on likelihood function. We get posterior probability distribution by using prior probability distribution and likelihood function. The posterior distribution is used to decide how to do for a given problem.

Each node of the output layer achieves clustering by competitive learning from training data. Each object crisply belongs to only one exclusive cluster after the last training. And the clustering result is only one type because the weights have fixed values in nodes of SOM after final training. This result is usually not optimal[1],[10],[11],[12] and it is impossible to repeat the different experiments to determine membership function of fuzzy set. In this paper, we get a fuzzy set with repeated experiments by using Bayesian inference[3],[9] that consists of prior probability distribution, posterior probability distribution, and likelihood distribution to SOM. The proposed HSOM updates parameters of probability distribution without having the fixed values of weights on each node of output layer. This strategy makes it possible to create the membership function by performing repeated experiments with same data to get different results. The proposed method does not always offer same results for the same training data because it uses a random number from the last updated distribution for clustering. The membership function of fuzzy set is determined by Bayesian learning[6] based SOM that computes a posterior by combining prior and likelihood. The proposed algorithm in this paper is composed of four phases. The first one is an initial phase. In this phase, standardization for the input data is performed to use the Euclidean distance that computes distance input data and weights which can be used as a measurement to determine winner node. Generally SOM is normalized from 0 to 1 while the proposed algorithm changes the input data to the standardized data that follow the Gaussian distribution of mean 0 and variance 1 to combine proposed distribution. The size of feature maps is decided in this phase. This decision can be subjective. But it is relatively objective compared to the K-means or hierarchical clustering that requires an initial number of clusters in advance. For example, if the size of the feature maps is determined as 5*5, the results of clusters are from 1 to 25 optimally. Of course if the size of feature maps grows, the number of the optimal cluster can grows accordingly. But the advantage of SOM is that it allows objective clustering without exact information on clustering. We determine prior probability distribution of the weights of the nodes in feature maps. A Gaussian distribution with mean 0 and variance 1 is used because input data are standardized and we use an Euclidean distance as a similarity measure. In second phase, the distance between input data and weights is measured and the winner node is determined as minimum distance node. Next phase is the parameters updating step of weights distribution. Yet the parameters update of weights distribution is limited to the winner node. This learning is repeated until the given stopping rules are satisfied. Generally the given sopping rules are determined by the number of iteration of learning data and the tolerance that has update range for the parameters of weights distribution. In the last phase, we find fuzzy set about optimal number of clusters through repeated experiments using the final updated weights distribution of feature maps. Figure 1 shows. the conception of ours.

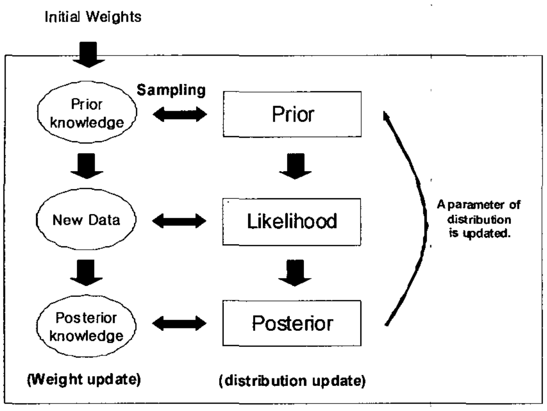

Fig. 1. HSOM learning

We also summarize HSOM by pseudo code in the following.

```
Step1: Initialize
  (n: data size, p: the dimension of input vectors)
  Normalization of input vectors
  $x_i = (x_{i1}, \cdots, x_{ip})$ represents the ith input pattern
  $x_i^{normal} = \left( \frac{x_{i1} - \mu_1}{\sigma_1}, \cdots, \frac{x_{ip} - \mu_{ip}}{\sigma_p} \right) = (x_{i1}^{normal}, \cdots, x_{ip}^{normal})$
  $x_i^{normal} \sim N(0, 1)$, $(i = 1, \ldots, n)$: likelihood
  1.2 Initialize the weights vectors: Prior of weights
    1.2.1 determine the distribution type of $f(\cdot)$
    $f(\cdot)$ is any probability density function(pdf)
    $w \sim f(\theta)$
    optionally, $\theta \sim g(\phi)$: $\phi$ is the hyper-parameter of $\theta$,
    $g(\cdot)$ is also pdf
Step2: Determine winner node
  (m: feature map dimension)
  2.1 Weights sampling from current prior
  2.2 Compute the $dist(x_i^{normal}, w_j)$
      (Euclidean distance of $x_i^{normal}$ and $w_j$)
      $dist(x_i^{normal}, w_j) = \sqrt{(x_{i1}^{normal} - w_{j1})^2 + \cdots + (x_{ip}^{normal} - w_{jp})^2}$
      $(i = 1, \ldots, n)(j = 1, \ldots, m^2)$
  2.3 Determine winner node
      $w_k$ is winner node if $dist(x, w_k) < dist(x, w_j)$,
      $(j = 1, \ldots, m^2)$, that is, $w_k = \arg\neg A_j \{dist(x, w_j)\}$
Step3: Update distribution of weights
  3.1 Compute posterior of winner node using Bayes' rule
  3.2 Replace current posterior by new prior
  Repeat Step2 and Step3 until given conditions are satisfied
Step4: Extract Fuzzy Set for the number of Clusters
  4.1 Repeat experiments until given number
  4.1 Determine the membership function of fuzzy set
```

Before trying to the experiment, we initialize the weight value of feature maps. By Gaussian distribution with mean μ_w and variance σ_w^2, the weight value of each feature node is generating number from this distribution.

$$w_i \sim N(\mu_w, \sigma_w^2),$$
$$f(w_i | \mu_w, \sigma_w^2) = \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(- \frac{(w_i - \mu_w)^2}{2\sigma_w^2} \right) \tag{5}$$

μ_w and σ_w^2 are determined by 0 and 1 respectively. We use it to match the scale of input data and weights because the proposed algorithm computes the Euclidean distance between each weight from this distribution and standardized input data for determining the winner node. In our algorithm, the input vectors are standardized to Gaussian random sample with (0, 1) as parameters. Then the likelihood distribution of each input value, x is a Gaussian distribution as (6).

$$x \sim N(w, \sigma_x^2),$$
$$l(x|w) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(- \frac{(x - w)^2}{2\sigma_x^2} \right) \tag{6}$$

Also, the mean and the variance in (6) are 0 and 1. The weight distribution of winner node is updated by Bayesian learning. The result of Bayesian learning is a probability distribution over model parameters that expresses our beliefs regarding the hoe likely the different parameters values are. The initial distribution of weight is prior distribution. We update this prior distribution to posterior distribution using Bayes'rule.

$$P(w|x_1, \cdots, x_p) = \frac{P(x_1, \cdots, x_p | w) P(w)}{P(x_1, \cdots, x_p)} \propto L(w|x_1, \cdots, x_p) P(w) \tag{7}$$

The posterior distribution combines the likelihood distribution, which contains the information about x derived from observation, with the prior, which contains the information about w. We compute a posterior probability distribution as (8), using Gaussian prior and Gaussian likelihood.

$$\begin{aligned} P(w|x) &\propto f(w) l(x|w) \\ &\propto \frac{1}{\sqrt{2\pi}\sigma_w} \exp\left(- \frac{(w - \mu)^2}{2\sigma_w^2} \right) \\ &\times \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(- \frac{(x - w)^2}{2\sigma_x^2} \right) \\ &\propto \exp\left(- \frac{\left(w - \frac{\sigma_x^2 \mu + \sigma_w^2 x}{\sigma_x^2 + \sigma_w^2} \right)^2}{2 \frac{\sigma_x^2 \sigma_w^2}{\sigma_x^2 + \sigma_w^2}} \right) \end{aligned} \tag{8}$$

Current posterior distribution is used for the next prior distribution. In next training we find the weight value of output node is generated from this prior probability distribution. The updating processes are continued until satisfying stopping conditions. The stopping conditions are no noticeable change to feature map has occurred and predefined iteration size of total training data and given iteration of training data. We get the final updated weight distribution of feature node by Bayesian learning and SOM.

3.3 New Criterion for Evaluating the Clustering Result

A good clustering will produce high quality clusters with high intra-cluster similarity and low inter-cluster similarity[4]. So, we propose a new criterion for clustering on the ground of above good clustering. This criterion is composed of two components which were the variance of objects in clusters and the penalty of increasing the number of clusters. We call our new criterion to clustering criterion based on variance and penalty(CCVP). Our CCVP measure is defined as the following.

$$CCVP_M = \frac{1}{M}\sum_{i=1}^{M}\overline{v}_i + \frac{1}{\overline{V}_M}M \qquad (9)$$

In the above equation, M is the number of clusters, $\overline{v}_i$ is the average of variance of objects in the ith cluster. $\overline{V}_M$ is the variance of M clusters. This is defined as the following.

$$\overline{V}_M = \frac{1}{M-1}\sum_{j=1}^{M}(c_j - \overline{c})^2 \qquad (10)$$

In the equation (10), the c_j is the center of jth cluster and $\overline{c}$ is the average of the centers of M clusters. The smaller the CCVP value is, the better the clustering result is.

4. Experimental Results

For the experiments, we use Iris plants, Glass identification, and Abalone data in UCI machine learning repository[10]. The summary information of these data sets is shown in following table.

Table 1. Summary of Training Data

Data set	# of instances	# of labels	# of attributes
Iris	150	3	4
Glass	214	7	9
Abalone	4177	29	8

In above table, the # of labels is the number of labels in target variable. And the # of attributes is the number of input attributes for clustering. By our proposed method, we get the following fuzzy clustering results.

In figure 2, the Iris plants data have 3 as the optimal number of clusters by proposed HSOM. So, the value of membership function is the largest.

In figure 3, the Glass identification data have 6 as the optimal number of clusters by proposed HSOM.

In figure 4, the Abalone data have 20 as the optimal number of clusters by proposed HSOM. Next we verify our model with other clustering methods. The CCVP measure is good when it is smaller. And the clustering is good when the SD(standard deviation) is smaller. Because the smaller SD of clustering is the more similar objects of data are. We use the k of K-means clustering and the stopping cluster size of hierarchical clustering by the number of labels of target variable.

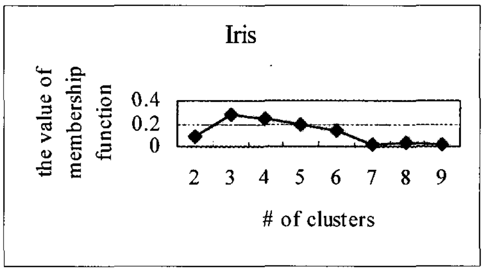

Fig. 2. Clustering result: Iris plants

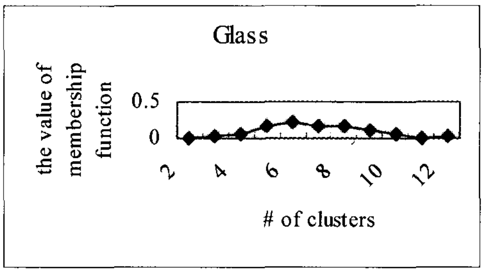

Fig. 3. Clustering result: Glass identification

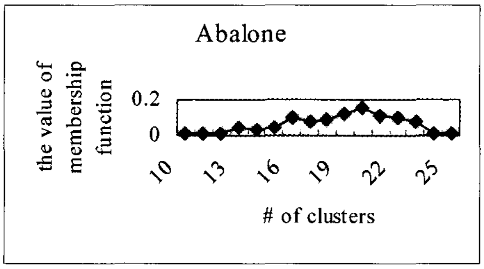

Fig. 4. Clustering result: Abalone

Table 2. Clustering results of comparative models

Data set	Methods	# of clusters	CCVP	
			Mean	S.D.
Iris	SOM	5	0.017	0.146
	K-means	3	0.093	0.583
	Hierarchical	3	0.121	0.912
	HSOM	3	0.002	0.058
Glass	SOM	11	0.184	0.364
	K-means	7	0.312	0.986
	Hierarchical	7	0.498	1.014
	HSOM	6	0.105	0.215
Abalone	SOM	24	2.515	6.311
	K-means	29	4.319	11.358
	Hierarchical	29	5.914	12.984
	HSOM	20	1.313	4.560

The above result showe the CCVP mean and CCVP SD of HSOM is the smallest among comparative clustering methods. So, we find the improved performance of HSOM.

5. Conclusion

In this paper, we propose a method to determine the number of clusters in cluster analysis using HSOM and new criterion for evaluating the clustering result. Unlike other comparative methods, we show an objective method of finding fuzzy set using Bayesian learning and SOM. We make a fuzzy set for determining the number of clusters heuristically. In our work, we use Bayesian inference using conjugate prior probability distribution to unsupervised machine learning algorithm. This has the independent assumption between input variables. If we have not independent assumption, we can consider Bayesian computing with Markov Chain Monte Carlo simulation. Bayesian inference with Markov Chain Monte Carlo can be used for complex domain to get more exact results. This calls for considerable computing time. So, we think over the computing time in machine learning with Markov Chain Monte Carlo. It will be remained for future work.

Reference

[1] C. M. Bishop, M. Svensen, C. K. Williams, "GTM: A Principled Alternative to the Self Organizing Map", Proceedings of ICANN 96, vol. 1112, pp. 165-170, 1996.

[2] D. Dumitrescu, B. Zazzerini, L. C. Jain, *Fuzzy Sets and Their Application to Clustering and Training*, CRC Press, 2000.

[3] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Rudin, *Bayesian Data Analysis*, Chapman & Hill, 1995.

[4] J. Han, M Kamber, *Data Mining Concepts and Techniques*, Morgan Kaufmann, 2001.

[5] T. Kohonen, *Self Organizing Maps*, Second Edition, Springer, 1997.

[6] R. M. Neal, *Bayesian Learning for Neural Networks*, Springer, 1996.

[7] A. S. Pandya, R. B. Macy, *Pattern Recognition with Neural Networks in C++*, IEEE Press, 1995.

[8] M. J. Park, S. H. Jun, K. W. Oh, "Determination of Optimal Cluster Size Using Bootstrap and Genetic Algorithm", *Journal of Fuzzy Logic and Intelligent Systems*, vol. 13, no. 1, pp. 12-17, 2003.

[9] M. A. Tanner, *Tools for Statistical inference*, Springer, 1996.

[10] UCI Machine Learning Repository, ics.uci.edu/~mlearn

[11] A. Utsugi, "Topology selection for self-organizing maps, Network", *Computation in Neural Systems*, vol. 7, no. 4, pp. 727-740, 1996.

[12] A. Utsugi, "Hyperparameter selection for self-organizing maps", *Neural Computation*, vol. 9, no. 3, pp. 623-635, 1997.

[13] L. Zadeh, "Fuzzy Sets", *Information and Control*, 1965.

[14] H. J. Zimmermann, *Fuzzy Set Theory and its Applications*, Third Edition, 1996.

Sung-Hae Jun

He received the BS, MS, and PhD degrees in department of Statistics, Inha University, Korea, in 1993, 1996, and 2001. He is currently Assistant Professor in department of Bioinformatics & Statistics, Cheongju University, Korea. He is also PhD candidate of Computer Science, Sogang University, Korea. He has researched machine learning and evolutionary computing.

Phone : +82-43-229-8205
Fax : +82-43-229-8432
E-mail : shjun@cju.ac.kr