International Journal of Fuzzy Logic and Intelligent Systems, vol. 8, no. 2, June 2008 pp. 116-120

Improvement of Self Organizing Maps using Gap Statistic and Probability Distribution

Sung-Hae Jun

Department of Bioinformatics & Statistics, Cheongju University, 360-764 Chungbuk, Korea
shjun@cju.ac.kr

Abstract

Clustering is a method for unsupervised learning. General clustering tools have been depended on statistical methods and machine learning algorithms. One of the popular clustering algorithms based on machine learning is the self organizing map(SOM). SOM is a neural networks model for clustering. SOM and extended SOM have been used in diverse classification and clustering fields such as data mining. But, SOM has had a problem determining optimal number of clusters. In this paper, we propose an improvement of SOM using gap statistic and probability distribution. The gap statistic was introduced to estimate the number of clusters in a dataset. We use gap statistic for settling the problem of SOM. Also, in our research, weights of feature nodes are updated by probability distribution. After complete updating according to prior and posterior distributions, the weights of SOM have probability distributions for optimal clustering. To verify improved performance of our work, we make experiments compared with other learning algorithms using simulation data sets.

Key Words : Self Organizing Maps, Number of Clusters, Gap Statistic, Probability Distribution

1. Introduction

Diverse clustering algorithms have been used in machine learning applications. Also, many researchers have studied on the clustering approaches. Self organizing map(SOM) is a tool of neural networks for unsupervised learning[1]. The SOM proposed by Kohonen has been used in various clustering fields such as text mining, customer relationship management(CRM), bioinformatics and so forth[2],[3]. In original SOM, the weights of feature nodes are updated by a learning process depending on training data set. After complete updating process, the final weights of SOM are determined by fixed weight values. So, we get only clustering result. If the result is not optimal, we are not able to get another clustering. To overcome this problem of SOM, some researches were proposed in machine learning fields[4],[5],[6],[7],[8],[9]. These were alternative probabilistic models similar to SOM to settle the problem of SOM. But, they had some limitation because the alternative models were not SOM. On the other hand, some works have been applied to combine probabilistic methods with general SOM directly[10],[11],[12]. They were probability models based on original SOM. Therefore, the models were improved approaches for solving the problems of SOM. In this paper, we propose improvement of SOM using gap statistic and probability distribution. Our improved SOM is able to provide diverse outcomes from the posterior distribution of SOM weights by a given training data set. What is more, our model is able to determine the number of clusters by automatic learning based on gap statistic. In our experimental results, we verify the performances of our model to compare other clustering methods using the data sets from simulation data sets.

Manuscript received Jan. 26, 2008; revised Jun. 4, 2008.

2. Related Works

2.1 SOM

In general, the cluster is a set of adjacent points in given training data. The points in the same group have close similarity and objects in other groups have dissimilarity[13],[14]. We use Euclidean distance as a measure of similarity between points. The first problem considered in clustering is optimal determination of the number of clusters. For example, k-means method needs initial number of clusters and hierarchical clustering technique also requires optimal number of clusters for stopping clustering process[13],[14]. The number of clusters has been mostly determined by the arts of researchers with their subjectively prior information. It is difficult to find objective method to settle the clustering problem. So, we have studied on the problem in SOM case[11],[12],[15]. Also, the following algorithm is for typical SOM[1].

step1) Choose random values for the initial weights

step2) Find the winner node with the smallest values of Euclidean distance measure

step3) Update the weights of winner node (winner takes all)

step4) Repeat until given conditions are satisfied

When we complete the weights updating of SOM, we can determine the number of clusters from the topological result of feature map. But, what we get is only the result of the number of clusters from SOM learning without knowing that the number of

clusters is not optimal. So, in this paper, we propose improved SOM using gap statistic and probability distribution to overcome that problem.

2.2 Gap Statistic

Gap statistic was proposed to estimate the number of clusters by R. Tibshirani et al[16]. The statistic has been used in the clustering results of k-means or hierarchical clustering algorithms. According to the previous simulation study, we knew the good performance of gap statistic technique. Data of gap statistic are shown in the following.

$$(x_{11}, x_{12}, \ldots, x_{1m}), \cdots, (x_{n1}, x_{n2}, \ldots, x_{nm}) \quad (1)$$

Where, n and m represent the sizes of variables and observations. Let G_1, G_2, …, and G_K are K clusters from a given data set. G_k denotes the indices of observations in cluster k, and $n_k=|G_k|$. in gap statistic, the following measure is defined.

$$\tilde{d}_k = \sum_{j,j' \in G_k} d_{jj'} \quad (2)$$

Where,

$$d_{jj'} = \sum_r \left(x_{jr} - x_{j'r}\right)^2 \quad (3)$$

Also, r is the index of variable. So, $\tilde{d}$ is the sum of the pairwise distances for all observations in cluster k. therefore, following W_K is defined.

$$W_K = \sum_{k=1}^{K} \frac{1}{2n_k} \tilde{d}_k \quad (4)$$

When the value of W_K is the smallest, the K is optimal number of clusters for given training data.

2.3 Prior and Posterior Distributions

Prior and posterior distributions are used in Bayesian statistics. Also, Bayesian statistics is based on the following Bayes' theorem[17].

$$p(\theta \mid x_1, \ldots, x_n) = \frac{f(x_1, \ldots, x_n \mid \theta) g(\theta)}{\int f(x_1, \ldots, x_n \mid \theta) g(\theta)} \quad (5)$$

Where x_1, …, x_n is i.i.d.(independent and identical distributed) random vector. Also, $f(x|\theta)$ is a pdf(probability density function). In Bayesian statistics, $f(x|\theta)$ is called as a likelihood function. In above expression, $g(\theta)$ and $p(\theta|x_1, \ldots, x_n)$ are the prior distribution and posterior distribution respectively. So, the updating process of Bayesian learning is performed by the following equivalent statement[18],[19].

Prior distribution $\propto$ Likelihood function×Prior distribution (6)

Where prior distribution is the prior knowledge of given training data. Also, likelihood function has information about given training data. Using posterior distribution, we perform optimal clustering which updates the weights of feature nodes in SOM.

3. Improvement of SOM using Gap Statistic and Probability Distribution

To cluster data points into optimal group, we propose an improved SOM using gap statistic and probability distribution. In this paper, we use Bayesian learning process for probability distribution. That is, a probability distribution over all unknown weights of SOM after Bayesian learning. Bayesian procedure assigns a degree of plausibility to adaptive model. It also is based on the following Bayes' rule[17],[20].

$$P(\theta \mid Y) = \frac{P(Y \mid \theta) P(\theta)}{P(Y)} \quad (7)$$

Where, Y is a vector of training data and θ is a vector of weights of feature map in SOM. The use of priors is strength of Bayesian approach, since it allows incorporating prior knowledge and constraints into the modeling process. Using the rule with a chosen probability model means that the data, D affect the posterior inference only through the $P(Y|\theta)$, is called the likelihood function. The rule can now be used to combine the information in the data with the prior distribution. In this paper, we focus our work into the posterior probability. To make a decision about new data, often called predictive inference, we perform the following expression.

$$p(\theta|Y)=cp(\theta)L(\theta|Y) \quad (8)$$

In above, $L(\theta|Y)$ and $p(\theta)$ are likelihood and prior probability density. The posterior density $p(\theta|Y)$ describes what is known about θ given the data Y. Also, the constant c is able to be computed by the following equation.

$$c=\int p(\theta)L(\theta|Y) \quad (9)$$

In our work, Y represents input data for optimal clustering in SOM. θ is weight vector of nodes of feature maps. So, $p(\theta|Y)$ is a probability of θ given Y. That is, we assign Y into θ with the value of $p(\theta|Y)$ is the largest. The following shows our proposed clustering process.

S1) Given training data (input data)

S2) Determination of the dimension of feature map

S3) Initialization of the weights of nodes in feature map

$$w \sim p(\theta|x) \sim N(0,1)$$

S4) Normalization of the input data

$$\frac{x - \mu_x}{s_x}$$

S5) Set likelihood function based on training data

$$L(\theta|x) \sim N(0,1)$$

S6) Determining the winner node using Euclidean distance

$$d(x,w)=\sqrt{\sum_{i=1}^{n}(x_i-w_i)^2}$$

The winner node has the smallest d(x,w).

S7) Weights update by Bayesian learning

$$p\theta|x)=cp(\theta)L(\theta|x)$$

S8) Update repeating of weights' distributions until stopping conditions which are given number of repeating and the convergence criterion

S9) Determination of the number of clusters using the following gap statistic rule

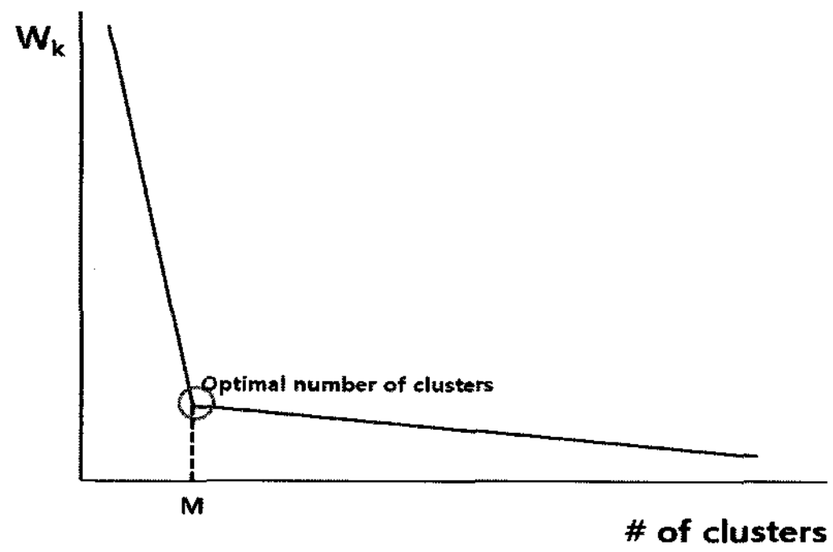

Fig. 1. Gap statistic and number of clusters
x-axis: the number of clusters
y-axis: W_k of gap statistic

We select the number of clusters by the W_k value with a significant fall in the W_k.

S10) Assign all data points into their optimal groups

Using above steps, we are able to get improvement of SOM. From S1 to S8, we perform our SOM which is combined with probability distribution. Also, optimal determination of the number of clusters is performed in S9. Finally the points are assigned into their adaptive cluster in S10.

4. Experimental Results

To verify improved performance of our work, we make experiments using data sets from synthetic data simulation. For usage of the data sets, we generate multivariate random data from finite mixture density(FMD)[21],[22]. FMD is a probability density function as the following form[22].

$$f(x;\pi,\theta)=\sum_{i=1}^{c}\pi_i g_i(x;\theta_i) \tag{10}$$

Where, x, π, and θ are random vector, mixing proportions, and model parameters respectively. Also, density g is parameterized by θ. In FMD, each cluster comes from a population with a different probability distribution[21],[22]. So, we get random data sets from the following expression.

$$f(cluster\ i|x_i)=\frac{\hat{\pi}_i g_i(x_i,\hat{\theta}_i)}{f(x_i;\hat{\pi},\hat{\theta})} \tag{11}$$

In this experiment, we need to generate multivariate random vector x. Based on a d-dimensional vector of standard normal random numbers, the following transformation is performed[22].

$$x_{(d\times 1)}=R^T_{(d\times d)}z_{(d\times 1)}+\mu_{(d\times 1)} \tag{12}$$

Where, z is the standard normal random vector and μ is a mean vector. $R^TR=\Sigma$ is a covariance matrix. Using different Σ s, we get two synthesis data sets which are high and low correlated data. In the following illustration, Σ_{high}, Σ_{middle}, and Σ_{low} are covariance matrices for high, middle, and low correlated data between attributes. The data set with low correlation are independent. The following figure shows the correlation structures of our experimental simulation.

$$\Sigma_{high}=\begin{pmatrix}1\\0.97&1\\0.89&0.78&1\\0.73&0.74&0.81&1\\0.86&0.79&0.90&0.81&1\end{pmatrix}$$

$(a)\ high\quad corr.$

$$\Sigma_{middle}=\begin{pmatrix}1\\0.34&1\\0.45&0.39&1\\0.38&0.35&0.41&1\\0.45&0.38&0.29&0.35&1\end{pmatrix}$$

$(b)\ middle\quad corr.$

$$\Sigma_{low}=\begin{pmatrix}1\\0.03&1\\0.04&0.12&1\\0.11&0.14&0.17&1\\0.09&0.15&0.20&0.15&1\end{pmatrix}$$

$(c)\ low\quad corr.$

Fig. 2. Correlation coefficient matrix

In the above covariance matrices, the number of attributes is four respectively. We generate data sets which have 1000 data points randomly from the multivariate normal distribution with the above covariance matrices. First of all, we make an experiment on optimal determination of the number of clusters

with our process. Using the weight distributions of feature nodes, we are able to generate random vector of the nodes of the feature map. So, we perform from S1 to S8 by repeated random sampling from the weight distributions. We show the results which determine the number of clusters in the following figures. The result of Iris plant database is shown in the below figure. In this experiment, we use 50 repeated random samplings. Following 3 figures express the improved results of our improved SOM which are the experimental results from the synthetic data sets with high, middle, and low correlation coefficients between input variables.

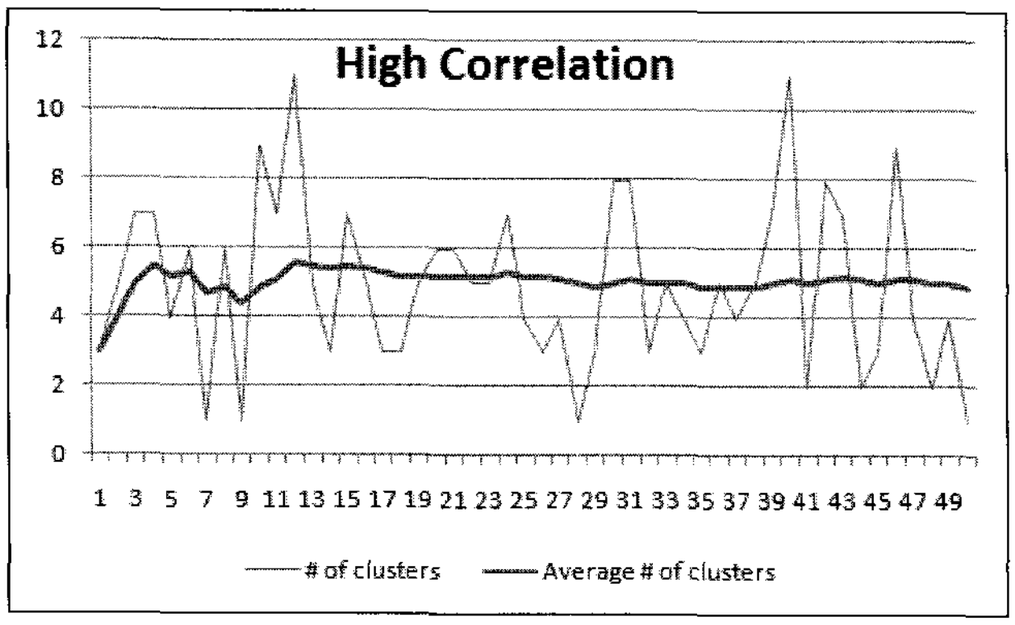

Fig. 3. Clustering result of high correlation

We find the number of clusters of high correlation synthetic data is 5 from the above result. In the figure, the line of # of the clusters represents each result of a random sample from the weight distributions. Our result is shown by the bold line of average # of clusters. In this line, the value of each step from 1 to 50 is computed by averaging previous values. By the result, optimal number of clusters of the data set is determined as 5. This is equal to the number of classes of our synthetic data.

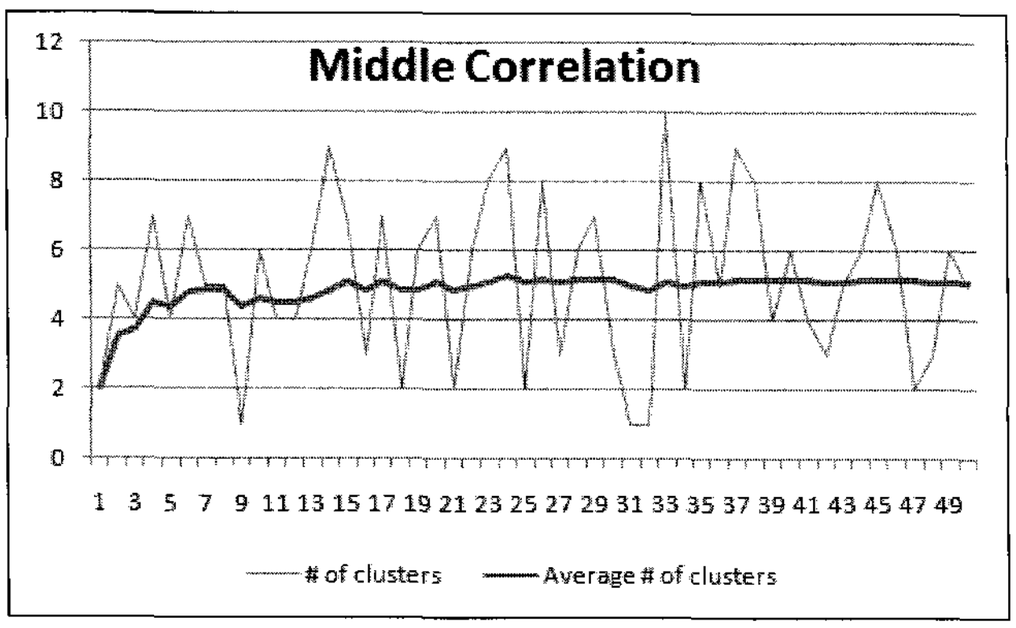

Fig. 4. Clustering result of middle correlation

Similar to the result of the data with high correlation, the heuristic result in synthetic data with middle correlation shows about 5 as optimal number of clusters. In next figure, the result of synthetic data with low correlation is analogous to the heuristics with high and middle correlation coefficients.

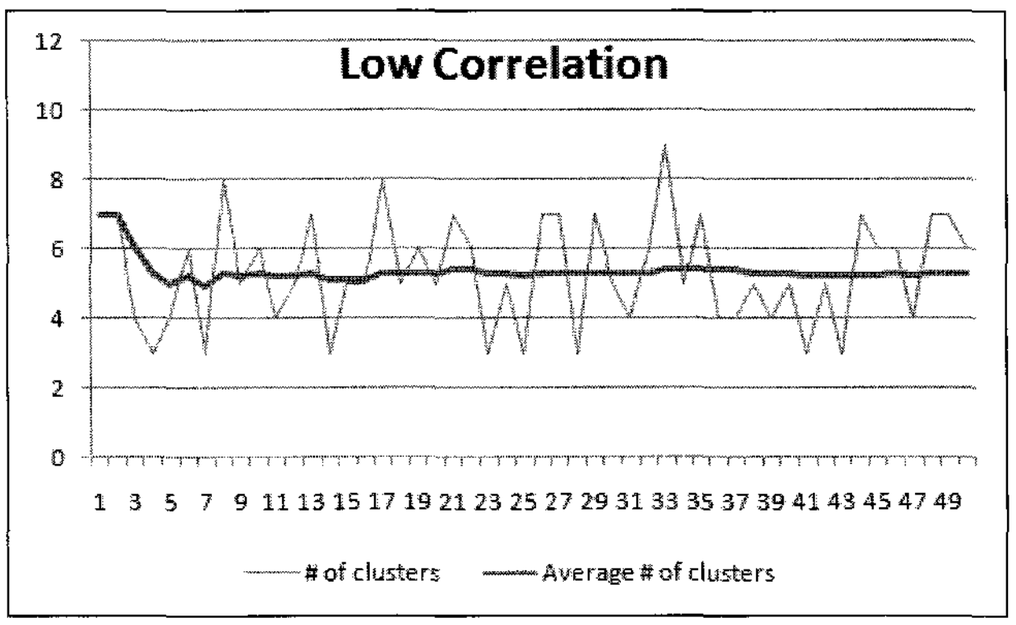

Fig. 5. Clustering result of low correlation

From the above 3 figures of synthetic data sets, we know the variance of the cluster numbers is larger according to increasing correlation coefficient. So, the variance of the cluster numbers in the data with low correlation is the smallest in the synthetic data sets. Next, to verify the improved performance of our research, we compare our study with competitive learning algorithms[2],[3],[23],[24]. The following result is gotten by misclassification rate.

Table 1. Evaluation result of compared algorithms

Methods		Synthesis		
		High	Middle	Low
SOM		0.21	0.20	0.19
SVC		0.15	0.18	0.18
K-means		0.29	0.30	0.28
Hierarchical	Agg.	0.34	0.38	0.29
	Div.	0.41	0.41	0.38
Improved SOM		0.18	0.19	0.16

The result table shows the performance values of our improved SOM are better than others. But, in the experimental result of synthetic data with high correlation coefficient, the evaluative value of SVC is smaller than our SOM. Also, we get the result that the number of clusters of above 3 data sets is 3 using gap statistics according to S9 and S10 in section 3.

5. Conclusions

In this paper, we proposed improvement of SOM using gap statistics and probability distribution for optimal clustering. Our research was a trial to settle the problems of general SOM and alternative SOMs. We verified improved performances of our improved SOM compared with other learning algorithms using synthetic data sets with high, middle and low correlations.

References

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

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

[3] S. Haykin, *Neural Networks*, Prentice Hall, 1999.

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

[5] A. Ngan, S. Thiria, F. Badran, M. Yaccoub, C. Moulin, M. Crepon, Clustering and classification based on expert knowledge propagation using probabilistic self-organizing map(PRSOM): application to the classification of satellite ocean color TOA observations", *Proceeding of IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications*, pp. 146-148, 2003.

[6] D. A. Stacey, R. Farshad, "A probabilistic self-organizing classification neural network architecture", *Proceeding of International Joint Conference on Neural Networks*, vol. 6, pp. 4059-4063, 1999.

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

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

[9] H., Yin, N. M., Allinson, "Bayesian learning for self-organising maps", *Electronics Letters*, vol. 33, issue 4, pp. 304-305, 1997.

[10] S. H. Jun, H. Jorn, J. Hwang, "Bayesian Learning for Self Organizing Maps", *The Korean Journal of Applied Statistics*, vol. 15, no. 2, pp. 251-267, 2002.

[11] S. H. Jun, "An Optimal Clustering using Hybrid Self Organizing Map", *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 6, no. 1, pp. 10-14, 2006.

[12] S. H. Jun, "New Heuristic of Self Organizing Map using Updating Distribution", *Proceeding of the 1st International Conference on Cognitive Neurodynamics - 2007 (ICCN'07) and the 3rd Shanghai International Conference on Physiological Biophysics - Cognitive Neurodynamics (SICPB'07)*, 2007.

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

[14] B. S. Everitt, S. Landau, M. Leese, *Cluster Analysis*, Arnold, 2001.

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

[16] R. Tibshirani, G. Walther, T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistics", *Journal of the Royal Statistical Society*, B, 63, pp. 411-423, 2001.

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

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

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

[20] S. J. Press, *Bayesian Statistics: Principles, Models, and Applications*, John Wiley & Sons, 1989.

[21] W. L. Martinez, A. R. Zartinez, *Computational Statistics Handbook with MATRAB*, Chapman & Hall, 2002.

[22] G. Mclachlan, D. Peel, *Finite Mixture Models*, John Wiley & Sons, 2000.

[23] T. M. Mitchell, *Machine Learning*, McGraw-Hill, 1997.

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

Sung-Hae Jun

He received the BS, MS, and PhD degrees in department of Statistics, Inha University, Korea, in 1993, 1996, and 2001. Also, He received PhD degree in department of Computer Science, Sogang University, Korea in 2007. He is currently Assistant Professor in department of Bioinformatics & Statistics, Cheongju University, Korea. He has researched statistical learning theory and evolutionary algorithms.

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