International Journal of Fuzzy Logic and Intelligent Systems, vol. 8, no. 3, September 2008 pp. 196-201

Improvement of Support Vector Clustering using Evolutionary Programming and Bootstrap

Sung-Hae Jun

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

Abstract

Statistical learning theory has three analytical tools which are support vector machine, support vector regression, and support vector clustering for classifica-tion, regression, and clustering respectively. In general, their performances are good because they are constructed by convex optimization. But, there are some problems in the methods. One of the problems is the subjective determination of the parameters for kernel function and regularization by the arts of researchers. Also, the results of the learning machines are depended on the selected parameters. In this paper, we propose an efficient method for objective determina-tion of the parameters of support vector clustering which is the clustering method of statistical learning theory. Using evolutionary algorithm and bootstrap method, we select the parameters of kernel function and regularization constant objectively. To verify improved performances of proposed research, we compare our method with established learning algorithms using the data sets form UCI machine learning repository and synthetic data.

Key words : Support Vector Clustering, Evolutionary Programming, Bootstrap

1. Introduction

Clustering is to assign the objects into a number of clusters that are internally homogeneous and heterogeneous from group to group. Cluster analysis is a method of unsupervised learning for examining multivariate data with a view to uncovering or discovering clusters of homogeneous observations[1],[2]. Many researchers have used clustering methods for their data exploration and analysis. Numerous clustering works have been researched in the machine learning algorithms. There are some problems in the cluster analysis in spite of the researches. One of the problems is to determine the optimal number of clusters. In *K*-means algorithm, the number of clusters *K* is determined by the art of researchers[3],[4]. Hierarchical clustering methods require a termination condition which is also the number of clusters[1],[5]. The majority of learning algorithms for clustering are not free from this problem[3][4],[6]. So, diverse studies which are objective clustering algorithms to avoid the results according to the problem have been researched. In this paper, we also propose an objective clustering algorithm to settle the problem. Our algorithm is constructed by combining support vector clustering(SVC) with evolutionary programming(EP). SVC is statistical learning theory(SLT) for clustering[7],[8]. Recently using SVC, many clustering works have been performed[9],[10],[11]. The basic concept of SVC is to map data points by Gaussian kernel to high dimensional feature space and to find a sphere with minimal radius that contains most of the mapped data point in the feature space. After mapped back to data space, this sphere is able to separate into several components, each enclosing a separate cluster of points. In this clustering process, SVC is able to determine the number of clusters by support vectors[12]. So, this advantage of SVC is a good idea for clustering works. But, we are confronted with some problems in SVC. They are the selections of kernel parameters and regularization constant. The clustering results of SVC are also depended on the subjective determinations of the kernel and regularization parameters. To settle the problems of SVC, we introduced to combine competitive co-evolving into SVC[13]. The research was a trial for automatic determinations of the parameters of SVC. But, it had a problem which needed heavy computing cost. Because the competitive co-evolving has two evolving steps. So, in this paper, we consider evolutionary programming based on a step and bootstrap to reduce computing cost. Also, using a stochastic search of SVC parameters, we determine the parameters objectively. To verify improved performances of our algorithm, we experiment on data sets from UCI machine learning repository and synthetic data.

2. Research Backgrounds

In this section, we give related works which are SVC, EP, and bootstrap methods in the existing literature. So, we propose improvement of SVC using EP and bootstrap in the following section. SLT is the best currently available theory for statistical estimation and predictive learning[4]. SLT has three learning methods which are support vector machine(SVM), support vector regression(SVR), and SVC for classification, regression, and clustering respectively. From among them, SVC is a good clustering algorithm based on SLT. SVC has been applied to

Manuscript received Feb. 1, 2008; revised Jul. 16, 2008.

solve diverse clustering problems. The basic concept of SVC is to map data points by Gaussian kernel to high dimensional feature space and to find a sphere with minimal radius that contains most of the mapped data point in the feature space. After mapped back to data space, this sphere is able to separate into several components, each enclosing a separate cluster of points. SVC determines the number of clusters by support vectors[12]. This advantage of SVC is a good idea for our clustering study. SVC is a clustering method even though the boundary of the cluster may be shaped arbitrarily[9]. SVC is also a non-parametric clustering algorithm using kernel mapping unlike some clustering algorithms based on parametric model or distance measurement. In the SVC, data points are mapped by means of Gaussian kernel to a high dimensional feature space, where the minimal enclosing sphere is founded. After this sphere is mapped back to data space, it forms a set of contours which enclose the data points. These contours are interpreted as cluster boundaries. Points enclosed by each separate contour belong to the same cluster.

EP is to find a set of optimal behaviors from a space of observable behaviors. EP is not genetic algorithm(GA) because EP emphasizes the development of behavioral models and not genetic models. In evolutionary process, a simulation of adaptive behavior is able to derive EP. We find optimal behaviors from a space of observable behaviors in evolutionary process. The crossover of GA is not implemented in EP. The mutation is only used in EP. EP is another member of evolutionary computing(EC) family. EC is a special type of computing, which draws inspiration from the process of natural evolution. The fundamental of EC relates powerful natural evolution to a particular style of problem solving, that of trial and error[15]. Environment, individual, and fitness of the basic EC are linked respectively problem, candidate solution, and quality of the natural evolution to problem solving. EP is originally developed to simulated evolution as a learning process with the aim of generating artificial intelligence[1],[16]. Intelligence is viewed as the capability of a system to adapt its behavior in order to meet some specified goals in a range of environments. Therefore, EP is typically used for continuous parameter optimization. In this paper, we combine EP into SVC to construct an efficient tool for objective clustering.

Bootstrap method is an assessment tool for the result of statistical data analysis[17]. In the training data, $\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$, its idea is to draw datasets with replacement from the training data randomly. In general, the size of sampled data is equal to the training data. This is done K times to construct K bootstrap datasets. Then the model to each of the bootstrap datasets is refitted, we examine the behavior of the fits over the P replications. Re-sampling technique provides estimates of any statistic. For example, P new samples are drawn with replacement from total training data. The statistic is computed for each new set of data, yielding a bootstrap distribution for the statistic. In this paper, this statistic is considered for the kernel parameters and regularization constant of SLT. The basis assumption of bootstrapping is that the training data are representative of the underlying population. By re-sampling data from the training data, the process of sampling data from the population is mimicked. So, using re-sampling approach of bootstrap methods, we are able to decrease the computing time of machine learning algorithms. In this paper, the size of bootstrap datasets is smaller than the training data size. This is the basic idea of the reduction of the computing cost in learning algorithm. Therefore, to reduce the computing time of our algorithm, we apply re-sampling approach of bootstrap.

3. Improvement of SVC using EP and Bootstrap

In this paper, to overcome the problem of SVC, we propose improvement of SVC using EP and bootstrap. We call our proposed algorithm an evolutionary support vector clustering(ESVC). Using ESVC, we determine the kernel parameters and regularization constant of SVC objectively. That is, ESVC settle the problem of subjective selection for the parameters of SVC. The parameters of kernel functions play an important role in SVC. Though the researches want to determine them automatically, most searching processes are performed by the art of researchers. The regularization constant C is also plays a crucial part in the result of SVC. But, as well as the kernel parameters, it is determined subjectively. In ESVC, we are able to determine the parameters objectively using evolutionary algorithm. Also, using ESVC, we perform optimal clustering. The initial population of ESVC parameters is randomly determined. Good result of clustering has high intra-cluster similarity and low inter-cluster similarity. So, we use this idea to define fitness function. In the following, our ESVC is shown.

Begin (Initialization). In this step, an initial population of kernel and regularization parameters is created by random numbers from Uniform distribution from -1 to 1. Also, we introduce a criterion for evaluating the results of clustering. Let *k*th cluster and its center be defined as the followings.

$$\{x_{k1},x_{k2},\ldots,x_{kn_k}\}\in G_k \qquad (1)$$

Where, G_k is *k*th cluster and $\{x_{k1},x_{k2},\ldots,x_{kn_k}\}$ are its objects. A center of *k*th cluster is shown.

$$c_k=\frac{1}{n_k}\sum_{s=1}^{n_k}x_{ks} \qquad (2)$$

If the total number of clusters is K, then the followings are defined.

$$n_1+n_2+\cdots+n_K=N\ ,\ \ \bar{c}=\frac{1}{K}\sum_{l=1}^{K}c_l \qquad (3)$$

Where, N is the data size and $\bar{c}$ is average of all cluster center values.

Repeat (EP Search). According to the results of this step, we find the cluster boundaries by mapping data points into a high dimensional feature space by a Gaussian kernel which computes the minimal radius enclosing sphere. After mapped back into input space, a set of contours enveloping the input data determines the cluster boundary rule. The shape of the contours is depended on kernel width and regularization constant. The kernel parameter q is determined by evolutionary search. Also, in this step, the regularization parameter C is also found by the following. Firstly using select operator, we construct parents. Next the result of offspring is taken by mutation operator. After evaluating new candidates we select individuals for the next generation. So, in this step, we are able to determine the parameters of kernel function and regularization.

```
BEGIN
Set t=0;
Create an initial population $x=(q_1, \ldots, q_p, C)\in R^{p+1}$
  $(q_1, \ldots, q_p)$; kernel parameters
  C: regularization constant
Repeat Until (stop condition is satisfied)
Mutation: draw $Z_i$ from N(0,1);
    yi(t)=xi(t)+zi for all i∈{1, ..., n};
  If (f(x(t))≤f(y(t))) then x(t+1)=x(t);
            Else x(t+1)=y(t);
  End if
Set t=t+1;
End
```

Where *N(0,1)* is Gaussian distribution with mean 0 and variance 1. In the above, the stop condition has two cases of termination requirements. Firstly, the process is stopped when the total number of fitness evaluations reaches a given limit. Secondly, for a given period of time, until the fitness improvement is remained under a threshold value, our algorithm has been processed. The mutation operator is implemented by adding some random noise drawn from *N(0,1)*. $f(\cdot)$ is the fitness function explained in next step.

Clustering I (Fitness function). In this paper, we introduce a criterion for evaluating efficient clustering. It is composed of two parts which are the variance of points in clusters and the penalty of excessive increasing the number of clusters. We use this criterion for fitness function of ESVC. The criterion is defined as the following.

$$f(i,j,g;G)=\sum_{g=1}^{G}\frac{1}{n_g}\sum_{(i\neq j)\in g}(x_{ig}-x_{jg})^2 \qquad (4)$$

Where, x_{ig} and x_{jg} are points of group g. G is the number of clusters. Also, n_g is the number of points in cluster g. The smaller the $f(\cdot)$ value is, the better the clustering result is.

Clustering II (Assigning points to each cluster). The similarity between centers of clusters is computed in this step. We use the similarity measure as the following[19].

$$S(c_i,c_j)=\sum_{k=1}^{n}\sum_{l=1}^{m}\frac{k(s_{ik},s_{jl})}{m\cdot n} \qquad (5)$$

Where, s_i and s_j are support vectors. For combining two centers, the average affinities among support vectors are used as the threshold which should be researched. So, data point x is assigned by above measure $S(x,\ c_i)$. Also, to overcome a computing cost of proposed algorithm, we use bootstrap method as the following.

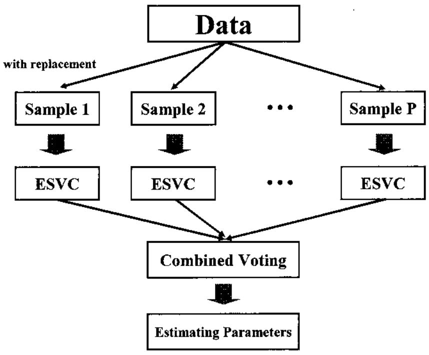

Fig. 1. Bootstrap of ESVC

In above figure, each sample is constructed from original data with replacement sampling. The size of each sample is very smaller than given data. For optimal determinations of the kernel and regularization parameters, we use combined voting of the results from *P* samples.

4. Experimental Results

Using data sets from UCI machine learning repository and synthesis data, we verify our improved performances of ESVC by error rate and clustering time[18]. Firstly we make experiments using Wine recognition, and Thyroid data sets from UCI machine learning repository[19]. These data sets are popular in the machine learning literature[19]. Wine recognition data set is gotten from the results of chemical analysis of wines grown in the same region. In Thyroid data set, the problem is to determine whether a patient referred to the clinic is hypothyroid. The following table shows summary of these data sets.

Table 1. Data sets for UCI ML repository

Data	Attributes	Classes	Data points
Wine	13	3	178
Thyroid	21	3	3772

We also generate multivariate normal data sets for this experiment. So, we need a method for generating multivariate normal random variables. Based on a d-dimensional vector of standard normal random numbers, the following transformation is performed[20].

$$x_{(d\times 1)} = R^T_{(d\times d)} z_{(d\times 1)} + \mu_{(d\times 1)} \quad (6)$$

Where, z is the standard normal random vector and μ is a mean vector. $R^T R = \Sigma$ is a covariance matrix. Using different Σ s, we get two synthesis data sets which are high and low correlated data. In the following, Σ_{high} and Σ_{low} are covariance matrices for high and low correlated data between attributes.

$$\Sigma_{high} = \begin{bmatrix} 1 & & & \\ 0.75 & 1 & & \\ 0.81 & 0.92 & 1 & \\ 0.66 & 0.85 & 0.73 & 1 \end{bmatrix}, \Sigma_{low} = \begin{bmatrix} 1 & & & \\ 0.12 & 1 & & \\ 0.26 & 0.35 & 1 & \\ 0.32 & 0.19 & 0.09 & 1 \end{bmatrix} \quad (7)$$

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. In this section, we compare ESVC with established machine learning algorithms which are SVC, SVM, multi-layer perceptron(MLP), K-nearest neighbor(K-NN), and Gaussian mixture models[1],[20],[21]. ESVC methods with and without bootstrap are compared with other comparative methods. Firstly we show the error rate(%) of misclassified points according to each learning algorithm. For the experiment, given data are divided into training and validation data sets. We use one-third of the given data for the validation set, and other two-thirds for the training[22]. In the following table, T and V represent training and validation data respectively. Also, *low* and *high* are low and high correlated data sets.

Table 2. Error rate(%) of synthetic data

		low		high	
		T	V	T	V
ESVC	With	0.9	1.1	1.3	1.7
	without	0.5	0.8	1.1	1.5
SVC		1.2	1.5	1.9	2.3
SVM		1.4	1.9	2.1	2.6
MLP		2.2	2.8	3.0	3.9
K-NN		2.1	3.1	2.9	3.5
Mixture		1.9	2.7	2.3	3.0

Table 3. Error rate(%) of UCI ML data

Methods		Wine		Thyroid	
		T	V	T	V
ESVC	With	2.4	2.9	1.8	2.1
	without	1.7	2.3	1.5	1.8
SVC		3.4	5.1	2.1	2.4
SVM		3.9	5.6	2.1	2.4
MLP		6.2	8.4	2.6	3.4
K-NN		6.7	8.4	2.8	4.0
Mixture		8.4	11.8	4.0	5.2

From above tables, we find the error rate of ESVC is the smallest among the comparative methods. All results of ESVC with and without bootstrap are shown good performances compared with other popular learning algorithms. Also, the difference between training and validation about the number of error rate of ESVC is smaller than others. So, we are able to show improved performance of ESVC. Next we compute the clustering times of compared methods including ESVC. The experimental result is shown in the following tables.

Table 4. Clustering time(second) of synthetic data

Methods		Synthesis	
		low	high
ESVC	with	21.07	22.31
	without	30.55	22.31
SVC		22.65	23.58
SVM		22.90	24.10
MLP		24.11	26.31
K-NN		20.51	22.45
Mixture		20.09	21.53

Table 5. Clustering time(second) of UCI ML data

Methods		Wine	Thyroid
ESVC	with	32.43	61.00
	without	59.63	108.50
SVC		33.11	62.58
SVM		34.73	63.78
MLP		46.04	69.47
K-NN		29.11	59.98
Mixture		28.05	57.33

For reducing the clustering time of ESVC, we use re-sampling technique of bootstrap methods. In this experiment, we perform 10% random sampling with replacement from given data. It is simple as stated previous bootstrap figure, in this experiment we are able to decrease the computing time of ESVC fairly. From the above results, we know that the clustering time of ESVC with bootstrap is similar level to SVM and SVC.

5. Conclusions

In this paper, we showed improvement of SVC using EP and bootstrap. Our algorithm combined EP into SVC for better clustering result. We called our proposed algorithm ESVC. Using data sets from UCI machine learning repository and synthetic data, we verified improved performances of ESVC compared with popular machine learning algorithms. Also, to reduce the clustering time of ESVC, we applied re-sampling approach of bootstrap into our algorithm. So, we found an effective clustering algorithm which is ESVC. ESVC contributes to modeling decisions for artificial intelligence by theoretical and practical implications. On the theoretical side, our study was applied to automatic determinations of the parameters in SLT which has SVM, SVR, and SVC. Also, the study is extended to model parameter selections of learning algorithms for intelligent modeling. On the practical side, we suggested that ESVC is an efficient clustering tool for customer relationship management(CRM), bioinformatics, information security, and so forth.

However, the study had some limitations. First, the error rates were increased while bootstrap applied to ESVC. Second, because original SVC uses only Gaussian kernel function, we also used Gaussian kernel parameter in ESVC. In future work, to solve our first limitation, more advanced bootstrap methods are needed for increasing accuracy of ESVC. By developing more delicate kernel functions for improving the performances of ESVC, we will be able to overcome the second limitation. Also, using diverse evolving approaches which are evolutionary strategies, co-evolving, competitive evolving, differential evolution, and other evolutionary computation methods, we will settle the limitations of our research.

References

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

[2] P. Giudici, *Applied Data Mining*, Wiley, 2003.

[3] A. L. N. Fred, A. K. Jain, Robust Data Clustering, Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 128-133, 2003.

[4] K. Krishna, K. Narasimha Murty, Genetic K-means algorithm, *Proceeding of IEEE Transactions on Systems, Man and Cybernetics*, Part B, vol. 29, no. 3, pp. 433-439, 1999.

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

[6] L. Y. Tseng, S. B. Yang, Genetic Algorithms for Clustering, Feature Selection and Classification, *Proceeding of International Conference on Neural Networks*, vol. 3, pp. 1612-1616, 1997.

[7] B. Ribeiro, On Data Based Learning using Support Vector Clustering, *Proceeding of the 9th International Conference on Neural Information Processing*, vol. 5, pp. 2516-2521, 2002.

[8] B. Y. Sun, D. S. Huang, Support Vector Clustering for Multiclass Classification Problems, *Proceeding of IEEE Evolutionary Computation Congress*, vol. 2, pp. 1480-1485, 2003.

[9] A. Ben-Hur, D. Horn, H. T. Siegelmann, V. Vapnik, Support Vector Clustering, *Journal of Machine Learning Research*, vol. 2, pp. 125-137, 2001.

[10] J. Lee, D. Lee, An Improved Cluster Labeling Method for Support Vector Clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 461-464, 2005.

[11] P. Ling, Y. Wang, N. Lu, J. Y. Wang, S. Liang, C. G. Zhou, Two-Phase Support Vector Clustering for Multi-Relational Data Mining, *Proceeding of the International Conference on Cyber-worlds*, 2005.

[12] J. Wang, X. Wu, C. Zhang, Support vector machine based on K-means clustering for real-time business intelligence systems, *International Journal of Business Intelligence and Data Mining*, vol. 1, no. 1, pp. 54-64, 2005.

[13] S. H. Jun, K. W. Oh, A Competitive Co-Evolving Support Vector Clustering, *Lecture Note in Computer Science*, vol. 4232, pp. 864-873, 2006.

[14] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[15] A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.

[16] S. H. Jun, Web Usage Mining Using Evolutionary Support Vector Machine, *Lecture Note in Artificial Intelligence*, vol. 3809, pp. 1015-1020, 2005.

[17] A. C. Davison, *Bootstrap methods and their application*, Cambridge University Press, 1997.

[18] J. C. Chiang, J. S. Wang, A Validity-Guided Support Vector Clustering Algorithm for Identification of Optimal Cluster Configuration, *Proceeding of IEEE International Conference on Systems, Man and Cybernetics*, pp. 3613-3618, 2004.

[19] UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html

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

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

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

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