nternational Journal of Fuzzy Logic and Intelligent Systems, vol. 5, no. 4, Dec 2005 pp. 281-285

A Co-Evolutionary Computing for Statistical Learning Theory

Sung-Hae Jun

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

Abstract

Learning and evolving are two basics for data mining. As compared with classical learning theory based on objective function with minimizing training errors, the recently evolutionary computing has had an efficient approach for constructing optimal model without the minimizing training errors. The global search of evolutionary computing in solution space can settle the local optima problems of learning models. In this research, combining co-evolving algorithm into statistical learning theory, we propose an co-evolutionary computing for statistical learning theory for overcoming local optima problems of statistical learning theory. We apply proposed model to classification and prediction problems of the learning. In the experimental results, we verify the improved performance of our model using the data sets from UCI machine learning repository and KDD Cup 2000.

Key Words : Co-Evolutionary Computing, Statistical Learning Theory, Classification, Predictive model

1. Introduction

The learning and evolving have been researched in the diverse intelligent systems[6,11,12,15]. The evolving artificial neural networks have been studied as combining learning and evolving[22,23]. The goal was to find optimal neural network model using genetic algorithm[13]. Some works have shown that the evolving artificial neural networks had more chance to find global optima than the learning artificial neural networks[13,22,23]. But the possibility of local optima was still existed in the evolutionary artificial neural networks using genetic algorithm[12].

To solve them, we consider the Hills' idea which is a host-parasites co-evolutionary approach[7]. It was applied to function optimization problems and was shown to be good in finding optimal and near optimal solutions[8,16]. In Hills' model, the host evolves the defensive ability against parasites from their attacks. The parasites also evolve to circumvent the defense of host. So, the co-evolutions of host and parasites are performed. According to the co-evolving, the host updates its model. So, the approach of co-evolving can settle the local optima problems because of the parasites' evolving.

In this paper, we propose the competitive co-evolution as advanced evolving computing to construct a co-evolutionary computing for statistical learning theory(CEC-SLT). The traditional statistical learning theory(SLT) model by Vapnik is optimized by Lagrange multipliers[19]. But our CEC-SLT model is fitted by the competitive co-evolutionary computing. The Lagrange multipliers are not used in proposed model. So, we can avoid the local optima problems of SLT. Using the data sets from UCI machine learning repository and KDD cup 2000, we verify the performance of the proposed model compared to the existing models.

Manuscript received Nov. 28, 2005; revised Dec. 10, 2005.

2. Related Works

2.1 Evolutionary Computing

Evolutionary computing is a special type of computing, which draws inspiration from the process of natural evolution. The fundamental of evolutionary computing relates powerful natural evolution to a particular style of problem solving, that of trial and error[4]. Environment, individual, and fitness of the basic evolutionary computing were linked respectively problem, candidate solution, and quality of the natural evolution to problem solving.

2.2 Statistical Learning Theory

Our world is changing too fast for us to keep up with it based only on our logics. Due to digitalization, amount of data is increasing continuously. Most of information in huge data are remained undiscovered. So, we need tools for discovering knowledge from large data sets. One of them is Statistics. Statistics is the art of learning from data[17]. The learning is to construct a claim by observing data. The learning procedure contains from this till performing experiments and making conclusion. SLT was proposed and developed by Vapnik[19,20]. SLT is perhaps the best currently available theory for finite sample statistical estimation and predictive learning[2]. SLT is consisted of three types which are support vector machine(SVM), support vector regression (SVR), and support vector clustering(SVC). SVM, SVR, and SVC are respectively classification, prediction, and clustering tools[20,21]. All types of SLT are based on support vector. The approaches of support vector are projection instances into high dimensional spaces, learning linear separators with maximum margin, and learning as optimizing upper bound on expected error. The classification problem of SLT can be restricted to consideration of the two-class problem. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to

produce a classifier that will work well on unseen examples, that is, it generalizes well. Consider the problem of separating the set of training vectors belonging to two separate classes,

$$D = \{(x_1, y_1), \cdots (x_l, y_l)\},\ x \in R^n,\ y \in \{-1, 1\} \quad (1)$$

with the hyperplane,

$$< w, x > + b = 0 \quad (2)$$

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vector to the hyperplane is maximal. There is some redundancy in (2), and without loss of generality it is appropriate to consider a canonical hyperplane[19]. Where the parameters w, b are constrained by,

$$\min_i |< w, x_i > + b| = 1 \quad (3)$$

This incisive constraint on the parameterization is preferable to alternatives in simplifying the formulation of the problem. In words it states that: the norm of the weight vector should be equal to the inverse of the distance, of the nearest point in the data set to the hyperplane. A separating hyperplane in canonical form must satisfy the following constraints,

$$y_i[< w, x_i > + b] \geq 1,\ i = 1, \cdots l \quad (4)$$

The distance d(w,b; x) of a point x from the hyperplane (w,b) is,

$$d(w, b; x) = \frac{|< w, x_i > + b|}{\| w \|} \quad (5)$$

The optimal hyperplane is given by maximizing the margin, subject to the constraints of (4). This approach of SVM is used for the prediction model, SVR.

3. A Co-Evolutionary Computing for Statistical Learning Theory

Evolutionary computing is a special type of computing, which draws inspiration from the process of natural evolution. The fundamental of evolutionary computing relates powerful natural evolution to a particular style of problem solving, that of trial and error[4]. Environment, individual, and fitness of the basic evolutionary computing are linked respectively problem, candidate solution, and quality of the natural evolution to problem solving.

In this paper, we apply competitive co-evolving to SLT. In SLT, our given training data consist of N pairs $(x_1, y_1), \cdots, (x_N, y_N)$, where x denotes the input patterns and y is target variable. In SLT with ε-insensitive loss function, our goal is to find a function f(x) that has at most ε-deviation from the actually obtained targets yi for all the training data, and at the same time, is as flat as possible[18]. In other words, we do not care about errors as long as they are less than ε, but will not accept any deviation larger than this. The ε-insensitive loss function is defined as,

$$M(y, f(x, \alpha)) = L(|y - f(x, \alpha)|_\varepsilon) \quad (6)$$

where we denote,

$$A_\epsilon = \begin{cases} 0 & , if\ A \leq \epsilon \\ A - \epsilon & , o.w. \end{cases} \quad (7)$$

where A is $|y - f(x, \alpha)|$. and α is a positive constant. The loss is equal to 0 if the discrepancy between the predicted and the observed values is less than ε. The case of linear function f is described.

$$f(x) = < w, x > + b \quad (8)$$

where $< \bullet, \bullet >$ denotes the dot product. For the SLT, the Euclidean norm $\| w \|^2$ is minimized. Formally this problem can be written as a convex optimization problem by requiring[19], Analogously to the loss function in [3], we introduce slack variables $\xi_i,\ \xi_i^*$ to copy with otherwise infeasible constraints of the optimization problem.

$$\text{minimize} \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) \quad (9)$$

$$\text{subject to} \quad \begin{cases} y_i - < w, x_i > - b \leq \varepsilon + \xi_i \\ < w, x_i > + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (10)$$

The constant C(>0) determines the trade off between the flatness of f and the amount up to which deviations larger than ε are tolerated. Using a standard dualization method utilizing Lagrange multipliers, the parameters are determined from equation (9) and (10)[5].

In this section, we propose CEC-SLT. It is combined evolutionary computing into SLT. The most widely known type of evolutionary computing is the Genetic algorithm (GA)[4]. GA has provided a analytical method motivated by an analogy to biological evolution[11]. General GA computes the fitness of given environment where is fixed. Distinguished from traditional GA, co-evolving approach is evolu-tionary mechanism of the natural world with competition or cooperation. The organism and the environment including organism evolve together[12]. We apply not cooperation but competition to our proposed co-evolutionary model. Our competitive co-evolving approach is used host-parasites co-evolution. The host and parasites are used for modeling CEC-SLT and training data set. CEC-SLT and training data set are considered as the organism and the environment including it. That is, the evolving CEC-SLT is followed the evolution of host. The initial parameters for CEC-SLT model are determined as uniform random numbers from -1 to 1. The fitness function of CEC-SLT is the inverse form of the squared error between real value and predict value as following.

$$f_{host}(x) = \frac{C}{\sum_{i=1}^{F} \sum_{j=1}^{N_{out}} (o_{ij}(x) - t_{ij})^2} \quad (11)$$

In above equation, t is the value of known target variable and o is computed output value for prediction. C is a constant. The F and Nout are the numbers of patterns and

items. Next, the training of given data set is performed by evolving parasites. The evolution of training data is performed to retain larger training errors. So the fitness function for training data set is inverse of the fitness function of CEC-SLT model as the following.

$$f_{parasites}(x) = \sum_{i=1}^{D}\sum_{j=1}^{N_{out}}(o_{ij}(x) - t_{ij})^2 \quad (12)$$

The D and Nout are the numbers of patterns and items in the above equation. Our evolutionary approaches of CEC-SLT and training data set are competitive. In other words, proposed model is two different groups' competitive co-evolving. One is the parasites' evolution of given training data set. Another is the host's evolution of CEC-SLT. The following figure shows the process of proposed method.

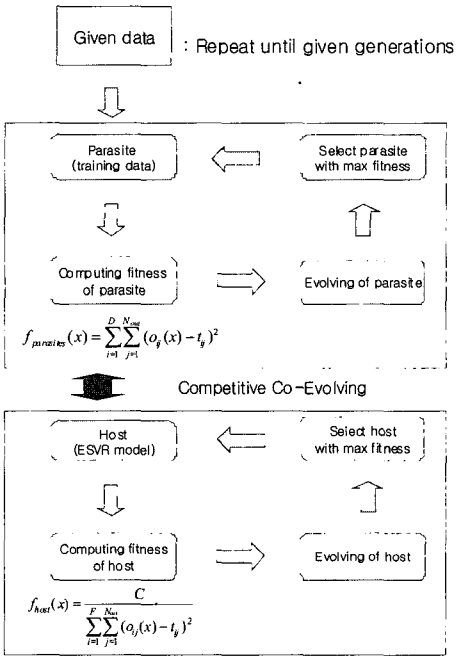

Figure 1. CEC-SLT method

In the CEC-SLT, the model and training data set are respectively evolved. During evolution for weight optimization of CEC-SLT, the competitive co-evolving is occurred between evolving model and evolving training data set. In this place, the co-evolutionary computation is used in CEC-SLT instead of Lagrange multipliers of traditional SLT for parameter optimization. The following is a pseudo-code of ESVR.

(CEC-SLT Algorithm)
Input:
1. Mutation: draw z_i from N(0,1)
2. Population size, p+1
3. Maximum number of generation, M

Output:
Optimal parameter string, s^*

BEGIN
Set t=0;
Initialize the population, $(w, C) \in R^{p+1}$;
w: weight vector of SVR
C:: regularization constant
Evaluate
1. ESVR model(host) by fitness, f_{host};
2. training data(parasites) by fitness, $f_{parasites}$;
if (host_evolving = True) then use fitness, f_{host}
else use fitness, $f_{parasites}$;
Repeat until (t<M)
{ Mutation: draw z_i from N(0,1);
$y_i^t = x_i^t + z_i$ for all $i \in \{1, 2, \cdots n\}$;
if $(f(x^t) \le f(y^t))$ then $x^{t+1} = x^t$
else $x^{t+1} = y^t$;
end if
Set t=t+1;
}
END

In the above algorithm, N(0,1) is the standard normal distribution with mean=0 and variance=1[32]. Therefore, in CEC-SLT method, the host and parasite are evolved respectively by competitive co-evolving approach.

4. Experimental Results

In our experiments, we show the performances of proposed model using Iris plants, Glass identification, and Abalone data sets from UCI machine learning repository and the web log data from KDD cup 2000[24,25]. The experiments have two experimental results. Firstly we apply CEC-SLT to classification models. Next, for prediction problems, the proposed model is used.

4.1 CEC-SLT for Classification

To make experiments for classification problems, we use Iris plants and Glass identification data sets from UCI machine learning repository. The Iris plants data have 4 input variables. The numbers of instances and classes are 150 and 3 respectively. Another data set, the Glass identification, has 214 instances. We use 9 input variables for classifying 7 classes in Glass identification data. To compare CEC-SLT with the established methods of classification, decision tree, GLM(generalized linear model), discriminant of statistics, and Bayesian classifier are used[6,14]. The following table shows the experimental results of Iris and Glass data.

Table 1. Misclassification rates of Iris and Glass data sets

Method	Iris	Glass
Decision Tree	2.58	1.95
GLM	1.25	2.68
Discriminant	1.95	3.11
Bayesian Classifier	1.66	2.59
CEC-SLT	0.43	0.87

We use the misclassification ratio as a performance measure for classification methods. In the above table, we see that the misclassification rate of CEC-SLT is the smallest among the comparative methods.

4.2 CEC-SLT for Prediction

To verify the performance of CEC-SLT for predictive problems, the Abalone and KDD 2000 data sets are used. In abalone data set, the numbers of instances and input variables are 4177 and 8 respectively. Using this data set, we make a data pre-processing experiment in data mining. Among data pre-processing approaches, our experiment is the missing value imputation. Abalone data set is complete without missing values. So, for the experiment, we make the complete abalone data to the incomplete with random missing patterns[10]. The incomplete abalone data set is consisted of four types which are 5%, 10%, 20%, and 30% missing ratios. We compare CEC-SLT with general SLT, non-linear regression, polynomial regression, and multiple linear regression[14,18]. The regression methods have been good missing value imputations[10]. Also we have known that the SLT was the good imputation tools according to the previous works[8,9]. One common heuristic is to withhold one-third of the available instances for the testing set, using the other two-thirds for training[11]. So, we make a experiment by this heuristic. In our experiment for prediction, for the evaluation measure of performance, the MSE(mean squared error) is used as following[1].

$$MSE=\frac{1}{n}\sum_{j=1}^{n}(y_j-y_j^*)^2 \qquad (13)$$

where y_j is the jth known value of target variable, y_j^* represents the jth predicted value of output, and n is the size of data set. The smaller the value of MSE is, the better the performance of model is. The experimental result is shown in the following table.

Table 2. MSE of Abalone data set

Method	MSE			
	5%	10%	20%	30%
CEC-SLT	0.31	0.38	0.42	0.49
SLT	0.51	0.59	0.63	0.69
Non-linear regression	0.68	0.71	0.81	0.97
Polynomial regression	0.87	0.96	1.26	1.63
Multiple linear regression	0.95	1.21	1.69	2.58

In above table, the non-linear regression model has the better performance than other regression models. And the SLT method had the smaller MSE value than three regression models. But we find CEC-SLT has the smallest MSE value among the competitive methods. So, we can verify the performance of proposed method using the Abalone data set.

Next, for another experiment, we use the KDD Cup 2000 data. The data set is web log file of real internet shopping mall(gazelle.com). The size of given data was 1.2GB. With similar to previous experiment, we use the one-third of given data for the testing and the other two-thirds for training. After data cleaning, we get the data as the following.

Table 3. Summary of KDD Cup data set

Attributes	Value range
cookie-id	13,109 (users)
assortment-id	269 (web pages)
duration-time	0~1000 (second(s))

In table 3, the cookie-id is the index of user accessing to web site, namely it is each user. The assortment-id represents each web page containing the descriptive contents of each item in the shopping mall and the duration-time of web page has the value between 0 and 1000 seconds. The data have originally many missing values. So, we replace missing values by predictive values using CEC-SLT. The experimental result is shown in the following table.

Table 4. MSE of KDD Cup data set

Method	MSE
CEC-SLT	4.13
SLT	5.98
Non-linear regression	7.48
Polynomial regression	8.14
Multiple linear regression	10.43

From above result, we also find the updated performance of CEC-SLT. Therefore, we can know CEC-SLT is to settle the local optima problems of SLT because its performance is better than SLT. Of course, CEC-SLT has better performance than the existing methods for classification and prediction.

5. Conclusion

To settle the local optima problems of SLT as a learning model, we combine competitive co-evolutionary computing into SLT which we called CEC-SLT. It do not use the Lagrange multipliers of SLT for optimizing parameters. As a substitute, using co-evolving, the parameters of CEC-SLT are optimized. According to the experimental results using data sets from UCI machine learning repository and KDD cup 2000, the performance of CEC-SLT is verified.

In the future works, we will apply the co-evolutionary computing to other learning methods for avoiding their local optima problems.

Reference

[1] G. Casella, R. L. Berger, *Statistical Inference*, Duxbury Press, 1990.

[2] V. Cherkassky, F. Mulier, *Learning From Data Concepts*, Theory, and Methods, John Wiley & Sons, 1998.

[3] C. Cortes, V. Vapnik, "Support Vector Networks," *Machine Learning*, vol. 20, pp. 273-297, 1995.

[4] A. E. Eiben, J. E., Smith, *Introduction to Evolutionary*

Computing, Springer, 2003.
[5] R. Fletcher, *Practical Methods of Optimization*, John Wiley & Sons, 1989.
[6] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, 2001.
[7] W. D. Hill, "Co-evolving parasites improve simulated evolution as an optimization procedure," *Physica*, vol. 42, pp. 228-234, 1992.
[8] S. H. Jun, M. Gen, "Evolutionary Computation for Statistical Learning Theory", *Proceedings of Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems*, 2004.
[9] S. H. Jun, "Web Usage Mining Using Support Vector Machine", *Lecture Note in Computer Science*, vol. 3512, pp. 349~356, Springer, 2005.
[10] R. J. A. Little, D. B. Rubin, *Statistical Analysis with Missing Data*, John Wiley & Sons, 2002.
[11] T. M. Mitchell, *Machine Learning*, WCB McGraw Hill, 1997.
[12] T. M. Mitchell, *An introduction to Genetic Algorithms*, MIT Press, 1998.
[13] D. J. Montana, L. Davis, "Training Feedforward Neural Network Using Genetic Algorithm," *Proceedings of the International Conference on Artificial Intelligence*, pp. 763-767, 1989.
[14] R. H. Myers, *Classical and Modern Regression with applications*, Duxbury, 1990.
[15] A. S. Pandya, R. B. Macy, *Pattern Recognition with Neural Networks in C++*, CRC Press, 1995.
[16] M. J. Park, *Competitive Co-Evolving Neural Network: Host and Parasites*, Master's thesis, Sogang University, 2002.
[17] S. H. Ross. *Introductory Statistics*, McGraw-Hill, 1996.
[18] A. J. Smola, *Regression estimation with support vector learning machines*, Master's thesis, Technische University, 1996.
[19] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.
[20] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.
[21] V. N. Vapnik, S. Golowich, A. Smola, "Support vector method for function approximation, regres-sion estimation, and signal processing," *Advanced in Neural Information Processing Systems*, Cambridge, MA, 1997.
[22] X, Yao, "A review of evolutionary artificial neural networks," *International Journal Intelligent System*, vol. 8, no. 4, pp. 539-567, 1993.
[23] X. Yao, "Evolving Artificial Neural Networks," *Proceeding of the IEEE*, vol. 87, issue 9, pp. 1423-1447, 1999.
[24] KDD Cup 2000, www.ecn.purdue.edu /KDDCUP
[25] UCI M. L. Repository, www.ics.uci.edu/~mlearn

Sung-Hae Jun
He received the B.S., M.S., and Ph.D. degrees in department of Statistics from Inha University, Korea, in 1993, 1996, and 2001. He is currently an Assistant Professor in the department of Statistics, Cheongju University, Korea and Ph. D candidate of Computer Science of Sogang University, Korea. His research interests include machine learning and evolutionary computing.
Phone : +82-43-229-8205
Fax : +82-43-229-8432
E-mail : shjun@cju.ac.kr