Advances in Technology Innovation, vol. 1, no. 2, 2016, pp. 53 - 57

Improved SVM Classifier Incorporating Adaptive Condensed Instances Based on Hybrid Continuous-Discrete Particle Swarm Optimization

Chun-Liang Lu[1,*], Tsun-Chen Lin[2]

[1]Department of Applied Information and Multimedia, Ching Kuo Institute of Management and Health, Keelung, Taiwan.

[2]Department of Computer and Communication Engineering, Dahan Institute of Technology, Hualien, Taiwan.

Received 20 Sep 2016; received in revised form 24 Sep 2016; accepted 28 Sep 2016

Abstract

In recent years, support vector machine (SVM) based on empirical risk minimization is supervised learning model which has been successfully used in the classification and regression. The standard soft-margin SVM trains a classifier by solving an optimization problem to decide which instances of the training data set are support vectors. However, in many real applications, it is imperative to perform feature selection to detect which features are actually relevant. In order to further improve the performance, we propose the adaptive condensed instances (ACI) strategy based on the hybrid particle swarm optimization (HPSO) algorithm for the SVM classifier design. The basic idea of the proposed method is to adopt HPSO to simultaneously optimize the ACI and SVM kernel parameters for the classification accuracy enhancement. The numerical experiments on several UCI benchmark datasets are conducted to find the optimal parameters for building the SVM model. Experiment results show that the proposed framework can achieve better performance than other published methods in literature and provide a simple but subtle strategy to effectively improve the classification accuracy for SVM classifier.

Keywords: hybrid particle swarm optimization (HPSO), adaptive condensed instances (ACI), support vector machine (SVM)

1. Introduction

Support vector machine (SVM) was originally proposed by Vapnik [1] which is a powerful classification method with state-of-the-art performance in machine learning theory, has drawn considerable attentions due to its high generalization ability for a wide range of applications including speaker recognition [2], bioinformatics [3] and text categorization [4]. In many pattern classification tasks, we are confronted with the problem that the input space is high dimensional and to find out the combination of original input features which contribute most to the classification is crucial. The computational cost of classification grows heavily with data dimension size, making feature selection an important issue for the SVM. The feature selection mechanism falls into three categories: filtering, wrapper, and embedded methods [5]. Filters generally involve a non-iterative computation on the original features, which can execute very fast, but not usually optimal since the learning algorithm are not taken into account. Wrapper methods usually achieve better results than filters since they are tuned to the specific interactions between the classifier with original feature set and very computationally intensive. Finally, unlike filters and wrappers, the embedded techniques simultaneously determine features and classifier during the training process but the computational time is smaller than wrapper methods.

Feature selection problem is a challenging task because there can be complex interaction among features. Therefore, an exhaustive search is practically impossible, and the efficient global search technique is needed. Evolutionary computation (EC) are well known heuristic approaches global search ability such as simulated annealing (SA) [6], genetic algorithm (GA) [7], and particle swarm optimization (PSO) [8], have gained a lot of attention from researchers in the area. Compared with other EC algorithms such as SA and GA, PSO is computationally less expensive and can converge more quickly. A GA-based feature selection method, which optimized both the feature selection and parameters for SVM, was proposed by Huang [9], and the authors pointed out that the algorithm may work superior to the conventional grid search method. However, the treatment of these redundant or irrelevant instances is not taken into account in the classification procedure. In this paper, the effectively adaptive condensed instances (ACI) strategy that we previously published [7] is applied to decide which instances of the training data set are support vectors for coping with the problem mentioned above. Short communications of the early stages of this work have appeared in [7]. Here we significantly extend our approach to account for the relevant instances selection during the SVM training process.

In order to further improve the classification performance, the ACI strategy based on the hybrid particle swarm optimization (HPSO) algorithm is proposed for the SVM classifier design. Several UCI benchmark datasets are conducted to validate the effectiveness and the experiment results show that the proposed framework can achieve better performance than other GA-based existing methods in literature. The remainder of this paper is organized as follows. Section 2 describes the related work including the basic PSO and SVM classifier. Section 3 illustrates particle representation, hybrid PSO with disturbance operation, ACI scheme and the proposed framework for the SVM classifier. Section 4 provides the experiment results, and conclusions are made in section 5.

2. Related work

2.1. Particle swarm optimization (PSO) algorithm

The PSO algorithm, which is originally developed by Kennedy and Eberhart [10], is a search algorithm modeling the social behavior of birds within a flock. In the PSO algorithm, individuals referred to as particles, are flown through hyper dimensional search space. PSO is easy to implement, few parameters to adjust, and usually faster convergence rates

*Corresponding author, Email: leucl@ems.cku.edu.tw

Copyright © TAETI

than other evolutionary algorithms. During the optimization procedure, particles communicate good positions to each other and adjust position according to their history experience and the neighboring particles. The basic concept of the PSO algorithm is illustrated as follow:

$$v_{id}^{k+1} = w \cdot v_{id}^{k} + c_1 \cdot r_1 \cdot (pb_{id}^{k} - x_{id}^{k}) + c_2 \cdot r_2 \cdot (gb_{id}^{k} - x_{id}^{k}) \tag{1}$$

$$x_{id}^{k+1} = v_{id}^{k+1} + x_{id}^{k} \tag{2}$$

where $d = 1,2,...,D$, $i = 1,2,..., N$, and D is the dimension of the search space, N is the population size, k is the iterative times; v_{id}^{k} is the i-th particle velocity, x_{id}^{k} is the current particle solution, the i-th particle position is updated by equation (2). pb_{id}^{k} is the i-th particle best (p_{best}) solution achieved so far; gb_{id}^{k} is the global best (g_{best}) solution obtained by any particle in the population; r_1 and r_2 are random values in the range [0,1] for denoting remembrance ability. Both of c_1 and c_2 are learning factors, w is inertia factor. A large inertia weight facilitates global exploration, while a small one tends to local exploration. Generally, the value of each component in V can be clamped to the range $[-V_{\max}, V_{\max}]$ for controlling excessive roaming of the particle outside the search space. The PSO procedure is organized in the following sequence of steps.

Step 1: **Initialize**: Randomly generating initial particles.

Step 2: **Fitness evaluation**: Calculate the fitness values of each particle in the population.

Step 3: **Update**: Compare fitness values of each particle to update the velocity and position by using equation (1) and (2).

Step 4: **Termination criterion**: Repeat the Step 2 to Step 3 until the number of iteration reaches the pre-defined maximum number or a termination criterion is satisfied, and the best solution g_{best} is displayed.

2.2. Support vector machine (SVM) classifier

Support vector machine (SVM) have drawn much attention due to their good performance and solid theoretical foundations [11]. The main concepts of SVM are to first transform input data into a higher dimensional space by means of a kernel function and then to find an optimal separating hyper-plane between the two data sets. A practical difficulty of using SVM is the selection of parameters such as the penalty parameter C of the error term and the kernel parameter γ in RBF kernel function. The appropriate choice of parameters is to get the better generalization performance. The description of SVM is as follows. Given a set of training data $(x_1, \cdots, x_p)$ with corresponding class labels $(y_1, \cdots, y_p)$ and $y_i \in \{+1, -1\}$. The SVM attempts to find a decision surface $f(x)$, to jointly maximize the margin between the two classes and minimize the classification error on the training set.

$$f(x) = \sum_{i=1}^{p} \alpha_i y_i K(x, x_i) + b \tag{3}$$

where K is a kernel function, α_i is the Lagrange multiplier corresponding to the i-th training data x_i and b is the bias. The simple kernel is the inner product function $K(x, \hat{x}) = \langle x, \hat{x} \rangle$ which produces the linear decision boundaries. Nonlinear kernel function maps data points to a high-dimensional feature space as linear decision spaces. Two commonly used kernels are the polynomial kernel $K_p(x, \hat{x})$ and the radial basis function (RBF) kernel $K_r(x, \hat{x})$. The integer polynomial order θ in K_p and the width factor δ in K_r are hyper-parameters which are tuned to a specific classification problem.

$$K_p(x, \hat{x}) = \left(1 + \langle x, \hat{x} \rangle\right)^{\theta} \tag{4}$$

$$K_r(x, \hat{x}) = e^{-\delta \| x - \hat{x} \|^2} \tag{5}$$

3. Method

Since the optimal hyper-plane obtained by the SVM depends on only a small part of the data points (support vectors), it may become sensitive to noises or outliers in the training set. In this section, the ACI scheme based on hybrid PSO (HPSO) is proposed to tackle feature selection, condensed instances extraction and parameters setting simultaneously for SVM. The particle representation, fitness definition, disturbance strategy for PSO operation, ACI scheme and the proposed hybrid framework for SVM are described as follows.

3.1. Particle representation

This study used the RBF kernel function for the SVM classifier to implement our proposed method. The RBF kernel function requires two parameters C and γ should be set. Using the adaptively condensed instance rate for ACI scheme and the RBF kernel for SVM, these three parameters θ_{cond} , C , γ and features used as input attributes must be optimized simultaneously for our proposed hybrid system. The particle, therefore, is comprised of four parts, θ_{cond} , C , γ are the continuous variables and the features mask are the discrete variables. Table 1 shows the particle representation of our design.

Table 1 The hybrid particle representation

	Discrete-variables					Continuous -variables		
	Input features mask					θ_{cond}	C	γ
Particle representation	$x_{i,1}$	$x_{i,2}$	…	$x_{i,d}$	… x_{i,n_f}	x_{i,n_f+1}	x_{i,n_f+2}	x_{i,n_f+3}

As shown in Table 1, the representation of particle i with dimension of $n_f + 3$, where n_f is the number of features that varies from different datasets, $x_{i,1} \sim x_{i,n_f}$ are the features mask, x_{i,n_f+1} indicates the parameter value θ_{cond} , x_{i,n_f+2} represents the parameter value C and x_{i,n_f+3} denotes the parameter value γ .

Fitness function F is the guide of HPSO operation to maximize the classification accuracy and minimize the number of selected features. A_{cc} is the SVM classification accuracy, f_i denotes the feature mask which 1 represents the feature i is selected and 0 indicates that feature i is not selected. Thus, the particles with high classification accuracy and a small number of features produce a high fitness value and affect particle's positions on the next iteration. Considering the tradeoff between the classification accuracy and selected feature number, these two weight w_1 and w_2 values can be adjusted according to the preference for SVM classifier design.

Copyright © TAETI

$$F = w_1 \times Acc + w_2 \times \left(\sum_{i=1}^{n_f} f_i \right)^{-1} \quad (6)$$

3.2. PSO with disturbance operation

In the discrete PSO, the particle's personal best and global best is updated as in continuous value. The major different between discrete PSO with continuous version is that velocities of the particles are rather defined in terms of probabilities that a bit whether change to one. By this definition, a velocity must be restricted within the range $[V_{\min}, V_{\max}]$. This can be accomplished by a sigmoid function $S(v)$, and the new particle position is calculated using the following rule:

If $v_{id}^{k+1} \notin (V_{\min}, V_{\max})$ then $v_{id}^{k+1} = \max(\min(V_{\max}, v_{id}^{k+1}), V_{\min})$,

where

$$S(v_{id}^{k+1}) = \frac{1}{1+e^{-v_{id}^{k+1}}} \quad (7)$$

If $rand() < S(v_{id}^{k+1})$, then $x_{id}^{k+1} = 1$; else $x_{id}^{k+1} = 0$.

The function $S(v_{id})$ is a sigmoid limiting transformation and $rand()$ is a random number selected from a uniform distribution in [0, 1]. Note that the discrete PSO is susceptible to a sigmoid function $S(v)$ saturation which occurs when velocity values are either too large or too small. For a velocity of zero, it is a probability of 50% for the bit to flip.

According to the searching behavior of PSO, the g_{best} value will be an important clue in leading particles to the global optimal solution. It is unavoidable for the solution to fall into the local minimum while particles try to find better solutions. In order to allow the solution exploration in the area to produce more potential solutions, a mutation-like disturbance operation is inserted between Eq. (1) and Eq. (2). The disturbance operation random selects k dimensions ($1 \le k \le$ problem dimensions) of m particles ($1 \le m \le$ particle numbers) to put Gaussian noise into their moving vectors (velocities). The disturbance operation will affect particles moving toward to unexpected direction in selected dimensions but not previous experience. It will lead particle jump out from local search and further can explore more diversity of searching space.

3.3. Adaptive condensed instances (ACI) scheme

The effectively ACI scheme that we previously published in [7] is extended to decide which instances of the training data set are support vectors. The adaptively condensed instances coefficient in the data reduced process is flexible to edit out noisy samples, reduce the superfluous data points and make the SVM less sensitive to noises and outliners. The adaptively condensed instances coefficient $\theta_{cond} \in [0,1]$ is defined as the ratio of the selected number of condensed instances to overall dataset. The ACI scheme is described as follows.

Step 1: **Initialize**: Randomly generating initial instance set.

Step 2: **Condense**: To decide whether all samples have been achieved the user defined threshold θ_{cond}. If so, terminate the process; otherwise, go to Step 3.

Step 3: **Extend**: If there are any un-condensed instances, using the nearest neighbor voting to renew the condensed instances; otherwise, no more new data is joined and go to Step 2.

3.4. The proposed framework for SVM classifier

Based on the particle representation, fitness definition, PSO with disturbance operation and ACI scheme mentioned above, details of the proposed hybrid framework for SVM procedure from step 1 to step 9 are described as follows.

Step 1: Data preparation

Given a dataset D is considered using the 10 fold cross validation to split the data into 10 groups. Each group contains training and testing sets. The training and testing sets are represented as D_{train} and D_{test}, respectively.

Step 2: Hybrid PSO initialization and parameters setting

Set the PSO parameters including the number of iterations, number of particles, velocity, particle dimension, disturbance rate, and weight for fitness. Generate initial hybrid particles comprised of the features mask, θ_{cond}, C and γ.

Step 3: Condensed instances selection via the ACI scheme

According to the condensed instances coefficient θ_{cond} represented in the particle and calculated from Step 2, when all the condensed instances are computed, the nearest neighbor voting is used to renew the condensed instances set.

Step 4: Feature scaling

Feature scaling is to properly reveal the interactions between features and to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Normalization by Eq. (8) can be linearly scaled to range [-1, +1] or [0, 1], where $A_j(x_i)$ is the original attribute value of feature x_i, $A_j^{'}(x_i)$ is scaled value, $\max_j$ and $\min_j$ correspond to the maximum and minimum values for A_j over all samples.

$$A_j^{'}(x_i) = \frac{A_j(x_i) - \min_j}{\max_j - \min_j}, \quad \forall i \quad (8)$$

Step 5: Feature selection

According to the feature mask, which is represented in the particle from Step 2, is to select input features for training set D_{train} and testing set D_{test}. The selected features subset can be denoted as D_{f_train} and D_{f_test}, respectively.

Step 6: To train and test for SVM classifier

For the parameters θ_{cond}, C and γ which are represented in the particle, to train the SVM classifier on the training dataset D_{f_train}, then the classification accuracy A_{cc} for SVM on the testing dataset D_{f_test} can be evaluated.

Step 7: Fitness evaluation

For each particle, the fitness value is to be calculated by the Eq. (6). The optimal fitness value can be stored on the evolution process of PSO to search for the better fitness of particle in the next particles evolution procedure.

Step 8: Termination criteria

When the number of iteration reaches the pre-defined maximum number or a termination criterion is satisfied, the best solution g_{best} is obtained, and the program ends; otherwise, go to the next step.

Copyright © TAETI

Step 9: Hybrid PSO operation

In the evolution process, discrete-valued and continuous-valued dimension of HPSO with the disturbance operator continued to be applied for searching better particle solutions.

4. Experiment results

4.1. Dataset and system description

To measure the performance of the developed hybrid framework, several real benchmark datasets in Table 2 are conducted to verify the effectiveness of performance. These

Table 2 The UCI benchmark datasets [7]

Names	Classes	Instances	Nominal Attributes	Numeric Attributes	Total Attributes
Australian	2	690	6	8	14
German	2	1000	0	24	24
Heart	2	270	7	6	13
Iris	3	150	0	4	4
Vehicle	4	940	0	18	18
Vowel	11	990	3	10	13

datasets are partitioned using the 10-fold cross validation. Our implementation platform was implemented on Matlab 2013, by extending the LIBSVM which is originally designed by Chang and Lin [12]. Through initial experiment, the parameter values of the PSO were set as follows. The swarm size is set to 100 particles. The searching ranges of continuous type dimension parameters are: $\theta_{cond} \in [0,1]$, $C \in [10^{-2},10^{4}]$ and $\gamma \in [10^{-4},10^{4}]$. The discrete type particle for features mask, we set $[V_{\min},V_{\max}]=[-6,6]$, which yields a range of $[0.9975,0.0025]$ using the sigmoid limiting transformation by Eq. (7). Both the cognition learning factor c_1 and the social learning factor c_2 are set to 2. The disturbance rate is 0.05, and the number of generation is 500. The inertia weight factor $w_{\min}=0.4$ and $w_{\max}=0.9$. The linearly decreasing inertia weight is set as Eq. (9), where i_{now} is the current iteration and $i_{\max}$ is the pre-defined maximum iteration. According to the fitness function defined by Eq. (6), set the accuracy's weight $w_1=0.8$ and the feature's weight $w_2=0.2$.

$$w = w_{\max} - \frac{i_{now}}{i_{\max}}(w_{\max} - w_{\min}) \tag{9}$$

Table 3 The parameters setting for PSO and GA

PSO parameters	GA parameters
Swarm size: 600	Population size: 600
Generation number: 500	Generation number: 500
Learning factor: $(c_1,c_2)=(2,2)$	Crossover rate: 0.7
Inertia weight factor: $(w_{\min},w_{\max})=(0.4,0.9)$	Mutation rate: 0.05
Disturbance rate: 0.05	

In [9], the authors presented the GA-based method without ACI mechanism for searching the best C , γ , and features subset. The existing GA-SVM method without ACI scheme deals solely with feature selection and parameters optimization by means of genetic algorithm, and the treatment of these redundant or noisy instances in a classification process did not be taken into account. Our proposed hybrid framework has been tested fairly extensively and compared with these approaches including both GA-SVM and PSO-SVM approaches without ACI mechanism. Furthermore, the comparison of PSO and GA technique with the ACI scheme for SVM is also presented. While applying GA algorithm, a number of parameters are required to be specified. The two algorithms (both PSO and GA) are run for the same number of fitness function evaluations. Table 3 summarized the parameters used for PSO and GA technique. The empirical results are reported in section 4.2.

4.2. Result and comparison

The results obtained by the developed HPSO-ACI-SVM approach are compared with those of GA-SVM proposed by Huang et al. [9] without ACI scheme. Taking the Heart disease dataset, for example, the classification accuracy A_{cc} , number of selected features n_f , and the best parameters θ_{cond} , C , γ for each fold are shown in Table 4. For the HPSO-ACI-SVM method, average classification accuracy rate is 96.87%, and average number of features is 4.9. For the GA-SVM without ACI approach, its average classification accuracy rate is only 94.81%, and average number of features is 5.6.

Table 4 Comparison for HPSO-ACI-SVM and GA-SVM

Fold#	HPSO-ACI-SVM algorithm					GA-SVM algorithm			
	Acc	n_f	Optimized θ_{cond}	Optimized *C*	Optimized γ	*Acc*	n_f	Optimized *C*	Optimized γ
1	94.7	4	0.92504916	36.9081772	0.42653017	92.4	5	5.5801297	0.57708836
2	98.4	6	0.95114065	63.9130285	0.26394783	98.1	7	60.2411283	0.01708658
3	100.0	7	0.96018735	74.8052413	0.10487605	100.0	7	171.3113128	0.21574369
4	95.8	4	0.95350726	103.6081825	0.03795136	92.6	4	95.2776294	0.05706643
5	98.5	6	0.93890508	57.5232067	0.17608865	98.4	7	267.4913877	0.33562374
6	95.4	4	0.94817207	45.4091347	0.47518046	92.6	6	53.6696245	0.07895168
7	96.3	4	0.97017842	96.5037885	0.08953627	92.1	4	216.8613768	0.12075873
8	97.6	5	0.96178237	28.6501678	0.17805265	97.5	6	170.5518422	0.45896385
9	94.6	4	0.93660675	147.9616028	0.01687434	90.7	4	75.5779491	0.08705219
10	97.4	5	0.92738012	54.9401383	0.30184766	93.7	6	83.6912762	0.11775911
Average	**96.87**	**4.9**	**-**	**-**	**-**	**94.81**	**5.6**		

Table 5 Comparison for HPSO-ACI-SVM, PSO-SVM and GA-SVM

Name	(1)HPSO-ACI-SVM	(2)PSO-SVM	(3)GA-SVM	(1) vs. (2) p-value	(1) vs. (3) p-value
Australian	90.92±1.27	89.3±1.54	88.17±1.81	0.059	0.005*
German	88.26±1.42	86.17±2.54	85.64±1.80	0.037*	0.013*
Heart disease	96.87±1.79	95.13±3.23	94.81±3.32	0.028*	0.008*
Iris	100±0	100±0	100±0	1.0	1.0
Vehicle	87.25±2.28	84.96±2.15	84.06±2.40	0.005*	0.005*
Vowel	99.60±0.42	99.02±0.47	98.52±0.89	0.005*	0.005*

Copyright © TAETI

Table 5 shows the summary results for the average classification accuracy rate of the HPSO-ACI-SVM hybrid framework and PSO-SVM, GA-SVM without ACI method on six UCI datasets. In Table 5, the classification accuracy rate is represented as the form of 'average ± standard deviation'. To highlight the advantage, we used the non-parametric Wilcoxon signed rank test for all of the datasets. In Table 5, the p-values of HPSO-ACI-SVM versus PSO-SVM and GA-SVM are smaller than the statistical significance level of 0.05 except Iris dataset. That is to say, the developed HPSO-ACI-SVM yields higher classification accuracy rate across different datasets to enhance the performance for the SVM.

Further, under our proposed hybrid framework with ACI mechanism, the performance of both PSO and GA optimization techniques in terms of the average classification accuracy rate Avg_A_{cc} and the average number of selected features Avg_n_f is compared. In Table 6, the HPSO exhibits slightly higher classification accuracy and fewer selected features than GA. It is observed that, from an evolutionary point of view, the performance of the HPSO is better than GA. However, the results indicate that both PSO and GA algorithms can be used in optimizing the parameters under our developed hybrid framework to effectively improve the classification accuracy for SVM classifier design.

Table 6 Comparison for HPSO-ACI-SVM and GA-ACI-SVM

Name	HPSO-ACI-SVM		GA-ACI-SVM		Wilcoxon
	Avg_A_{cc}	Avg_n_f	Avg_A_{cc}	Avg_n_f	p-value
Australian	90.92±1.27	4±0.94	90.25±1.42	4.3±0.82	0.012*
German	88.26±1.42	12.4±0.97	87.38±0.97	12.6±0.84	0.008*
Heart disease	96.87±1.79	4.9±1.10	96.1±2.32	5.0±1.15	0.018*
Iris	100±0	1±0	100±0	1±0	1.0
Vehicle	87.25±2.28	8.3±1.16	85.92±1.85	8.6±0.97	0.012*
Vowel	99.60±0.42	7±0.82	99.21±0.75	7.1±0.74	0.027*

5. Conclusion

For SVM classifier, it is imperative to perform feature selection to detect which features are actually relevant. In this study, the effectively adaptive condensed instances (ACI) strategy is applied to decide which instances of the training data set are support vectors. Furthermore, we propose the ACI scheme based on the hybrid particle swarm optimization (HPSO) algorithm to simultaneously optimize the condensed instances and SVM kernel parameters for the classification accuracy enhancement. Several UCI benchmark datasets are conducted to validate the effectiveness of the proposed method, and the experiment results show that the proposed hybrid framework can achieve better performance than other existing methods in literature. Investigating more large scale dataset as well as combining other heuristic algorithm for hybrid system may be interesting future work.

References

[1] V. N. Vapnik, "An overview of statistical learning theory," IEEE Transactions on Neural Networks, vol. 10, pp. 988-999, 1999.

[2] I. J. Ding, C. T. Yen, and D. C. Ou "A method to integrate GMM, SVM and DTW for speaker recognition," International Journal of Engineering and Technology Innovation, vol. 4, pp. 38-47, 2014.

[3] R. Kothandan and S. Biswas, "Identifying microRNAs involved in cancer pathway using support vector machines," Computational Biology and Chemistry, vol. 55, pp. 31-36, Apr. 2015.

[4] B. Ramesh and J. G. R. Sathiaseelan, "An advanced multi class instance selection based support vector machine for text classification," Procedia Computer Science, vol. 57, pp. 1124-1130, 2015.

[5] X. Zhang, G. Wu, Z. Dong and C. Crawford, "Embedded feature-selection support vector machine for driving pattern recognition," Journal of the Franklin Institute, vol. 352, pp. 669-685, 2015.

[6] S.W. Lin, Z. J. Lee, S. C. Chen and T. Y. Tseng, "Parameter determination of support vector machine and feature selection using simulated annealing approach," Applied Soft Computing, vol. 8, pp. 1505-1512, 2008.

[7] C. L. Lu, I. F. Chung and T. C. Lin, "The hybrid dynamic prototype construction and parameter optimization with genetic algorithm for support vector machine," International Journal of Engineering and Technology Innovation, vol. 5, pp. 220-232, 2015.

[8] B. Sahu and D. Mishra, "A novel feature selection algorithm using particle swarm optimization for cancer microarray data," Procedia Engineering, vol. 38, pp. 27-31, 2012.

[9] C. L. Huang and C. J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," Expert systems with applications, vol. 31, pp. 231-240, 2006

[10] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," In Proceedings of the World on Systematics, Cybernetics and Informatics, vol. 8, pp. 4104-4109, 1997.

[11] V. Vapnik, The nature of statistical learning theory, New York: Springer-Verlag, 1995.

[12] C. C. Chang, and C. J. Lin, "Training nu-support vector regression: theory and algorithms," Neural Computation, vol. 14, pp. 1959-1977, 2002.

Copyright © TAETI