Parameter Optimization with Restarting Genetic Algorithm for the Forest Type Classification

Keerachart Suksut, Nuntawut Kaoungku, Nittaya Kerdprasop, and Kittisak Kerdprasop

***Abstract*—Data mining is the process to find the knowledge from the huge amount of stored information and use the discovered knowledge to predict or classify the new data item that its class label is unknown. Among many available algorithms to do data classification, support vector machine is one of the most accurate mining methods. Support vector machine is a parametric approach such that proper setting of parameter value can directly influence the classifying performance of the machine. Currently, genetic algorithm can find the best parameter for support vector machine. The genetic algorithm is the search algorithm for optimal answer with adaptive heuristic search based on the evolutionary characteristic of nature. But the problem of genetic algorithm is that sometime the algorithm cannot find the best parameter because the improper setting of a random initial value. In this research, we propose the new technique to improve performance of genetic algorithm to find the best parameter with restarting concept. We show the performance of the proposed technique with application for image-based forest type classification over the forest area in Japan with the satellite image data from the ASTER satellite. The results show that the proposed technique can classify the forest type more accurate than other existing techniques.**

***Index Terms*—Data classification, restarting genetic algorithm, support vector machine, forest type classification.**

I. Introduction

Data mining is to find the valuable knowledge from the stored information and database. The knowledge means a pattern or relationship that is hidden in the data [1]. Currently, we can use mathematical method, statistics or other computational methods for knowledge extraction. There are many types of data mining tasks such as data classification, association rule mining, clustering, forecasting, and other analysis tasks. Data classification is the majority of data mining task being applied in many real application areas and it is also the focus of our work.

The popular techniques in the data classification include artificial neural network (ANN), decision tree, naïve Bayes, support vector machine (SVM), and many more. There is no single technique that is good for any application. However, SVM is recently the most applicable technique because of its overall high performance on data classification [2], [3]. The main concept of SVM is trying to create the hyperplane for separating object or data with high distance between each groups of data.

To further improve the SVM performance for data classification, some techniques to adjust learning algorithm with optimal parameters have been proposed. For the parameter optimization purpose, most researchers turn to the genetic algorithm as an efficient tool to learn the suitable parameter values [4]-[6]. But the main problem of applying genetic algorithm is in process of assigning the initial population. Sometimes we cannot find the best parameters because the random initial values of the initial population process do not cover the set of the best parameters.

In this research, we propose the technique for restarting the random initial population when the new generation is less powerful than the old population. This concept is called "Restarting Genetic Algorithm" to be used for parameter optimization. The parameters to be further used by SVM learning algorithm include C, epsilon, and gamma parameters.

II. Background Theories

A. Genetic Algorithm

Genetic algorithm is a computational method to find the solution with adaptive heuristic search based on the evolutionary characteristic of nature such that the one who is stronger has more chance to survive than those who are weaker and the stronger one can inherit strength to their children [7]-[9]. The simple genetic algorithm is shown in Fig. 1.

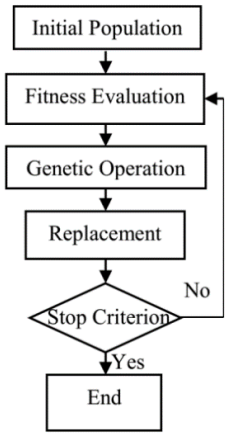

Fig. 1. General genetic algorithm process.

From Fig. 1, the genetic algorithm is composed of 5 main steps. Step 1 is the assigning for initial population; based on random selection. Step 2 is fitness evaluation; used for evaluating the fitness of each population. Step 3 is the genetic operation that can be either randomly selecting the

Manuscript received September 19, 2017; revised December 1, 2017. This work was supported by grants from Suranaree University of Technology through the funding of Data Engineering Research Unit and the Knowledge Engineering Research Unit.

The authors are with the School of Computer Engineering, Suranaree University of Technology (SUT), 111 University Avenue, Muang, Nakhon Ratchasima 30000, Thailand (corresponding author: K. Suksut, Ph: +66879619062; e-mail: mikaiterng@gmail.com, nuntawut@sut.ac.th, nittaya@sut.ac.th, kerdpras@sut.ac.th).

population, crossing over two parent chromosomes to create better offspring, or mutating a chromosome at randomly selected point. Step 4 is replacing individual in the population; the replacement of the old or parent population with the new generation. Step 5 is checking for stopping criterion, such as stop the process when it has created the new generation over 10 times.

B. Support Vector Machine

Support Vector Machine (SVM) [10] is a machine learning algorithm for classifying objects of different classes. The idea of SVM for accurately classifying objects with mixed classes is that transforming data to lie on a hyperplane. The hyperplane should be an optimal one in the sense that mixed objects can be clearly separated into different classes.

The optimal hyperplane is the line or plane that has maximum margin between the data point from each different classes. The hyperplane will split the data such that objects with the same class label form themselves as a single group, whereas objects with different class labels should be in a different group.

Fig. 2 shows an optimal hyperplane represented as a dashed line. The two classes in the figure are class 1 (represented as a small square) and class -1 (represented as a circle). To use the hyperplane as a model to classify object, the formula given in equation (1) is to be deployed.

$$\begin{aligned} w^T x + b \ &\geq 1, \text{when } y_i = +1 \\ w^T x + b \ &\leq 1, \text{when} y_i = -1 \end{aligned} \tag{1}$$

where w is weight vector, and b is bias.

The weight vector is used for determining the direction and inclination of the hyperplane and bias is used for determining the distance between the hyperplane and the origin. Consider two dimensional data $X = (x_1, x_2)^T$, we can compute the linear hyperplane with equation (2).

$$h(x) = w^T x + b = w_1 x_1 + w_2 x_2 + b = 0 \tag{2}$$

Given two data points on hyperplane $A = (A_1, A_2)$ and $B = (B_1, B_2)$, the equation (3) used for computing the weight vector.

$$\text{weight vector} = -\frac{w_1}{w_2} = -\frac{(B_2 - A_2)}{(B_1 - A_1)} \tag{3}$$

The margin can be computed with equation (4) and the size of weight vector is computed as in (5).

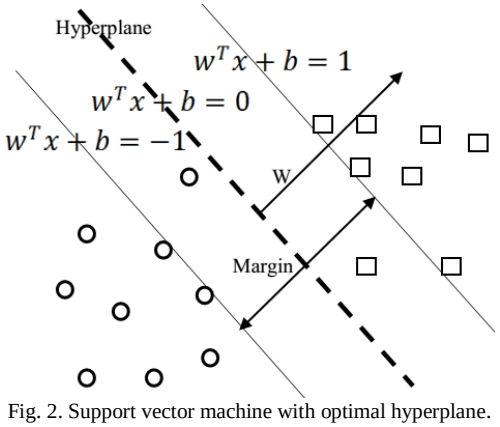

Fig. 2. Support vector machine with optimal hyperplane.

$$margin = \frac{2}{||w||} \tag{4}$$

$$||w|| = \sqrt{w_1^2 + w_2^2} \tag{5}$$

To apply support vector machine for the classification task, users have to set three important parameters (C, epsilon, and gamma). Parameter C is used to control the influence of each individual support vector (i.e., the data points on the borderlines which are up and below the optimal hyperplane. Setting the C parameter involves trading error penalty for stability.

Parameter epsilon is for controlling the width of the epsilon-insensitive zone. The value of epsilon can affect the number of support vectors that are used to find the optimal hyperplane. Parameter gamma is the kernel parameter of the Gaussian radial basis function. The small gamma implies that the learned model will have the large margin. The large gamma means that learned model will have small margin (may cause overfitting).

C. Classification Performance Evaluation

To evaluate performance of classification model, we use the accuracy metric for assessing the performance for each classification technique. The computation of this metric is based on the values in confusion matrix [11] as shown in Table I.

Accuracy is a measure for overall performance of the classification model, and the computation is shown as in equation (6).

$$\text{ycaruccA} = \frac{(\text{TP} + \text{TN})}{(TP + FN + FP + TN)} \tag{6}$$

where:

TP is the number of actual data from positive class and the model can correctly predict that data to be in a positive class.

TABLE I: CONFUSION MATRIX FOR TWO CLASS CLASSIFICATION

		Actual Data	
		Positive	Negative
Predicted Data	Positive	TP	FP
	Negative	FN	TN

TN is the number of actual data from negative class and the model can correctly predict that data to be in a negative class.

FP is the number of actual data from negative class but the model incorrectly predicts that data to be in a positive class.

FN is the number of actual data from positive class but the model predicts that the data incorrectly as in a negative class.

III. MATERIALS AND METHODS

In this research, we designed the process of parameter optimization with restarting genetic algorithm for the forest type classification as shown in Fig. 3. The learned parameters are to be used by the SVM algorithm on classifying different types of forest.

From Fig. 3, we can describe our proposed framework as follows. We find the optimal parameter for the classification process by introducing restarting genetic algorithm. To get the initial population, we random initial population until obtaining the specified population size. The fitness value of

each population is evaluated based on the accuracy from classifying the training data set with support vector machine by using the parameters from each population. After that, we select elite population, which are the top *k* population with the highest fitness values. Then apply the genetic operation to obtain new generation.

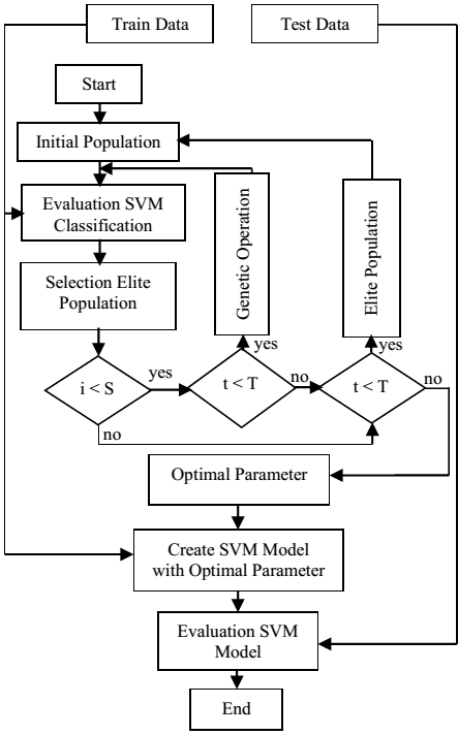

Fig. 3. The classification research framework.

If the new generation is less powerful than the old population, our algorithm repeats the process by replacing initial population with elite population and proceeds until the stopping criterion has been met. After completion, the algorithm creates model with optimal parameters for support vector machine and evaluate model with a separate set of testing data. Then, compute overall classification accuracy to assess the model's performance.

Restarting Genetic Algorithm

Input : C, Epsilon, Gamma, Number of generation T, Number of worst generations S
Output : Optimal of parameter C, Epsilon, Gamma

Method :

1. initial population p
2. evaluate fitness values for each population
3. for $t \leq T$
   - 3.1 if new generation having fitness value higher than the old population, that is, $i < S$
     - 3.1.1 Perform operations: selection, crossover, mutation
     - 3.1.2 Population replacement
     - 3.1.3 Evaluate fitness values to select elite population
   - 3.2 if new generation having fitness value less than the old population, that is, $i \geq S$
     - 3.2.1 Re-create initial population with elite population
     - 3.2.2 Evaluate fitness values
4. select the best population

Fig. 4. The pseudocode of restarting genetic algorithm.

Restarting genetic algorithm in this research is the simple genetic algorithm with the addition of condition to re-create the initial population when the new generation has fitness value less than the old population and the stopping criterion has not been met. The steps as a pseudocode in restarting genetic algorithm are shown in Fig. 4.

IV. Experimental Results

A. Dataset

To experiment our proposed classification method, we used the forest type mapping dataset from the UCI Machine Learning Repository [12]. The dataset is publicly available for testing forest type classification model over the forest area in Ibaraki Prefecture, Japan, covering 13 km × 12 km ground area. The image data were obtained from the ASTER satellite during different seasons in 2010 to 2011. The dataset has 523 data instances and 28 attributes with 4 different classes of forest: s, h, d, and o.

The class s means Sugi Forest, class h means Hinoki forest, class d means Mixed deciduous forest, and class o means other non-forest land. Fig. 5 shows image examples of each class. In our experiment, the training dataset has 198 data instances and testing dataset has 325 data instances.

Sugi forest
(Japanese cedar)

Class s

Hinoki forest
(Japanese cypress)

Class h

Mixed deciduous forest
(forest that trees lose their leaves in autumn)

Class d

Fig. 5. Examples of ground-based image data for the three main forest types. (Source: https://en.wikipedia.org/)

B. Parameter Setup

In restarting genetic algorithm process, the search for suitable values of important parameters include the probability of crossover, the probability of mutation, the population size, number of iteration, range of parameter c, gamma, epsilon, the number of restart, and the number of elite population. We search results of these parameters are summarized in Table II.

TABLE II: PARAMETER SETUP FOR RESTARTING GENETIC ALGORITHM

Probability of Crossover	0.8	C / Cost	$10^{-4} – 10^{2}$
Probability of Mutation	0.01	Gamma	$10^{-3} – 10$
Population Size	100	Epsilon	$10^{-2} – 10$
Iteration	100	Restart GA	2
Elite Population	10		

TABLE III: COMPARATIVE RESULTS FOR EACH CLASSIFICATION TECHNIQUE

Techniques	Accuracy
SVM + Default Parameters	83.69%
SVM + Genetic Algorithm (Worse)	61.54%
SVM + Genetic Algorithm (Best)	84.92%
SVM + Restarting Genetic Algorithm (Proposed Technique)	**85.23%**

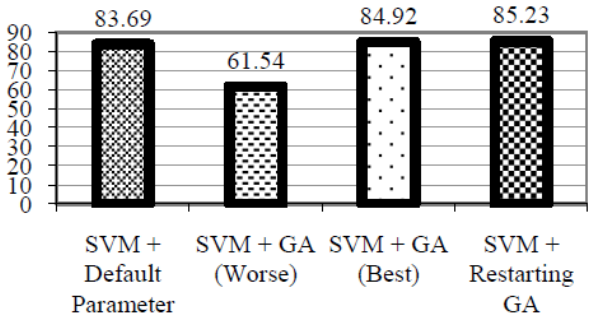

Fig. 6. Comparative chart showing the overall accuracy (in percentage) of each classification technique.

C. Results

We used overall accuracy of all four classes for evaluating performance of the classification model on recognizing the correct forest type. We compare the performance of our proposed method with other standard algorithms that are known to be high accurate. These benchmarking algorithms includes support vector machine (with default parameters), support vector machine with genetic algorithm (repeated 10 times). The comparative results are shown in Table III, and also graphically compared using a chart in Fig. 6.

From Table III, it can be seen that the proposed restarting genetic algorithm to be used with support vector machine can improve the performance for the forest type classification from the 83.69% up to the 85.23%. In case of simple genetic algorithm, we run the program repeatedly 10 times. The best accuracy from these repetitions is 84.92%. Sometimes the genetic algorithm cannot find the best parameters and this results in poor forest type classification as low as 61.54%. The results confirm that our proposed technique shows higher performance than others.

V. CONCLUSION

To improve the performance of data classification using support vector machine (SVM), we can apply the method or technique to find optimal or near-optimal set of learning parameters for SVM. One such suitable technique appropriate for optimal search is genetic algorithm. Nevertheless, the inherent problem of genetic algorithm for parameter optimization is that sometimes the simple genetic algorithm cannot find the best parameters for SVM because the randomly created initial population set does not cover the set of the best parameters.

We thus propose the technique to remedy the weak point of genetic algorithm by applying restart technique for re-generating the initial population. We add the condition to re-create the initial population when the new generation shows fitness value less than the parent population.

The experimental results show that our proposed technique can find the best parameters for support vector machine. Form our repeated experiments, we observe that our proposed technique shows constantly good performance on classifying forest types. The repetition with traditional genetic algorithm show fluctuate performances; sometimes the accuracy is quite high, but in some repetition it shows a very low performance.

REFERENCES

[1] J. Han, and M. Kamber, *Data mining: Concepts and Techniques*, Morgan Kaufmann, 2006.
[2] J. J. Liao, C. H. Shih, T. F. Chen, and M. F. Hsu, "An ensemble-based model for two class imbalanced financial problem," *Economic Modelling*, vol. 37, pp. 175-183, 2014.
[3] S. Cateni, V. Colla, and M. Vannucci, "A method for resampling imbalanced datasets in binary classification tasks for real-world problems," *Neurocomputing*, vol. 135, pp. 32-41, 2014.
[4] F. Yin, H. Mao, and L. Hua, "A hybrid of back propagation neural network and genetic algorithm for optimization of injection molding process parameters," *Materials & Design*, vol. 32, no. 6, pp. 3457-3464, 2011.
[5] M. Jamshidi, M. Ghaedi, K. Dashtian, S. Hajati, and A. Bazrafshan, "Ultrasound-assisted removal of Al 3+ ions and Alizarin Red S by activated carbon engrafted with Ag nanoparticles: Central composite design and genetic algorithm optimization," *RSC Advances*, vol. 5, no. 73, pp. 59522-59532, 2015.
[6] S. Shiff, M. Swissa, and S. Zlochiver, "A genetic algorithm optimization method for mapping non-conducting atrial regions: A theoretical feasibility study," *Cardiovascular Engineering and Technology*, vol. 7, no. 1, pp. 87-101, 2016.
[7] H. Holland, "Adaptation in natural and artificial systems," Ann Arbor: The University of Michigan Press, Michigan, 1975.
[8] R. A. C. Yang, Z. Zhou, L. Wang, and Y. Pan, "Comparison of different optimization methods with support vector machine for blast furnace multi-fault classification," *IFAC-Papers Online*, vol. 48, no. 21, pp.1204-1209, 2015.
[9] H. Zheng, L. X. Kong, and S. Nahavand, "Automatic inspection of metallic surface defects using genetic algorithms," *Journal of Materials Processing Technology*, vol. 125, pp. 427-433, 2002.
[10] C. Cortes, and V. Vapnik, "Support vector network," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.
[11] D. M. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37-63, 2011.
[12] UCI Dataset. (August 2016). *Forest Type Mapping Data Set*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Forest+type+mapping

K. Suksut is currently a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2011, master degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2013. His current research of interest includes data mining, genetic algorithm, and imbalanced data classification.

N. Kaoungku is currently a lecturer at School of Computer Engineering, Suranaree University of Technology, Thailand. He received his doctoral degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2014, bachelor degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2012, and master degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2013. His current research includes data mining and semantic web.

Kittisak Kerdprasop is an associate professor at the School of Computer Engineering and Chair of the School. He is also the head of Knowledge Engineering Research Unit, Suranaree University of Technology, Thailand. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in computer science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in computer science from Nova Southeastern University, U.S.A., in 1999. His current research includes Machine Learning, Artificial Intelligence and Probabilistic Knowledge Bases.

Nittaya Kerdprasop is an associate professor and the head of Data Engineering Research Unit, School of Computer Engineering, Suranaree Uiversity of Technology. She received her B.S. in radiation techniques from Mahidol University, Thailand, in 1985, M.S. in computer science from the Prince of Songkla University, Thailand, in 1991 and Ph.D. in computer science from Nova Southeastern University, U.S.A., in 1999. Her research of interest includes Knowledge Discovery in Databases, Data Mining, Artificial Intelligence, Logic and Constraint Programming, Deductive and Active Databases.