Gradient Evolution-based Support Vector Machine Algorithm for Classification

Ferani E. Zulvia* and R. J. Kuo

1Department of Logistics Engineering, Universitas Pertamina, Teuku Nyak Arief Road, Simprug, Kebayoran Lama, Jakarta, Indonesia
2Department of Industrial Management, National Taiwan University of Science and Technology, No. 43, Section 4, Kee-Lung Road, Taipei, Taiwan

*corresponding author. Email: Feranieva.z@universitaspertamina.ac.id

Abstract. This paper proposes a classification algorithm based on a support vector machine (SVM) and gradient evolution (GE) algorithms. SVM algorithm has been widely used in classification. However, its result is significantly influenced by the parameters. Therefore, this paper aims to propose an improvement of SVM algorithm which can find the best SVMs’ parameters automatically. The proposed algorithm employs a GE algorithm to automatically determine the SVMs’ parameters. The GE algorithm takes a role as a global optimizer in finding the best parameter which will be used by SVM algorithm. The proposed GE-SVM algorithm is verified using some benchmark datasets and compared with other metaheuristic-based SVM algorithms. The experimental results show that the proposed GE-SVM algorithm obtains better results than other algorithms tested in this paper.

1. Introduction
Classification is one of data mining approach which has been widely applied in many applications. Therefore, many studies have proposed classification methods such as decision tree, support vector machine, k-nearest neighbour, etc. [1]. Among these methods, support vector machine (SVM) has become popular among classification techniques. SVM is a supervised learning model which construct a set of hyperplanes to separate the dataset [2]. Although it has been applied in many applications and shown a good performance, this algorithm highly depends on the parameters [3, 4]. Therefore, determining SVM parameter is a crucial issue before applying SVM algorithm. Many studies have attempted to overcome this problem [5-7]. Some of the proposed methods employ metaheuristic to find the best parameters for the SVM. Lin, Lee, Chen and Tseng [8] use a simulated annealing (SA) algorithm, Huang and Wang [9] apply genetic algorithm (GA) algorithm, Lin, Ying, Chen and Lee [10] use particle swarm optimization (PSO) algorithm, and etc. [11, 12]. This paper also employs a metaheuristic, named gradient evolution (GE) algorithm to automatically tuning the SVM parameters. GE algorithm is a new metaheuristic inspired from Newton-Raphson algorithm [13]. It explores the search space using three operators, namely the vector updating, jumping, and refreshing. According to the previous researches, GE algorithm has promising results [13, 14]. Therefore, this paper attempts to apply GE algorithm to automatically optimize SVMs’ parameters.
The reminder of this paper is organized as follows. Section two presents the literature review. Section three discusses the proposed algorithm while the experimental results are discussed in Section four. Finally, the concluding remarks are made in Section five.

2. Literature Review
This section briefly reviews some basic theories applied in this paper.

2.1. Support Vector Machine
SVM has been used for classifying both linear and nonlinear separable data. In order to classify nonlinear data, SVM uses a Kernel function. This paper applies a Gaussian Kernel  as given in Eq. (1).

\[ k(\bar{x}, \bar{y}) = e^{-\frac{|\bar{x} - \bar{y}|^2}{2\sigma^2}} \],

where  \( \bar{x}, \bar{y} \in \mathbb{R}^n \) and  \( \sigma > 0 \) is a free parameter. The non-linear dual Lagrangian model of non-linear SVM can be defined in Eq. (2).

\[ \hat{f}(\bar{x}) = \text{sgn} \left( \sum_{i=1}^{l} \alpha_i^* y_i k(\bar{x}_i, \bar{x}) + b^* \right), \]

The training algorithm in terms of a kernel function is given by Eq. (3).

\[ \alpha^* = \arg\max_{\alpha} \left( \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(\bar{x}, \bar{y}) \right), \]

Subject to,

\[ 0 \leq \alpha_i \leq C, i = 1, ..., m \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0, \]

where \( l \) is the number of support vectors in the trained model and \( C \) is weight parameter.

2.2. Gradient Evolution Algorithm
Gradient Evolution (GE) algorithm is a new metaheuristic developed from Newton-Raphson Optimization algorithm [13]. This algorithm represents the solution as a vector. GE algorithm explores the search space using three operators, vector updating, jumping, and refreshing. Vector updating focuses on deep search while vector jumping focuses on wide search. The last operator, vector refreshing is an operator which manage the diversity in the population.

3. Methodology
This paper applies GE algorithm to fine tune SVMs’ parameters. There are two parameters in non-linear SVM using Gaussian, number of support vectors in the trained model and weight parameter. GE algorithm is employed to optimize these parameters. In GE algorithm, they are represented as a vector. Fitness of each vector is defined as the error of the training test. These parameters are improved iteratively. After all iterations, the final model obtained using the best parameter is used for predicting the testing error.

The flowchart of the proposed GESVM is given in Figure 1 while the procedure is as the following.

Step 1: Set parameter for GE: number of vectors, number of iterations, jumping rate, reduction rate, and minimum saturation.

Step 2: Generate initial vectors,  \( P^t = \{X_i^t \}_{i=1}^{L} \). 

Step 3: Calculate fitness of each vector by performing SVM with parameter as given by the vector. Use the average of training error as the fitness.

Step 4: Record the best vector.

Step 5: For each iteration,  \( t \), update all vectors.

Step 6: For each vector,  \( p \), perform vector updating towards  \( p \).
\[ u_{ij}^t = x_{ij}^t - \left( r'_a \cdot \frac{\Delta x_{ij}^t}{2} \right) \cdot \left( \frac{x_{ij}^W - x_{ij}^B}{x_{ij}^W - 2x_{ij}^t + x_{ij}^B} \right) + r_a \cdot (y_j - x_{ij}^t), \forall j = 1, \ldots, D, \]  
\[ \Delta x_{ij}^t = \frac{|x_{ij}^t - x_{ij}^j| + |x_{ij}^W - x_{ij}^j|}{2}, \forall j = 1, \ldots, D \]  
where \( u_{ij}^t \) is a mutant vector, \( y_j \) is the best vector, \( x_{ij}^W \) are \( x_{ij}^W \) a better and a worse vector than vector \( x_{ij}^j \), respectively. The \( r'_a \) and \( r_a \) are random numbers. For the best vector, \( x_{ij}^W \) is replaced by \( b_j \) defined in Eq. (7).

\[ b_j = x_{ij}^j - \Delta x_{ij}^j, \forall j = 1, \ldots, D. \]  
Where,

\[ \Delta x_{ij}^j = \frac{\gamma + |x_{ij}^W - x_{ij}^j|}{2}, , \forall j = 1, \ldots, D. \]  
On the other hand, \( x_{ij}^W \) is replaced by \( w_j \) defined in Eq. (9).

\[ w_j = x_{ij}^j + \Delta x_{ij}^j, \forall j = 1, \ldots, D \]  
Where,

\[ \Delta x_{ij}^j = \frac{|x_{ij}^j - x_{ij}^B| + \gamma}{2}, \forall j = 1, \ldots, D, \]  
where \( \gamma \) is a pre-defined parameter, defined as initial step size.

Step 7: Generate a random number \( r \) between \([0,1]\). If \( r \) less than jumping rate, perform vector jumping towards vector \( p \).

\[ u_{ij}^t = -u_{ij}^t + r_m \cdot (u_{ij}^t - x_{ij}^k), \forall j = 1, \ldots, D, \]  
where \( x_{ij}^k \in \mathbf{X}_k^t \) is a random neighbor vector in \( \mathbf{P}^t, \forall i \neq k \) and \( r_m \) is a random number in range \([0,1]\).

Step 8: Calculate fitness
Step 9: If the new fitness is worse than the previous fitness, use the previous vector \( p \) as the next vector \( p \) and reduce the vector saturation \( p \).
Step 10: If vector saturation \( p \) is less than minimum saturation, regenerate vector \( p \).
Step 11: Update the best vector.
Step 12: Back to Step 5 until the end of iteration.

| Table 1. Parameter Setting.  |
|-------------------------------|
| Algorithm  | Parameter | Value   |
| GESVM      | Jumping rate | 0.3     |
|            | Reduction rate | 0.05   |
|            | Minimum saturation | 0.2    |
|            | Initial Delta | 0.1     |
| PSOSVM     | Inertia weight | 0.8     |
|            | Learning rate 1 | 1.5     |
|            | Learning rate 2 | 1.5     |
| DESVM      | Crossover rate | 0.2     |
|            | Alpha | 0.5     |

4. Experimental Results and Analysis
The proposed algorithm is tested using two benchmark datasets, WDBC and Ionosphere taken from UCI repository. The WDBC dataset comprises of 569 instances with 30 attributes while ionosphere dataset has 351 instances with 34 attributes. For each dataset, 10-fold cross validation is applied. The results are compared with other metaheuristic-based SVM algorithms. They are PSO-SVM and DE-SVM algorithms. Since in this problem the metaheuristic algorithm focuses on performing global exploration, the parameter setting for metaheuristic algorithm is also set to achieve this purpose. Table 1 lists the parameter setting for each algorithm.
Figure 1. The proposed GESVM Flowchart

Table 2. Experimental Results.

| Dataset   | comparator | GESVM Training error | Testing error | PSOSVM Training error | Testing error | DESVM Training error | Testing error |
|-----------|------------|----------------------|---------------|------------------------|---------------|----------------------|---------------|
| WDBC      | average    | 0.00E+00             | 1.22E-01*     | 0.00E+00               | 1.28E-01     | 0.00E+00             | 2.24E-01     |
|           | St. deviation min | 0.00E+00         | 1.41E-02     | 0.00E+00               | 9.36E-03*    | 0.00E+00             | 5.43E-02     |
|           | max        | 0.00E+00             | 9.83E-02*     | 0.00E+00               | 1.11E-01     | 0.00E+00             | 1.47E-01     |
| Ionosphere| average    | 0.00E+00             | 2.64E-01*     | 0.00E+00               | 2.70E-01     | 0.00E+00             | 2.77E-01     |
|           | St. deviation min | 0.00E+00         | 3.43E-02*     | 0.00E+00               | 2.44E-02     | 0.00E+00             | 2.26E-02     |
|           | max        | 0.00E+00             | 1.96E-01*     | 0.00E+00               | 2.25E-01     | 0.00E+00             | 2.54E-01     |

*: best result

Ten independent runs are executed for each algorithm. Table 2 summaries the result. A further statistical test analysis is also conducted to evaluate if the results of these algorithms are significantly different. Table 3 gives the statistic test result. These tables show that combining SVM with metaheuristic helps
SVM to get very low training error. In terms of testing error, GESVM algorithm obtains significantly better results for WDBC datasets. For ionosphere dataset, GESVM algorithm still get better result but not significantly different than PSOSVM and DESVM algorithms. The reason might be because GE algorithm has operator to avoid local optima and maintaining diversity in the population. On the other hand, PSOSVM and DESVM do not have a specific operator focused on avoiding local optimal and maintaining diversity in the population.

| Testing          | WDBC | Ionosphere |
|------------------|------|------------|
| GESVM vs PSOSVM  | 0.499| 0.801      |
| GESVM vs DESVM   | 0.006*| 0.556      |
| PSOSVM vs DESVM  | 0.008*| 0.679      |

*: significantly different

5. Conclusion
SVM algorithm is a popular classification algorithm. Many studies have shown good performance of SVM algorithm. However, SVM involves parameters which significantly influence the result. This paper proposes an algorithm which can automatically fine tune SVMs’ parameters. The proposed algorithm employs a metaheuristic named GE algorithm as the parameter generator. This algorithm helps to find the best parameter setting for SVM algorithm. The proposed algorithm is verified using two benchmark datasets and compared with PSOSVM and DESVM algorithms. The results show that the proposed GESVM can provide better results than PSOSVM and DESVM algorithms. It might be happened due to GESVM has two operators focus on wide exploration. They are the vector jumping and vector refreshing. This study should be extended by testing more datasets and comparing with more algorithms. Other approaches to optimize SVM parameters are also should be further studied.

References
[1] Fayyad U M, Piatetsky-Shapiro G, Smyth P and Uthurusamy R 1996 Advances in knowledge discovery and data mining (CA, USA: American Association for Artificial Intelligence Menlo Park)
[2] Kuo R J, Mei C H, Zulvia F E and Tsai C Y 2016 An application of a metaheuristic algorithm-based clustering ensemble method to APP customer segmentation Neurocomputing 205 116-29
[3] Matheny M E, Resnic F S, Arora N and Ohno-Machado L 2007 Effects of SVM parameter optimization on discrimination and calibration for post-procedural PCI mortality Journal of Biomedical Informatics 40 688-97
[4] Cao L J and Tay F E H 2003 Support vector machine with adaptive parameters in financial time series forecasting IEEE Transactions on Neural Networks 14 1506-18
[5] Devos O, Ruckebusch C, Durand A, Duponchel L and Huvenne J-P 2009 Support vector machines (SVM) in near infrared (NIR) spectroscopy: Focus on parameters optimization and model interpretation Chemometrics and Intelligent Laboratory Systems 96 27-33
[6] Min J H and Lee Y-C 2005 Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters Expert Systems with Applications 28 603-14
[7] Guo H, Liu H and Wang L 2006 Method for selecting parameters of least squares support vector machines and application Journal of System Simulation 18 2033-6
[8] Lin S-W, Lee Z-J, Chen S-C and Tseng T-Y 2008 Parameter determination of support vector machine and feature selection using simulated annealing approach Applied Soft Computing 8 1505-12
[9] Huang C-L and Wang C-J 2006 A GA-based feature selection and parameters optimizationfor support vector machines Expert Systems with Applications 31 231-40
[10] Lin S-W, Ying K-C, Chen S-C and Lee Z-J 2008 Particle swarm optimization for parameter determination and feature selection of support vector machines *Expert Systems with Applications* **35** 1817-24

[11] Lorena A C and de Carvalho A C P L F 2008 Evolutionary tuning of SVM parameter values in multiclass problems *Neurocomputing* **71** 3326-34

[12] Huang H-L and Chang F-L 2007 ESVM: Evolutionary support vector machine for automatic feature selection and classification of microarray data *Biosystems* **90** 516-28

[13] Kuo R J and Zulvia F E 2015 The gradient evolution algorithm: A new metaheuristic *Information Sciences* **316** 246-65

[14] Kuo R J and Zulvia F E 2016 Cluster analysis using a gradient evolution-based k-means algorithm. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*, pp 5138-45