World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:12, 2017

Integrated ACO_R/$IACO_{MV-R}$-SVM Algorithm

Hiba Basim Alwan, Ku Ruhana Ku-Mahamud

Abstract**—**A direction for ACO is to optimize continuous and mixed (discrete and continuous) variables in solving problems with various types of data. Support Vector Machine (SVM), which originates from the statistical approach, is a present day classification technique. The main problems of SVM are selecting feature subset and tuning the parameters. Discretizing the continuous value of the parameters is the most common approach in tuning SVM parameters. This process will result in loss of information which affects the classification accuracy. This paper presents two algorithms that can simultaneously tune SVM parameters and select the feature subset. The first algorithm, ACO_R-SVM, will tune SVM parameters, while the second $IACO_{MV-R}$-SVM algorithm will simultaneously tune SVM parameters and select the feature subset. Three benchmark UCI datasets were used in the experiments to validate the performance of the proposed algorithms. The results show that the proposed algorithms have good performances as compared to other approaches.

Keywords**—**Continuous ant colony optimization, incremental continuous ant colony, simultaneous optimization, support vector machine.

I. Introduction

CLASSIFICATION is a major requirement that appears in problems such as object detection, face recognition, medical testing, credit card fraud forecasting, and machine fault detection. It is a necessary part of data mining and pattern recognition.

The main goal of the classification problem is to assign any given input to exactly one output [1] or presenting an object to one of a number of separate groups or classes built on a set of features known as the feature vector [2]. Pattern classification is a significant topic that is broadly exploited in decision-making [3]. Approaches such as a Neural Network (NN), Ant Colony Optimization (ACO), Fuzzy, SVM, and hybrid approach have been used to classify patterns.

One of the most successfully used approaches is SVM, which is a supervised machine learning approach. SVM classifiers have been applied for credit risk analysis, medical diagnostics, text categorization, and information extraction [4]. The SVM learning approach explores to obtain the isolating hyperplane that increases the margin among the patterns in the classes it is isolating, and these patterns operate as the support vectors. Practically, this is like taking into consideration the long term classification aim in contrast to lying for the first discriminant, which led to no training mistake [5]. In SVM, kernel functions are used to put data into high dimensionality domains. This is in accordance with the concept of structural risk minimization. Among the popular kernel functions are RBF, polynomial, sigmoid, and linear. The most popular kernel function is RBF as it is capable to manage high dimensional data with only one kernel parameter called gamma. However, RBF does not produce good performance when the number of feature is large. There are two problems associated with SVM classifier that influence the classification accuracy. The problems are selecting the optimal feature subset and tuning SVM parameters to be used in the SVM classifier. These problems affect each other. Another minor SVM problem is selecting suitable kernel functions [6]. In order to find the optimal feature subset for SVM, optimization techniques such as Tabu Search (TS), Genetic Algorithm (GA), Simulated Annealing (SA), ACO and Particle Swarm Optimization (PSO) have been applied [7]. In tuning the SVM parameters' values, approaches such as trial and error, generalization error estimation, grid, cross validation, gradient descent, and global stochastic optimization (such as GA, PSO, SA, and ACO) can be utilized [8].

ACO is one of the promising approaches in solving SVM problems. ACO was first introduced by Dorigo in 1991 and is an arbitrary seeking approach through simulating the nature of ants foraging operation. In ACO, the ant colony obtains the optimal way through detecting pheromones. In the operation of foraging, ants put pheromones on the way and can detect the strength of pheromones. The more the strength of pheromones means the more the possibility to choose the trail. The ACO approach includes two phases: (1) computing transition probability; and (2) pheromone updating [9]. ACO was first presented to solve discrete optimization problems and later modified to solve continuous and mixed optimization problems. However, several studies have expanded the use of ACO, by applying it to continuous and mixed-variables optimization problems [10]-[14]. One of the most interesting ACOs for continuous variables and mixed-variables works is [10]. Studies on continuous ACO (ACO_R) and mixed-variable ACO (ACO_{MV}) can be seen in [10]-[14]. ACO_R is considered the first algorithm that can handle the continuous variable. It follows the same ACO framework. ACO_R was later modified by [13], [14], who introduced two new algorithms called Incremental ACO_R ($IACO_R$) and Incremental ACO_R with Local Search ($IACO_R$-LS). In this paper, the authors focused on ACO_R and $IACO_{MV-R}$, which is based on $IACO_R$. This algorithm is considered as the first algorithm that can handle mixed variables and follows the same ACO framework.

The authors wish to thank the Ministry of Higher Education Malaysia for funding this study under the Long Term Research Grant Scheme, S/O codes 12490 and RIMC, Universiti Utara Malaysia, Kedah for the administration of this study.

Hiba Basim Alwan is with School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia (corresponding author; phone: 009647901575059; e-mail: hiba81basim@ yahoo.com).

Ku Ruhana Ku-Mahamud is with School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia (e-mail: ruhana@uum.edu.my).

IACO$_R$ is used to improve ACO$_R$'s performance through the seek diversification mechanism as well as through solving the stagnation problem by using the restarting mechanism. A number of successive iterations with an improvement in the solution lower than a certain threshold is used as the restart condition. The same classical ACO framework has been adopted by these variants except for the discrete probability used to build the ant solution, which is replaced by the continuous probability

This paper proposes two algorithms to tune the SVM parameters and select the SVM feature subset. The rest of the paper is organized as follows. Section II presents the proposed algorithms. Experimental results are discussed in the third section, while the concluding remarks and future works are presented in the final section.

II. The Proposed Algorithm

The algorithms that are proposed in this paper are related to optimizing two SVM parameters. The parameters are: i) weight, *C*; and ii) kernel function. The weight represents the trade-off between misclassifying certain points and correctly classifying others, while the kernel is used to simultaneously tune SVM parameters and select the feature subset.

The first algorithm, ACO$_R$-SVM, is to optimize SVM parameters. It is based on ACO$_R$, which is considered as the first ACO algorithm used to optimize continuous variables [10]. Fig. 1 depicts the pseudocode of the proposed ACO$_R$-SVM algorithm.

```
ACO_R-SVM Algorithm
Input: $k$, $m$, $q$, $C$, $\gamma$, and termination criterion
Output: Optimal value for SVM parameters and classification
        accuracy
Begin
  Initialize $k$ solutions
  call SVM algorithm to evaluate $k$ solutions
    $T$ = Sort ($S_1$, ..., $S_k$)
  while classification accuracy ≠ 100% or number of iteration ≠
  10 do
    for $i$ = 1 to $m$ do
        select $S$ according to its weight
        sample selected $S$
        store newly generated solutions
        call SVM algorithm to evaluate newly generated
        solutions
    end
    $T$ = Best (Sort $S_1$, ... $S_k$ + $m$), $k$)
  end
End
```

Fig. 1 Pseudocode of proposed ACO$_R$-SVM algorithm

In the algorithm, *k* is the size of solution archive, *m* is the number of ants that are used to generate solutions, *q* is the algorithm's parameter to control diversification of the search process, *C* is the regularization or soft margin parameter, *γ* is the kernel function parameter called the margin or the width parameter, and finally, the termination conditions for the best values for SVM parameters (*C* and *γ*).

ACO$_R$ differs from ACO in the way of computing the transition probability. It uses the Probability Density Function (PDF) in computing the weight vector, *w*, as follows:

$$w_l = \frac{1}{qk\sqrt{2\pi}} e^{-\frac{(l-1)^2}{2q^2k^2}} \qquad (1)$$

PDF is used by ACO$_R$ because it deals with variables that are continuous in nature, unlike the classical ACO, which was established to deal with discrete variable.

The second algorithm, IACO$_{MV\text{-}R}$-SVM, is to simultaneously tune SVM parameters and select the feature subsets by using IACO$_{MV\text{-}R}$. This is a direction of processing the data, where SVM is utilized to simultaneously optimize model selection and feature subset selection to improve classification accuracy. This is to avoid the two processes from affecting each other with the error that each process might introduce [15], [16]. IACO$_{MV\text{-}R}$ has the ability to optimize mixed-variable problems, namely the continuous variables for the SVM parameters and feature subset. The features are represented as discrete graph nodes.

Fig. 2 depicts the pseudocode of the proposed IACO$_{MV\text{-}R}$-SVM algorithm.

```
IACO_MV-R-SVM Algorithm
Input: $k$, $m$, $C$, $\gamma$, features, and termination condition
Output: classification accuracy, optimal feature subset, and
        optimal value for SVM parameters
Begin
  $C$ → solution archive$_C$
  $\gamma$ → solution archive$_\gamma$
  features → solution archive$_{feature}$
  call SVM algorithm to evaluate the initialize solution in
solution archive
  while classification accuracy ≠ 100% or number of iteration
≠ 10 do
      for $n$ = 1 to $m_{ants}$ do
        call ACO$_{MV\text{-tune SVM parameter}}$
        call ACO$_{MV\text{-feature subset selection}}$
        call SVM algorithm to evaluate the newly built solution
      end
      solution archive = first ← Rank ($S_{old}$ U $S_1$, ...$S_{Nants}$)
      update solution archives
  end
End
```

Fig. 2 Pseudocode for IACO$_{MV\text{-}R}$-SVM algorithm

In the algorithm, *k, m, q, C,* and *γ* parameters are the same as in the first algorithm depicted in Fig. 1. Features are the datasets' attributes, and the termination conditions will produce the best combination values for *C* and *γ* parameters as well as the selected feature subset.

A function named ACO$_{MV\text{-tune SVM parameter}}$ is used to tune SVM parameters, whereas another function known as ACO$_{MV\text{-feature subset selection}}$ is used to select the feature subset. These functions will be called by the second proposed algorithm. The pseudocodes for both functions are shown in Fig. 3 and Fig. 4, respectively.

```
$\text{ACO}_{\text{MV-tune SVM parameter}}$
Input: $p$, InitArhiveSize, Growth, MaxArchiveSize, MaxStagIter,
        $m$, and termination criterion
Output: Optimal Value for $C$ and $\gamma$
Begin
  $k$ = InitArhiveSize
  initialize $k$ solutions
  call SVM algorithm to evaluate $k$ solutions
  while classification accuracy $\neq$ 100% or number of   iteration
$\neq$ 10 do
      if $rand\ (0,1) < p$ then
        for $i$ = 1 to no. of ants do
          Select best solution
          Sample best selected solution
          Call SVM algorithm to evaluate the new generated
          solutions
          if Newly generated solution is better than $S_{best}$ then
            Substitute newly generated solution for $S_{best}$
          end
      end
    else
      for $j$ = 1 to $k$ do
        Select $S$
        Sample selected $S$
        Store newly generated solutions
        Call SVM algorithm to evaluate the new generated
        solutions
        if Newly generated solution is better than $S_j$ then
          Substitute newly generated solution for $S_j$
        end
      end
    end
    if current iterations are multiple of Growth & $k$ <
    MaxArchiveSize then
      Initialize new solution
      Add new solution to the archive
      $k++$
    end
    if # (number) of iterations without improving   classification
accuracy of $S_{best}$ = MaxStagIter then
      Re-initialize $T$ (solution archive) but keeping $S_{best}$
    end
End
```

Fig. 3 Pseudocode for tuning SVM parameters

In Fig. 3, *k, m, C,* and *γ* parameters are the same as in the first algorithm depicted in Fig. 1. $p \in [0, 1]$ will monitor the probability of utilizing just the best solution in the archive as a directing solution. The growth rate of the archives will be monitored by a parameter *Growth. InitArhiveSize* and *MaxArchiveSize* are the minimum and maximum solution archive sizes.

$\text{ACO}_{\text{MV-tune SVM parameter}}$ will tune the continuous value of SVM parameters through IACO_R [13, 14]. An ant will establish its solution, which includes the *C* value, the width, *γ,* for RBF kernel parameter, and the feature subset. This constructed solution will be sent to SVM together with the feature subset generated by the second function, $\text{ACO}_{\text{MV-feature subset selection}}$.

III. Experimental Result

The proposed algorithms were tested on three datasets from the UCI repository [17]. Three UCI datasets were used in the experiments to evaluate the proposed algorithms. The datasets are Pima Indians Diabetes, Splice, and Image Segmentation.

The Image Segmentation dataset was generated by Vision group, University of Massachusetts. This dataset contained 2,310 instances with 19 features, divided into seven classes (brick face, sky, foliage, cement window, path, and grass). There are no missing data in this dataset.

```
$\text{ACO}_{\text{MV-feature subset selection}}$
Input: Features
Output: Optimal feature subset
Begin
  calculate features subset size randomly
  initialize pheromone table
  for $i$ = 1 to no. of features do
    compute weight for each feature
    compute probability for each feature
    select feature with highest probability
     append feature with highest probability to features subset
     remove appended feature from original features set
  end
  for $j$ = 1 to features subset size - 1 do
    compute probability for remaining features
    select feature with highest probability
    append feature with highest probability to features
    subset
   remove appended feature from original features set
  end
  update pheromone table
End
```

Fig. 4 Pseudocode for feature subset selection

The Pima Indians Diabetes dataset was collected from female patients of at least 21 years old. It includes 768 instances with 8 features, divided into two classes. The classes are 500 positive tested for diabetes instances and 268 not positive tested for diabetes instance. There were missing values in this dataset.

Splice junctions are positions on DNA series at which extra DNAs are eliminated during the task of protein generation in higher organisms. The problem presented in this dataset is to recognize the boundaries between exons and introns when a series of DNA is given. Exons are the parts of the DNA sequence retained after splicing, while introns are the parts of the DNA sequence that are spliced out. The problem comprises recognizing exon/intron boundaries (referred to as EI sites acceptors) and recognizing intron/exon boundaries (IE sites donors). This dataset includes 3,190 instances with 61 features, divided into three classes: 767 EI, 768 IE, and 1,655. The dataset contains no missing data.

The ten-fold cross validation (CV) technique has been applied in the experiments. This will enable each fold of data to be used as the testing datasets, which resulted in ten sets of classification accuracy. Finally, the experiment averages the accuracies and calculates the standard deviations. The results were compared with the ACO-SVM algorithm [18].

The proposed algorithm was implemented using C programming language. The experiments were performed on an Intel(R) Core (TM) 2 Duo CPU T5750, with 32-bit operating system, 4 GB RAM, and running at 2.00 GH_Z. The experiment parameters are displayed in Table I.

TABLE I
EXPERIMENT PARAMETERS

C	***γ***	***m***
$[2^{-1}, 2^{12}]$	$[2^{-12}, 2^{2}]$	2, 4, 6, 8, 10
q	**Initial *k* size**	**Maximum *k* size**
0.1, 0.3, 0.5, 0.7, 0.9	2, 4, 6, 8, 10, 12, 14	3, 5, 7, 9, 11, 13, 15

The results show that the performance of the proposed algorithms did not depend on the number of ants. Thus, only two ants were selected to execute the proposed algorithms. Also, the results show that good performance was obtained for a small value of *q*. For this study, the value of *q* was set to 0.1. Finally, the results show that the best values for the *Growth* parameter and *Stag* were 5 and 2, respectively. The best values for the initial solution archive size and maximum solution archive size are 10 and 15, respectively.

Table II displays the performances of the proposed algorithms against ACO-SVM [18]. ACO_R-SVM produced better results than ACO-SVM because discrete variable was not discretized. The error that might occur in the discretization process of ACO-SVM can be avoided. Better results were produced by the $IACO_{MV-R}$-SVM algorithm for all three datasets because of the simultaneous action of tuning the SVM parameters and selecting the feature subset. This can eliminate the error that might occur if the tuning and feature selection processes were performed in sequence because both processes can intensify the error produced by either processes.

TABLE II
CLASSIFICATION ACCURACY OF THE PROPOSED ALGORITHMS AND ACO-SVM ALGORITHM

Dataset	**ACO-SVM**	**ACO_R-SVM**	**$IACO_{MV-R}$-SVM**
Image Segmentation	94.76	98.00 ± 0	98.96 ±0.41
Pima-Indians Diabetes	76.28	88.00 ± 0	97.22 ± 0.81
Splice	94.65	96.72 ± 0.18	98.65 ±0.55

IV. CONCLUSION

The proposed algorithms were able to produce better results than ACO-SVM because it can handle continuous and mixed values of SVM. This will avoid error in converting continuous value to discrete value. It is also found that simultaneous action can reduce the error if the processes were executed in sequence. Better classification accuracy can be obtained if the processes of parameter tuning and feature selection were performed in parallel as shown by the $IACO_{MV-R}$-SVM algorithm.

The authors suggested applying the proposed algorithms on Support Vector Regression (SVR), because SVR has the same problems as SVM. The proposed algorithms can be tested in solving dynamic problems. In addition, the least square SVM could be used in solving classification problems. Future work on classification could also focus on using other kernel functions.

REFERENCES

[1] Chen, W., Chun-Na, L., Hua-Xin, P., Yan-Ru, G., & Yuan-Hai, S. (2017). Alternating direction method of multipliers for L1- and L2-norm best fitting hyper plane classifier. Procedia Computer Science, 108C (2017), 1292-1301.

[2] Kojo, G, James, B., Andrew H., & Elena, G. (2017). Linear classifier design under heteroscedasticity in linear discriminant analysis. Expert System with Applications, 79 (15 Augest 2017), 4-52.

[3] Acharya, J., Das, A., Orlitsky, A., Pan, S. & Santhanam, N. (2010). Classification using pattern probability estimators. Proceeding of The International Symposium on Information theory, 13-18 June 2010, Austin, TX, 1493 - 1497.

[4] L., Demidove, I., Klyueva, Y., Sokolova, N., Stepanov, & N., Tyart. (2017). Intellectual approaches to improvement of the classification decisions quality on the base of the SVM classifier. Procedia Computer Science, 103 (2017), 222-230.

[5] Craig, V., Stephen, V., Conrad, J., James, A., & Gregory, H. (2015). MapReduce SVM game. Procedia Computer Science, 53 (2015), 298-307.

[6] Wang, T., Huang H., Tian S. & Xu J. (2010). Feature selection for SVM via optimization of kernel polarization with gaussian ARD kernels. Expert Systems with Applications, 37(9), 6663-6668.

[7] Basiri, M., Aghaee, N. & Aghdam, M. (2008). Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins. In E. Marchiori & J. Moore (Eds.), Evolutionary computation, machine learning and data mining in bioinformatics, (pp. 12-23). Berlin Heidelberg: Springer.

[8] Qiu, D., Li, Y., Zhang, X. & Gu, B. (2011). Support vector machine with parameter optimization by bare bones differential evolution. Proceedings of The 7th International Conference of the Natural Computation, 26-28 July 2011, Shanghai, 263-266.

[9] Wen, S., Yao, M., Wu, C., & Li, J. (2017). An improved fastSLAM2.0 algorithm based on ant colony optimization. Proceedings of The 29th Conference of Control and Decision, 28-30 May 2017, Chongqing, China, 1948-9447.

[10] Socha, K. (2008). Ant colony optimization for continuous and mixed-variables domain. (Doctoral dissertation, Universite' Libre de Bruxelles, 2008). Retrieved from *iridia.ulb.ac.be/~mdorigo/HomePageDorigo/thesis/SochaPhD.pdf.*

[11] Socha, K. (2004). ACO for continuous and mixed-variable optimization. In M. Dorigo, M. Birattari, C. Blum, L. Gambardella & F. Mondada (Eds.), Ant Colony Optimization and Swarm Intelligence, (pp. 25-36). Berlin Heidelberg: Springer.

[12] Socha, K. & Dorigo, M. (2008). Ant colony optimization for continuous domain. European Journal of Operational Research, 185(3), 1155-1173.

[13] Liao, T. (2011). Improved ant colony optimization algorithms for continuous and mixed discrete-continuous optimization problems (Report No. 2010/2011). Retrieved from IRIDIA, Universite Libre de Bruxelles: Brussels, Belgium website: http://iridia.ulb.ac.be/~tliao/papers/DEAreport-Liao.pdf.

[14] Liao, T., Oca, M., Aydin, D., Stützle, T. & Dorigo, M. (2011). An incremental ant colony algorithm with local search for continuous optimization (Report No. TR/IRIDIA/2011-005). Retrieved from IRIDIA, Universite Libre de Bruxelles: Brussels, Belgium website: http://iridia.ulb.ac.be/IridiaTrSeries/rev/IridiaTr2011-005r002.pdf.

[15] Alwan, H.B. and Ku-Mahamud, K.R. (2017). Mixed-variable ant colony optimization algorithm for feature subset selection and tuning SVM parameter. *International Journal of Bio-Inspired Computational.* 9(1), 53-63.

[16] Alwan, H. B. & Ku-Mahamud, K. R. (2013). Feature selection and model selection algorithm using incremental mixed variable ant colony optimization for support vector machine classifier. *International Journal of Mathematics and Computers in Simulation.* 7(5), 406-414.

[17] *UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine, CA, <http://www.ics.uci.edu/mlearn/MLRepository>, 2012.*

[18] Huang, C. (2009). ACO-based hybrid classification system with feature subset selection and model parameters optimization. *Neurocomputing,* 73(1-3), 438-448.

Hiba Basim Alwan was born in Baghdad, Iraq on 23rd July 1981. She holds a Bachelor's degree in Computer Science from Al-Mansour University College, Iraq in 2004, and a Master's degree in Computer Science from University of

Technology, Iraq in 2006. Her PhD in Computer Science was obtained from Universiti Utara Malaysia in 2014. Now, she is a postdoc researcher at Universiti Utara Malaysia. As an academic, her research interests include ant colony optimization, pattern classification, support vector machine, and big data.

Ku Ruhana Ku-Mahamud holds a Bachelor's degree in Mathematical Sciences and a Master's degree in Computing, both from Bradford University, United Kingdom in 1983 and 1986, respectively. Her PhD in Computer Science was obtained from Universiti Pertanian Malaysia in 1994. As an academic, her research interests include ant colony optimization, pattern classification, and vehicle routing problem.