Metaheuristic Techniques for Support Vector Machine Model Selection

James Blondin and Ashraf Saad
Computer Science
Armstrong Atlantic State University
Savannah, GA 31419, USA
Email: jblondin@gmail.com, ashraf.saad@armstrong.edu

***Abstract*—The classification accuracy of a Support Vector Machine is dependent upon the specification of model parameters. The problem of finding these parameters, called the model selection problem, can be very computationally intensive, and is exacerbated by the fact that once selected, these model parameters do not carry across from one dataset to another.**

This paper describes implementations of both Ant Colony Optimization and Particle Swarm Optimization techniques to the SVM model selection problem. The results of these implementations on some common datasets are compared to each other and to the results of other SVM model selection techniques.

***Keywords*-Support Vector Machines; Metaheuristics; Ant Colony Optimization; Particle Swarm Optimization**

I. Introduction

This paper discusses the application of Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) techniques to the problem of finding the optimal set of parameters of a Support Vector Machine (SVM) classifier, referred to as the *SVM model selection problem*.

Support vector machines [1] have successfully demonstrated their effectiveness at classifying many types of datasets [2]. However, they can be very slow to train and require the specification of control parameters.

These SVM parameters have a substantial impact on the SVM's classification accuracy. For example, in the case of one of datasets tested, proper choice of the SVM parameters raised the SVM's classification accuracy from less than 55% at the lowest to over 96%. The process of finding these parameters, however, requires multiple training runs of the SVM, which is a computationally intensive process. Even once discovered, the parameter values at which the SVM exhibits the best classification performance are only optimal for a particular dataset. In applications that must classify either a large number of different datasets or datasets that are constantly changing, a fast and effective SVM parameter optimization method is essential.

In Section II, we briefly describe the SVM model selection problem. In Section III, we discuss PSO and ACO optimization techniques used for the SVM model selection problem and introduce our implementation of the ACO algorithm as applied to the SVM model selection problem which we call APS-SVM. In Section IV, we describe specific implementation details we used to to conduct our tests of the PSO and APS-SVM algorithms. In Section V, we present the results of these tests and analyze and compare the characteristics of the implemented algorithms. Finally, in Section VI, we discuss additional ways in which the performance of these SVM parameterization techniques can be analyzed.

II. SVM Model Selection

In this and the following sections, we will analyze the optimization of parameters for SVMs. A full discussion of the details of SVMs is beyond the scope of this paper; in this paper we are only considering the effectiveness of the parameters used to train an SVM. For further details on support vector machines, the reader is referred to [1], [2].

This paper focuses on the Cost-based Support Vector Classifier (C-SVC). This type of SVM requires the specification of two parameters: a cost parameter C, which is typically anywhere between 2^{-5} and 2^{20}, and a parameter γ which is typically anywhere between 2^{-20} and 2^{3} [3], [4].

The process by which we have chosen to discover these optimal—or as close to optimal as we can achieve under constraints—parameters is by iteratively guessing parameters, then performing 10-fold cross-validation of the SVM using those parameters. This paper discusses and compares search methods used to determine those parameters.

The naïve search method to find the near-optimal parameters is called the *grid search* [3], which tries evenly-spaced combinations of the SVM parameters [4]. For instance, if performing a coarse search of the region between 2^{-5} and 2^{20} for the cost parameter, we can choose to try every cost parameter 2^n for $n = -5, -3, -1, \ldots, 21$. For each of these cost parameters, we try every γ at the value 2^n for $n = -20, -18, -16, \ldots, 0, 4$. However, this search involves 156 different parameter combination choices, which requires 156 SVM training runs. A more granular seach would require an even more computationally intensive effort.

III. Optimization Techniques

A variety of search methods have been applied to the SVM model selection problem, including the grid search method described in Section II as well as metaheuristic

978-1-4244-7365-6/10/$26.00

techniques such as genetic algorithms [5], Gaussian process-based optimization [6], ant colony optimization [7], and particle swarm optimization [8], [9]. In this section, we will focus on the PSO and ACO techniques, as well as implementation issues in applying them to the SVM model selection problem.

A. Particle Swarm Optimization

The PSO algorithm draws its characteristics from the research areas of swarm intelligence and evolutionary computation. PSO is based on the metaphor of a swarm of *particles* "flying" through the fitness landscape to find the optimum values of a fitness function. Individual particles communicate their fitness values to the entire swarm, thus guiding the swarm to promising regions of the search space. For a full discussion of the PSO algorithm and its many variations, see [10], [11].

As the PSO algorithm is originally defined over a continuous multivariate domain, it is readily applicable to the SVM model selection problem. Descriptions of existing applications of PSO to the SVM model selection problem can be found in [8], [9].

B. Ant Colony Optimization

Ant Colony Optimization [12] is an optimization technique typically applied to combinatorial optimization problems with much success. Like PSO, it draws its inspiration from the area of swarm intelligence and specifically from the observation of ant colonies in nature. As ants explore their environment and discover food sources, they leave pheromone trails which other ants follow. Over time, these trails fade unless reinforced by additional ants which follow them and find food. The ACO technique follows the same metaphor, where the path can be considered to be a set of values for the variables in the search space, and reinforcement of these paths is based on the results of the objective function being optimized.

The typical formulation of ACO is based on discrete variables, for which the ant colony metaphor is an appropriate fit. This is the approach used in existing applications of ACO to the SVM model selection problem, as described in [7]. In order for this to work, however, the naturally continuous search space of the SVM model selection problem must be discretized, which limits the granularity at which promising areas of the search space can be examined.

To avoid this type of discretization step, many researchers have established methods of applying the ACO metaphor to a continuous domain. The predominant method of doing this is by extending the discrete probability mass function used to determine which path to take in the traditional ACO algorithm into a continuous Probability Density Function (PDF), such as is found in [13], which models each variable in the search space as its own mixture of univariate Gaussian PDFs.

C. Ant Colony Optimization in Continuous Domains

A more sophisticated extension of the ACO metaphor to the continuous space—and the basis for the implementation presented in this paper—was developed in [14]. This Aggregation Pheromone System (APS) treats the entire multivariate search space as a mixture of multivariate Gaussian PDFs.

The APS algorithm, as we have implemented it and applied it to the SVM model selection problem, will be referred to as APS-SVM in this paper. APS-SVM begins much like the traditional ACO algorithm, constructing paths for the ants by generating the initial set of values from a multivariate—in our case, bivariate—uniform distribution. Each of these ants is evaluated by supplying its specified SVM parameters to the SVM which undergoes cross-validation to determine classification accuracy.

The results of this cross-validation are used to rank the ants, and the rankings are used as weights in the creation of a mixture of multivariate Gaussian PDFs, where the highest-ranking ant is given the most influence on this mixture. This Gaussian mixture is then added to the overall pheromone density of the search space, itself a mixture of Gaussian and uniform distributions. During this pheromone update step, the influence of the mixtures of PDFs from earlier iterations of the algorithm are reduced in weight, thus incorporating the pheromone evaporation component of the ACO technique.

Three parameters must be specified which affect the way the APS-SVM algorithm works: ρ ($0 \leq \rho < 1$), controls the evaporation rate, and thus how much the Gaussian mixture is influenced by results from several iterations earlier in the process; α ($\alpha > 0$) controls how much the rank of an ant influences the weight of its corresponding part of the resultant Gaussian mixture; and β ($\beta > 0$) is a scaling factor used to control the exploratory tendencies of the algorithm by scaling the covariances among the variables in the Gaussian mixture. Higher vales of β tend to encourage exploitation of the search space, and lower values encourage exploration. More details of these parameters and the underlying mathematical formulation can be found in [14].

IV. Implementation Details

In the case of our SVM model selection problem, the two parameters C and γ create a two-dimensional search space to which we can apply our PSO and APS-SVM optimization techniques.

As both APS-SVM and PSO are iterative population-based optimization techniques, we can easily perform side-by-side comparisons of the two techniques. By using the same number of individuals and the same number of iterations for both algorithms, we can perform a more rigorous analysis of the comparative performance of the two algorithms. For both the PSO and the APS-SVM algorithm, we used eight individuals (particles or ants) over eight iterations.

We analyzed the results of the APS-SVM and PSO techniques in two ways: first, we compared the classification accuracy of parameterized SVMs using a constant number of training runs, which we also compared against the traditional grid search technique; and second, a comparison of how quickly each of the optimization techniques arrived at near-optimal parameters.

For the PSO algorithm, we used the *gbest* model of PSO [11] with an intertial coefficient of 0.75, a cognitive coefficient of 1.8, a social coefficient of 2, and a velocity clamping factor of 0.5. For the APS-SVM algorithm, we used a scaling factor (β) of 0.6, a rank influence factor (α) of 4, and a pheremone evaporation factor (ρ) of 0.9. These PSO and APS-SVM settings were found to encourage convergence toward a maximum in a limited number of iterations.

The SVM training and testing was performed using the LIBSVM software package [3]. The data used for our tests were retrieved from the UCI Machine Learning Repository [15]. The specific datasets used were a DNA splicing junction dataset, a German credit score dataset, an Italian wine characteristic dataset, a vehicle silhouette dataset, and the oft-used iris flower dataset.

V. Results

Table I shows the classification accuracy of SVMs where the parameters are discovered via grid search, PSO, and APS-SVM techniques. The grid search was run over a grid of 13 C values and 12 γ values, for a total of 156 SVM training runs. The APS-SVM and PSO algorithms were limited to 64 training runs: eight individuals (particles or ants) over eight iterations.

SVM Classifier Accuracy

	Grid*		PSO†		APS-SVM†	
	Avg	StDev	Avg	StDev	Avg	StDev
DNA	96.13%	0.14%	96.49%	0.12%	96.42%	0.12%
Credit	77.46%	0.15%	77.50%	0.28%	77.44%	0.17%
Wine	99.16%	0.39%	99.21%	0.29%	99.33%	0.24%
Vehicle	85.90%	0.36%	86.11%	1.79%	86.67%	0.18%
Iris	97.47%	0.28%	97.87%	0.28%	97.93%	0.42%

*Each grid search requires 156 SVM training runs
†PSO and APS-SVM searches limited to 64 total SVM training runs.
Averages and standard deviations computed over ten trials

Table I
Classification Accuracy Results

In order to mitigate the stochastic nature of the APS-SVM and PSO algorithms, we performed ten trials of each algorithm for the specified number of iterations, and provided the resulting average SVM classification accuracy and standard deviation over these ten trials.

Figure 1 offers an representative example of the rate at which the PSO and APS-SVM techniques approach the near-optimal SVM parameters. The "APS-SVM Best" and "PSO Best" lines represent the classification accuracy resulting from the best parameter choice by the algorithms during a particular iteration of the algorithm. The "APS-SVM Average" and "PSO Average" lines represent the average of the classification accuracy resulting from the parameters selected by each individual of the algorithm's population at a particular iteration. Again, much like in Table I, the values in Figure 1 represent averages over ten trials.

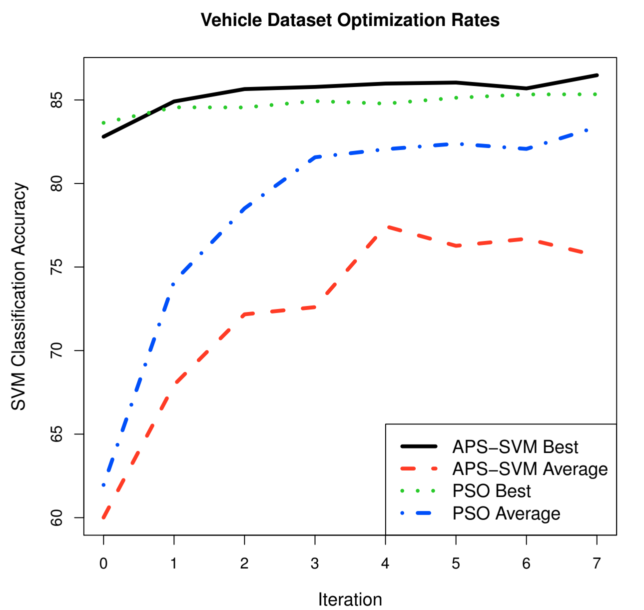

Figure 1. Comparison of APS-SVM and PSO performance, Vehicle Silhouette dataset

As can be seen in Table I, both PSO and APS-SVM demonstrate selection of parameters resulting in SVM classification accuracy roughly equivalent to the much more computationally-intensive grid search selection. They are able to find these parameters when restricted to 64 total training runs as opposed to the 156 training runs required by the grid search, a significant savings in the total amount of required training time.

The rates at which the PSO and APS-SVM algorithms approach near-optimal parameters, as exemplified in Figure 1, allow us to compare some of the operational characteristics of the PSO and APS-SVM techniques. By examining the "Best" and "Average" lines of the two algorithms, we can get a sense of the algorithms' underlying exploitative and exploratory abilities, respectively.

A quick inspection of the "Best" lines shows that both the PSO and APS-SVM algorithms are able to arrive at high-quality parameter selections at early iterations in the process. Thus, we conclude that both algorithms are able to adequately exploit the search space and provide us with a local near-optimum value in a reasonable number of iterations.

Analysis of the "Average" lines in Figure 1—and similar graphs developed for the other datasets but not included here—demonstrates more of a difference between the APS-SVM and PSO algorithms for the SVM model selection problem. The higher average classification accuracy for the PSO algorithm can be explained by the nature of our implementation of the PSO and APS-SVM algorithms; in particular, the *gbest* model of PSO tends to focus on exploiting one particular region as the cost of exploration. Meanwhile, the mixture of multivariate Gaussian distributions maintained by the APS-SVM algorithm tends to emphasize exploration of the search space.

It is important to note that much of the exploitative and exploratory abilities of the PSO and APS-SVM algorithms can be controlled by changing the implementation parameters used by these algorithms. However, we note that we arrived at our results using the same APS-SVM and PSO control parameters for each of the datasets attempted. Thus, while SVM classification accuracy is heavily dependent on the choice of SVM parameters customized for a specific dataset, the chosen APS-SVM or PSO control parameters were valid for every dataset we attempted. These implementation parameters, once discovered, should not need to be changed for new, previously unseen datasets.

VI. Conclusion and Future Work

We have demonstrated the application of two metaheuristic techniques—particle swarm optimization and ant colony optimization—to the SVM model selection problem, with promising results as compared to the more computationally-intensive naïve grid search technique often used for this problem. Furthermore, we have performed an assessment and side-by-side comparison of the results of our implementation of ACO and PSO algorithms to the SVM model selection problem.

While our implementation already provides a significant time savings over traditional grid search methods, additional testing and analysis may provide even greater improvement. By varying the parameters for both our APS-SVM and PSO implementations and examining the results, we may be able to increase the algorithms' abilities to explore and exploit the parameter search space of an SVM classifier. Further testing, comparison, and analysis may both provide us with a more finely-tuned SVM parameter selection technique and foster novel enhancements to our existing implementations.

References

[1] V. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.

[2] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[3] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," February 2009.

[4] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," May 2009.

[5] C. Chatelain, S. Adam, Y. Lecourtier, L. Heutte, and T. Paquet, "Multi-objective optimization for SVM model selection," in *ICDAR '07: Proceedings of the Ninth International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 427–431.

[6] H. Frohlich and A. Zell, "Efficient parameter selection for support vector machines in classification and regression via model-based global optimization," in *2005 IEEE International Joint Conference on Neural Networks, 2005. IJCNN'05. Proceedings*, vol. 3.

[7] J. Acevedo, S. Maldonado, S. Lafuente, H. Gomez, and P. Gil, *Ant Colony Optimization and Swarm Intelligence*. Springer Berlin / Heidelberg, 2006, ch. Model Selection for Support Vector Machines Using Ant Colony Optimization in an Electronic Nose Application, pp. 468–475.

[8] M. N. Kapp, R. Sabourin, and P. Maupin, "A PSO-based framework for dynamic SVM model selection," in *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*. New York, NY, USA: ACM, 2009, pp. 1227–1234.

[9] B. Feres de Souza, A. C. P. L. F. de Carvalho, R. Calvo, and R. P. Ishii, "Multiclass SVM model selection using particle swarm optimization," in *HIS '06: Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*. Washington, DC, USA: IEEE Computer Society, 2006, p. 31.

[10] J. Kennedy, R. Eberhart, and Y. Shi, *Swarm Intelligence*. Morgan Kaufmann, 2001.

[11] F. van den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, University of Pretoria, 2001.

[12] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, vol. 26, no. 1, pp. 29–41, 1996.

[13] K. Socha, *Ant Colony Optimization and Swarm Intelligence*. Springer Berlin / Heidelberg, 2004, ch. ACO For Continuous and Mixed-Variables Optimization, pp. 25–36.

[14] S. Tsutsui, "An enhanced aggregation pheromone system for real-parameter optimization in the aco metaphor," in *ANTS Workshop*, ser. Lecture Notes in Computer Science, M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, Eds., vol. 4150. Springer, 2006, pp. 60–71.

[15] A. Asuncion and D. Newman, "UCI machine learning repository," 2007. [Online]. Available: http://www.ics.uci.edu/∼mlearn/MLRepository.html