Hybrid SVM-Ant Colony Optimization for Identification of Protein Functions

Jayaraman VK[1], Ankur Gupta[2], Diwakar Patil[3], Prashant Shingade[4], Joydeep Mitra[5]
[1,2,3,4,5] National Chemical Laboratory, Pune, India
e-mail: vk.jayaraman@ncl.res.in

Abstract

Identification of functions of protein sequences requires employment of sophisticated classification algorithms. Support Vector Machines(SVM), rigorously based on statistical learning theory, is once such classification algorithm exhibiting superior performance. The classifier performance depends upon the ability to provide the most informative set of input features. Classification of data having high dimensionality is usually performed in conjunction with an appropriate feature selection method. We propose an Ant Colony Optimization (ACO)/SVM based hybrid filter-wrapper search technique, for simultaneously extraction of informative features and classification. We evaluate the performance of our algorithm with some important protein function identification problems.

Key words: Protein Functions, Ant colony, SVM-Hybrid.

I. INTRODUCTION

Support Vector Machines has recently been employed for solving several problems in bioinformatics and computational biology. For nonlinearly separable binary classification problems, SVM employs a linear hyperplane in a higher dimensional feature space. To deal with the problem of computational intractability, appropriate kernel functions are defined. With the help of kernel functions the computations can be carried out in the input space itself.

Ant colony optimization is a metaheuristic optimization methodology originally proposed by Dorigo et al. for solving combinatorial optimization problems [1,2]. ACO mimics the co-operative search behavior of real life ants for arriving at the optimal configuration. ACO has been employed for solving a variety of optimization problems in diverse fields.

Feature extraction is relevant and important for accurate classification performance. Irrelevant features not correlating to classification, if present, can interfere with the process of classification. Hence it is important to remove the redundant data to increase the performance of the learning system. The existing feature extraction procedures can be broadly categorized into filter and wrapper methods [3]. Filter methods generally use some heuristic information like statistical correlations to rank the features a priori (or feature subset), and are independent of the classification algorithm [4]. Wrapper methods on the other hand evaluate the quality of a feature subset by employing various learning classifiers. The learning schemes incorporate methods like sequential forward or backward searches, beam search, Genetic algorithms, Ant colony optimization, greedy variants of hill climbers, best first search etc. [5].

In this work we propose a hybrid wrapper-filter methodology for feature selection. ACO and SVM synergistically operate and simultaneously carry out the task of feature selection and identification of protein functions.

II. ALGORITHM DESCRIPTIONS

A. Support Vector Machines

Support Vector Machines (SVM) is rigorously based on Vapnik's statistical learning theory [6,7] and has been employed in several applications in different fields including bioinformatics. Given a set of sequences whose class labels are known, SVM builds a model, which can be employed for finding class labels of unknown sequences. SVM is able to obtain excellent accuracy in both training and testing stages and hence possesses the attractive feature of excellent generalization capabilities. Further, unlike many other algorithms it has the ability to converge to a single globally optimal solution. For a linearly separable training data the binary SVM builds an optimal hyperplane separating the two classes in the data. Such an optimal hyperplane maximizes the distance between itself and the nearest data points of each class. For nonlinearly separable problems, SVM first transforms the input data into a higher dimensional feature space and then constructs a linear hyperplane in the feature space. To deal with the computational problems arising due to high dimensionality of the feature space, an equivalent kernel function is defined so that the computations can be performed in the input space itself.

As SVM methodology is well established, we provide here, in brief, the steps involved in the SVM binary classification: 1) Maximize the margin of the linear hyper plane separating the data belonging to two classes by minimizing the norm of the weight vectors. This facilitates simultaneous optimization of training and test accuracy. 2) Transform the input data into a higher dimensional feature space. 3) Solve the computational problem by defining an appropriate kernel in the input space in place of the dot product in the high dimensional feature space Solve the dual formulation of the convex quadratic programming problem to obtain the unique global solution for the classifier. Employ the trained classifier model to test unseen sequences.

For multiclass classification it is customary to use one against all or one against one methodology to convert the multiclass problem into a series of binary classification tasks. For some problems such classifiers create imbalance in the classes with the data in one class far outnumbering the examples in the other class. Under these circumstances it is possible to give different weights to examples in different classes and subsequently employing a weighted SVM classifier.

Detailed discussions on SVM classification can be found in [6]. Hence, we provide here only the details of final form of objective function, decision function and the type of kernel function, which is employed in present study.

Let $x_i \in \mathbb{R}$, i = 1, 2...N be input training feature and y_i Є {+1,-1} be their corresponding target class label. Let N be the total number of input vectors.

The SVM-based classification is dependent on the sign of decision function f(x). It can be calculated as

$$f(x)=\sum_{i=1}^{nSV} y_i \alpha_i K(x_i,x_j)+b \qquad (1)$$

Where, nSV is the number of support vectors and b is the bias term. These are nothing but input data points having non-zero positive values of Lagrange multipliers (i). These can be obtained by solving a quadratic optimization problem,

$$\max_{\alpha}\left[\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j)\right] \qquad (2)$$

Subject to
0= $_i$= C i=1,2,.....,N

$$\sum_{i=1}^{N}\alpha_i y_i=0$$

Where *C* is a regularization parameter and controls tradeoff between the SVM complexity and the number of allowable errors in training error

K (xi, xj) denotes kernel function. Radial basis, polynomial and exponential kernel function are quite popular. In present study, simulations were performed using RBF function, defined by

$$K(x_i,x_j)=\exp\left(-\gamma\left\|x_i-x_j\right\|^2\right) \qquad (3)$$

Where γ is the RBF kernel parameter. It would be appropriate to mention here that several domain knowledge based kernel functions are available in bioinformatics and the researchers have a wide variety of these kernel functions to choose with a view to select the best for optimal classification performance.

In the RBF kernel, there are two free parameters, viz. kernel width parameter, sigma (σ) and regularization parameter, C, which have to be estimated to obtain best SVM performance. There are different measures by which SVM performance is assessed, cross validation and leave-one-out error estimates being the most commonly used ones. In the cross validation estimation the data is split randomly into k folds. The classification is performed k times, with one fold kept aside each time. The left out folds are employed to test the classification performance and the average of the test errors are estimated with each set of kernel parameters. Leave-one-out error estimate, on the other hand, requires the training algorithm to run as many times as the size of the training data since each element of the training data is tested on the classifier built with the Leave-one-out estimate is a good estimate of generalization error, but computationally it is very intensive and costlier than other estimates.

B. Ant Colony Optimization

Ant Colony Optimization is a meta-heuristic technique originally proposed by Dorigo et al. [1] for solving a class of combinatorial optimization problems. ACO mimics the real life cooperative search behavior of real life ants. These ants, on their sojourn from the nest to the food source and back, deposit a chemical known as pheromone. Further, these also get attracted to pheromone rich trails. It is easy to visualize that by employing the pheromone guided auto catalytic search they can quickly establish the optimal route. ACO, inspired by the pheromone-mediated search of real life ants, employs software ants for obtaining optimal solutions to different types of practical problems. ACO has been successfully applied to the combinatorial optimization problems like TSP, sequential ordering, scheduling, and process optimization [8]. It has recently been used in the context of feature selection [3,9,10,11].

C. Hybrid SVM-ACO algorithm for feature selection.

Let $S=\{f_1,f_2,f_3,\ldots,f_N\}$ be the complete set of N input features, and is a subset of most informative features to be selected , where . The feature selection algorithm is similar to the Traveling Salesman problem (TSP). In the TSP Problem the links between each cities are initially deposited with some random amounts of pheromone. Employing the process of exploration and exploitation the software ants are deputed to conduct tours visiting each city once. After the completion of the tours the ant conducting the shortest tour is allowed to increase the pheromone concentration on the links. The software ants conduct several such generations of tours and finally the globally shortest tour is identified. In the feature subset selection problem the features represent the cities algorithm is initiated by initializing the number of ants, number of generations (iterations), algorithm parameters and pheromone concentrations at all links. Unlike problems solved earlier by ant colony optimization, like the traveling salesman problem (TSP), feature selection by ant colony involves only a partial tour corresponding to the subset of features *s*. Considering each feature representing a state, the paths of the software ants consist of a subset of features. At any iteration level, every link in the path has a pheromone value connecting two states (features *i* and *j*), which are represented by . Every feature also has a heuristic function value associated with it, which is a measure of quality of an individual feature. In the TSP, the heuristic function was the distance matrix. In the feature selection problem we can employ the ranking of the features by a filter method like information gain can be employed as the problem heuristic. The pheromone concentrations act as an indirect means of communication between the ants, which is iteratively updated as the ants build their solutions. Thus in a way, this represents the knowledge gathered by the ants over time. The heuristic function incorporates prior knowledge about the relative importance of individual features. Incorporating the heuristic function allows us to bias the algorithm so as to selectively favor the better features, thereby improving their chances to get selected in the best feature subset.

1) *State transition rules:* Following this, a software ant currently residing on feature i, may select the next feature j employing one of the two following processes: exploitation or exploration. The choice between exploitation or exploration depends on the algorithm parameter q_0, the value of which is decided a priori. q_0 represents the probability of selecting exploitation at every generation. Thus, for an ant which is on feature i, the feature j which it will travel to will depend on this choice. To make this choice, a random number, *r* was generated in the range of 0 to 1. If *r* q_0, exploitation is selected, otherwise exploration is selected. If the process of exploitation was selected, the feature, j, with the highest quality value corresponding to the product of the pheromone concentration on the link connecting i and j (*(f_{ij})*) and the heuristic function for the node j (*(f_j)*) is selected. Exploration, on the other hand involves the selection of the next feature j with a probability proportional to the relative quality of the feature to the subset of features not selected yet.

$$f = \begin{cases} \max\left\{\tau(f_{ij})\eta(f_j)^{\beta}\right\} & if\ (r < q_0) \quad Exploitation \\ \dfrac{\tau(f_{ij})\eta(f_j)^{\beta}}{\sum \tau(f_{ij})\eta(f_j)^{\beta}} & otherwise \quad Biased\ Exploration \end{cases} \tag{4}$$

The relative importance of the heuristic function over the pheromone concentration is pre-determined using the parameter . A value of *=1* would mean that both the pheromone value and the heuristic function are given equal importance, whereas a value of *=0* will not consider the heuristic function for selection of a gene, thus giving equal chance for every gene to get selected.

As already mentioned, the state transition rules select the features by maximizing the product of the heuristic function, which represents the prior knowledge about the search space, and the pheromone values, which correspond to the knowledge gained by the ants by traversing the search space over time. Thus, the product of the two, in a way can be said to represent the total available dynamic knowledge about the features. The exploitation mode forces the ants to select the features with maximum product of pheromone and heuristic function values, thus exploiting the available knowledge. On the other hand, in the biased exploration mode, the ants probabilistically select those features, which were not selected by it before, thus exploring new paths. A feature with a higher value of pheromone and heuristic values will have a greater probability of getting selected. Thus the value of the exploitation probability factor, q_0 will decide the predominant mode of selection of paths for the ants. A high value of q_0 will ensure the selection of known "good" features, while keeping the exploration ability of the ants less. A low value of q_0, on the other hand, will force the ants to keep exploring newer paths. The value of $q_0$is user-defined which can be varied depending on the problem at hand.

Every ant uses the state transition rules to select a specified number of features for its solution subset (except for the initial condition when the ants build their solutions by randomly selecting features from the complete set of features). After all the ants have built their solutions, these are evaluated and the pheromone values of the features are updated accordingly using the Global and Local Updating Rules. This is repeated for a specified number of generations, to obtain a best feature subset of that subset size. After this the number of features to be selected by every ant is decremented, and the whole process is repeated till the number of features reaches a pre-specified minimum limit. The best solution obtained among all the subset sizes is recorded. We now explain the steps involved in detail.

In order to evaluate the quality of the feature subsets obtained from the ACO algorithm, we use the most commonly used method, Leave-One-Out Cross Validation (LOOCV) accuracy. In this method, one example is taken out from the calibration dataset every iteration, and the SVM classifier is trained on the remaining data. This model is then tested on the left out sample. This is repeated for every test sample, and the average accuracy over all the iterations is found out. This is called the LOOCV accuracy. Thus, the ant obtaining the highest LOOCV accuracy is called the winner ant. Consequently, the pheromone values are updated differently for the winner ant and the other ants.

2) *Global transition rules:* After all the feature subsets are evaluated, the pheromone value has to be updated so as to deposit more amount of pheromone on the more desirable paths. This will increase the desirability of those features, which produce a higher accuracy. For this a winner ant among all the ants is identified and a global updating rule is applied to only those features, which belong to the winning ant's subset. This rule increments the pheromone value of these genes according to the relation,

$$\tau(f)' = (1-\kappa)\times\tau(f) + \kappa\times\sigma \qquad (5)$$

Where,is inversely proportional to the SVM cross-validation accuracy estimate of the globally best solution obtained for the current subset size expressed the range of 0 to 1, andis the pheromone decay parameter also in the range of [0,1]. Thus, this rule allows only the ant that finds the best gene subset, to deposit pheromone on the genes it has selected. Obviously, these features will become more attractive for the other ants to select in the next generation. Gradually, over generations the ants will learn to select "good" features and discard "bad" features.

3) *Local transition rules:* Pheromone values of the features not selected by the winner ant are updated using the local updating rule. This rule slightly decrements the value of those features, which were selected by the ants but did not win, while keeping the pheromone values of the unexplored features unaltered. The local updating rule is given by,

$$\tau(f)' = (1-\alpha)\times\tau(f) + \alpha\times\tau_0 \qquad (6)$$

Where, is called the local pheromone update strength parameter which lies in the range of [0,1]. This rule ensures that the desirability of the features that have been identified as irrelevant is decreased thereby reducing their chances of selection.

III. DATASETS

Three different datasets were used for our analysis, taken from the literature. The first two datasets involve protein sequence annotation, with identification of candidate subunit vaccines and prediction of disorder from protein sequences, respectively. We have also used the same algorithm for gene selection from gene expression data, and henceforth included a third dataset, that involving gene expression profiles of breast cancer patients. The datasets are described in the following section.

A. Allergen Dataset

The discovery of subunit vaccines is a priority problem in clinical research, in the era of the second-generation vaccines. Experimental approaches that currently exist are highly labor and time-consuming. This necessitates efficient informatics based approaches for identifying subunit vaccines from proteins sequences. Subunit vaccines contain one or more pure or semi-pure antigens. In order to develop subunit vaccines, it is critical to identify the individual components out of a myriad of proteins and glycoproteins of the pathogen that are involved in inducing protection. The idea is to discriminate between proteins that contain antigenic properties within their sequence from those that do not.

The set of proteins in this dataset have been borrowed from Doytchinova and Flower [12], who used a set of molecular descriptors, which were defined by various physicochemical properties of the amino acids. The methodology used for the discrimination was PLS discriminant analysis.

The dataset contains 150 sequences for training and 75 for testing. Each protein sequence was represented in the feature space in terms of their amino acid and Dipeptide frequencies.

B. Disordered proteins

Disordered proteins are those, which contain stretches of amino acids that fail to fold on their own. The *in silico* prediction of protein disorder has gained importance in recent times, since these proteins are difficult to crystallize, which in turn makes it difficult to determine their structures experimentally.

The set of proteins in this dataset have been used previously in a work that combines a non-linear signal analysis technique, recurrence quantification analysis with SVMs [13]. The set for disordered proteins was taken from the DISPROT database. Proteins with a continuous stretch of disordered residues of length 40 or more were taken. The set of ordered protein was borrowed from Dunker et al. [14] who grouped the PDB proteins into families having >25% identity to create the PDB_SELECT_25 (PDB_S_25) database. All proteins that contain missing coordinates for backbone residues were removed from PDB_S_25 to create the ORDERED_PDB_SELECT_25 (O_PDB_S_25) dataset.

The dataset consisted of 350 disordered and 600 ordered proteins, from which, 30% sequences were taken out for testing. Each sequence was represented in terms of twenty physicochemical properties taken from AAIndex, including Kyte-Doolittle hydrophobicity scale, secondary structure propensity scales, surface accessibility and charge. Barring charge, for which the net charge was calculated for each sequence, all other properties were averaged out over the entire sequence length.

C. Breast cancer gene expression dataset

The Breast cancer dataset [15] consists of gene expression levels of 7129 genes in a set of 49 breast tumor samples, classified according to their estrogen receptor (ER) status. There are 25 ER positive samples, with 24 ER negative samples. The complete dataset is available on the website http://data.cgt.duke.edu/west.php.

Using standard preprocessing methodology, the numbers of genes were reduced to 5146. This data was then log-transformed using base 10 and normalized for zero mean and unit standard deviation.

IV. RESULTS AND DISCUSSION

Table 1 shows the various ACO parameters chosen for the feature selection process. The initial number of features was taken, with a decrement of 1-10 feature per iteration until the number of features reaches 2. The number of ants chosen was between 20-80. Obviously, a higher number of ants would use up a larger amount of computational time, however, a small number of ants may not be sufficient to effectively explore all the potential states. Similarly, a large number of generations will increase the computational time manifold, while a lesser number of generations may not be able to obtain the best solution in the given subset size. These values have been chosen by trial and error. We have found that 40 generations per subset size is sufficient for the given problems.

Table 1. Ant Colony Optimization Parameters

Parameters	Value
Number of Ants (r)	20-40
Number of Generations	20-100
Exploitation Probability Factor (q_0)	0.5-0.8
Pheromone Importance Factor (β)	0.5-0.9
Pheromone Decay Parameter (κ)	0.5-0.9
Pheromone Update Strength (α)	0.1-0.3

The exploitation probability factor (q_0) decides the relative importance of the exploitation and the biased exploration mode. A high value of q_0 has been chosen so as to favor exploitation mode, but allowing the ants to explore new features simultaneously. The pheromone decay parameter () and the pheromone update strength parameter () have been taken in the range 0.5-0.9 and 0.1-0.3 respectively. The value of the importance parameter of pheromone relative to heuristic function () has been varied between 0.5-0.9. It should be noted that the heuristic function values were scaled from 0 to 1, thus a value of *>1* will give less importance to the heuristic function than the pheromone value.

The 10-fold cross-validation accuracy for classification had improved significantly for all the three problems, on using SVM with the selected features. A grid-search was performed on the selected feature subset to determine the best performance with the selected feature set. Although several combinations of feature subsets giving good performance were obtained for each problem, only the smallest subsets with the best cross-validation accuracies have been reported here.

In the allergen dataset, the 100 dipeptides that were selected showed an enhanced performance of 81.6% 10-fold cross validation accuracy; improved from the 76.7% obtained using the full feature set of 400 dipeptides.

The disorder dataset, consisting of the 20 physicochemical properties showed a cross-validation accuracy of 68.23%. Following rigorous feature selection with our algorithm, an accuracy of 79.01% was obtained with only two features, viz. GRAVY and net charge.

The breast cancer dataset is a gene expression dataset with 5146 genes, of which several subsets are possible that give good accuracy. To validate our algorithm, we compared our results with equal number of features selected using InfoGain ranking only. Evaluation was done using leave-one-out cross validation (LOO-CV) using SVM using subsets selected using both the algorithms. The top 20 most informative genes from the InfoGain ranking gave a LOO-CV accuracy of 91.83%. The 20 features selected by our algorithm showed 100% LOO-CV accuracy.

High dimensionality of the input features for the classification of protein sequences for function prediction poses a formidable challenge in bioinformatics. In this work, we proposed an Ant Colony Optimization based hybrid filter-wrapper multivariate feature selection technique, which employs the use of feature-ranking procedures along with pheromone mediated search capabilities to successfully select gene subsets that are highly relevant for the classification purpose. The algorithm might bring out biologically relevant feature subsets, which can provide a deeper insight into the problem. In this work, only one classification method (Support Vector Classification) has been employed here. It will be interesting to study the use of other classification algorithms along with different feature-ranking procedures using the same ACO algorithm.

ACKNOWLEDGMENT

V.K.J acknowledges financial support from The Dept. of Biotechnology, Govt. of India, New Delhi.

REFERENCES

[1] M. Dorigo, L. M. Gambardella, 1997, "Ant colony system: a cooperative learning approach to the travelingsalesman problem" *BioSystems*, vol. 43, pp. 73-81,

[2] M. Dorigo, G. D. Caro, L. M. Gambardella, 1999, "Ant Algorithms for Discrete Optimization", *Artificial Life*, vol. 5(2), pp. 137-172,

[3] R. Kohavi, 1995, "Wrappers for performance enhancement and oblivious decision graphs", *Ph.D. Thesis*, Stanford University,

[4] I. Inza, P. Larranaga, R. Blanco. A.J. Cerrolaza. 2004, "Filter versus wrapper gene selection approaches in DNA microarray domains", Artificial Intelligence in Medicine, Data Mining in Genomics and Proteomics, vol. 31(2) , pp. 91103,

[5] R. Kumar, V.K. Jayaraman, B.D. Kulkarni, 2005, "An SVM classifier incorporating simultaneous noise reduction and feature selection: illustrative case examples", *Pattern Recognition*, vol. 38(1), pp. 41-49,

[6] V. Vapnik, 1995, in *The Nature of Statistical Learning Theory*, 1st Ed. NY Springer,

[7] C. J. C. Burges, 1998, "A Tutorial on Support Vector Machines for Pattern Recognition" *Data Min Knowl Disc.*, vol. 2, pp. 121167.

[8] A. Liaw, 2003, *http:www.stat.ncsu.edu/seminar/aliaw.html*, extracted on Sep 10.

[9] J. Doak, 1992, University of California at Davis, *CSE Technical Report*, pp. 92-18.

[10] D.E. Goldberg, 1989, in *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Pub. Co.,

[11] H. Almauallium, T.G. Dietterich 1991, "Learning with many irrelevant features", *Proceedings of Ninth National Conference on Artificial Intelligence* , 2, AAAI Press, Anaheim, pp. 547552, , CA.

[12] I.A. Doytchinova, D.R. Flower, 2007, "Identifying candidate subunit vaccines using an alignment independent based on principal amino acid properties", *Vaccine* vol. 25 (5), pp. 856-866.

[13] J. Mitra, P.K. Mundra, B.D. Kulkarni, V.K. Jayaraman, 2007, "Using recurrence quantification analysis descriptors for protein sequence classification with support vector machines", *J Biomol Struct Dyn*, vol 25(3), pp. 289-297.

[14] P. Romero, Z. Obradovic, X. Li, E. C. Garner, C. J. Brown, A. K. Dunker. 2001, "Sequence complexity of disordered proteins.", *Proteins*, vol. *42*, pp. 38-48.

[15] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.A. Olson, J.R. Marks, J.R. Nevins, 2001, "Predicting the clinical status of human breast cancer by using gene expression profiles", *Proc. Natl. Acad. Sci.*, vol. 98(20), pp. 11462-11467.