Modified Multi-Swarm PSO to Resolve Multi-dimensional Optimization Problems in Data Mining

Chandra Pushpanjali Patel
Dept. of CSE,
Inderprastha Engineering College

Suchismita Mishra
Dept. of CSE,
Inderprastha Engineering College

ABSTRACT

Particle Swarm Optimization (PSO) is one of the popular bio inspired artificial algorithm which is based on the communal behavioural aspects linked with bird gathering to resolve various optimization problems. In this research work, we propose a set of rules by formulating the mechanism for survival of the fittest, feigns the race attitude among the swarms. Based on this spect of swarms, we suggested a modified Multi-Swarm PSO (MSPSO) to solve multi-dimensional optimization problems. Further we propose an Improved Feature Selection (IFS) method by integrating MSPSO, Support Vector Machines (SVM). The IFS method aims to achieve higher generalization capability through performing kernel parameter optimization and feature selection simultaneously. The performance of the proposed method is compared with that of the standard PSO based methods on 4 benchmark datasets, taken from UCI machine learning.

Keywords

Particle swarm optimization, feature selection, data mining, support vector machines

1. INTRODUCTION

Advancement in the field of bioinformatics has facilitated many researchers in analyzing the data and understanding the structural, comparative and functional properties. Some of the enhancements being analysis of genomes and proteins, identifying metabolic and signaling pathways which define the gene to gene relationships, development of microarray chip and conducting microarray experiments to measure the gene expression levels. The availability ofthe data on public websites and repositories made it easier to carry out the research. One ofthe major advancement made in the field of bioinformatics is the emergence of microarray technology. Microarray technology facilitates in determining the expression values of several genes simultaneously. The gene expression data is used for various analyses to understand the biological significance of the species or the tissue from which the genes were extracted for the experiment. This study focuses on analysis and calculation of distance measure and margin of a support vector machine classifier for microarray dataset. It also deals with studying the effect of margin value on the classification accuracy and relation between them.

The characteristic features and behavior of a biological species largely depends on the genes and the proteins present in it. Proteins obtained from the genes vary depending upon the gene expression levels. Hence analyzing the expression levels of genes under various conditions will help us in identifying the reason behind abnormalities in diseased 3 species in addition toidentifying the genes responsible for the abnormality. Microarray technology is used to study and record the gene expressions of thousands of genes simultaneously.

LITERATURE REVIEW

Liu et al [1], has proposed four rules are designed by introducing the mechanism for survival of the fittest, which simulates the competition among the swarms. A modified Multi-Swarm Particle Swarm optimization (MSPSO) is designed to solve discrete problems, which consists of a number of sub swarms & a multi swarm scheduler that can monitor & control each sub swarm using the rule.

Jiuzhong et al [2], has proposed the evolutionary algorithm is an effective tool to solve optimization problems. A Multi-Swarm Self-Adaptive and Cooperative Particle Swarm optimization (MSCPSO) decomposed into four sub- swarms, which exchange information among themselves to evaluate overall fitness as the basis of the fitness adaptive equation. The MSCPSO is simple and easy to implement like the original PSO.

Ying Li, Jiaxi Liang, Jie Hu [5], has proposed A Multi-swarm Cooperative Hybrid Particle Swarm Optimizer, Abstract—Cooperative approaches have proved to be very useful in evolutionary computation. It involves a collection of two sub-swarms that interact by exchanging information to solve a problem. The two swarms execute IPSO (improved PSO) independently to maintain the diversity of populations, while introducing extremaloptimization (EO) to IPSO after running fixed generations to enhance the exploitation. States of the particles are updated based on global best particle that has been searched by all the particle swarms. Synchronous learning strategy and random mutation scheme are both absorbed in our approach. Simulations on a suite of benchmark functions demonstrate that this method can improve the performance of the original PSO significantly.

Yuanning Liu et al [9], formulate four rules by introducing the mechanism for survival of the fittest, which simulates the competition among the swarms. Based on the mechanism, paper proposes a modified Multi-Swarm PSO (MSPSO) to solve discrete problems, which consists of a number of sub-swarms and a multi-swarm scheduler that can monitor and control each sub-swarm using the rules. To further settle the feature selection problems, propose an Improved Feature Selection (IFS) method by integrating MSPSO, Support Vector Machines (SVM) with F-score method. The IFS method aims to achieve higher generalization capability through performing kernel parameter optimization and feature selection simultaneously. The performance of the proposed method is compared with that of the standard PSO based, Genetic

Algorithm (GA) based and the grid search based methods on 10benchmark datasets, taken from UCI machine learning

results and statistical analysis show that the proposed IFS method performs significantly better than the other three methods in terms of prediction accuracy with smaller subset of features.

Xiangyang Wang etal [11], Many feature subset selection algorithms have been proposed and discussed for years. However, the problem of finding the optimal feature subset from full datastill remains to be a difficult problem.

In this paper, we propose novel methods to find the relevant feature subset by using biologically-inspired algorithms such as Genetic Algorithm and Particle Swarm Optimization. Our feature selection methods based on the biologically-inspired algorithms produced better performance than other methods in terms of the classification accuracy and the feature relevance.

2. EXPERIMENTAL SET-UP

Microarray is a chip on which biological substrates are bound to the probes present on the silicon chip or a glass slide. The biological substrates can be DNAs, proteins molecules or carbohydrates that decide the type of microarray chip.

To standardize the analysis of microarrays,a commonly accepted form of the microarray data structure has evolved. The data structure is an (M×N) 2-D matrix of gene expressions of (M) genes for (N) samples. In some literature it is defined as the transpose of this definition, i.e. (M×N).

This data structure is usually referred to as (**X**):

$$x1(1) \quad \cdots \quad x1(N)$$

and Stat Log databases. The numerical

$$\mathrm{X(t)} = (\quad) = \quad (x1(t) \ldots$$
$$. \quad (t)) \ , \mathrm{t} = 1{,}2\ldots\ldots\mathrm{N}. \ (1)$$
$$(1) \quad \cdots \quad x(N)$$

$$\mathrm{X(t)} = (\quad) = \quad (x1(t) \ldots$$
$$. \quad (t)) \ , \mathrm{t} = 1{,}2\ldots\ldots\mathrm{N}. \ (1)$$
$$(1) \quad \cdots \quad x(N)$$

(Eq.1) shows the mathematical definition of the microarray. The expression x_i (t) denotes the value of the gene (i) for the sample (t). In most of the times this set of data is associated with groups' labels vector y(t) which maps each sample's gene expression vector to a group label .Usually the labels are discrete numeric values that represent different groups. For example if some of the samples belong to cancer tumors y(t) and the others to normal tumors then might be either 1 or 0 denoting a cancer sample or a normal sample respectively. (Eq.2) shows the mathematical mapping x (t) of y (t).

$$\mathrm{X(t)} = [x(1) \quad x(2) \quad \ldots\ldots.(N)] \ ,$$
$$\mathrm{Y(t)} = [\, y(1) \quad y(2) \quad \ldots y(N)]$$
$$(2)$$

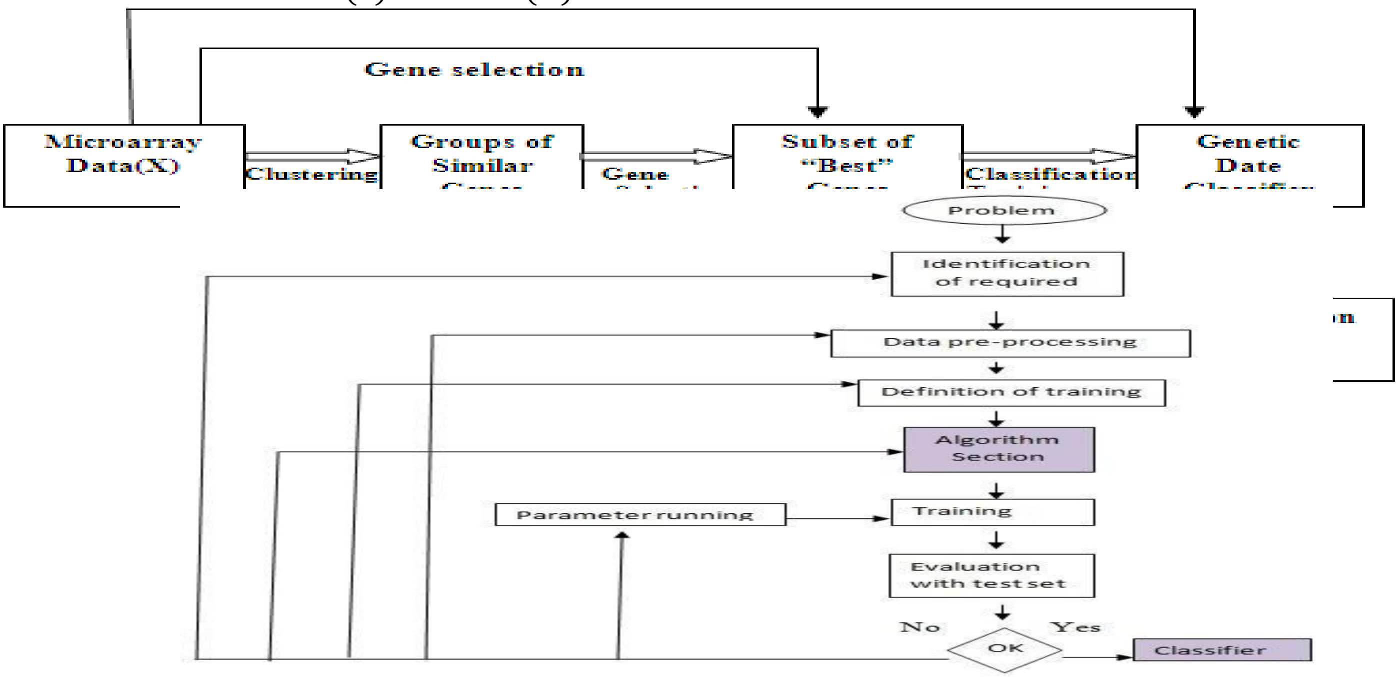

Fig. 1 Analytical model

Murthy (1998) provided an overview of work in decision trees and a sample of their usefulness to newcomers as well as practitioners in the field of machine learning. Thus, in this work, apart from a brief description of decision trees, we will refer to some more recent works than those in Murthy's article as well as few very important articles that were published earlier. Decision trees are trees that classify instances by sorting them based on feature values. Each node in a decision tree represents a feature in an instance to be classified,and each branch represents a value that the node can assume. Instances are classified starting at the root node and sorted based on their feature values. Figure 2 is an example of a decision tree

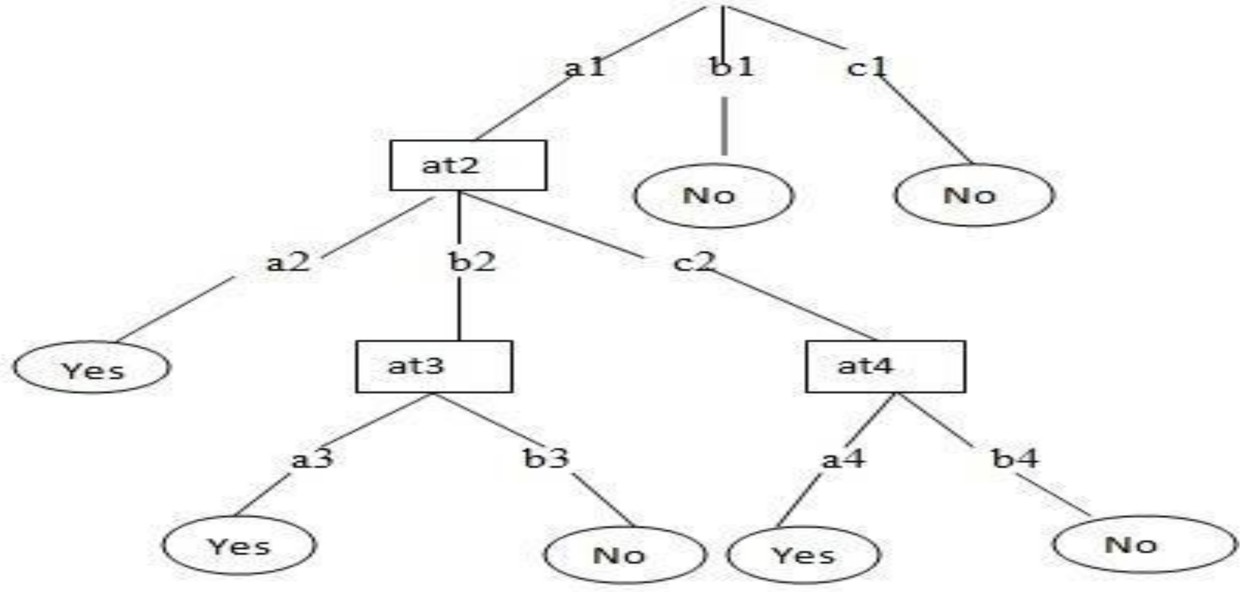

FIG3. Decision Tree

A decision tree, or any learned hypothesis *h*, is said to overfit training data if anotherhypothesis *h′* exists that has a larger error than *h* when tested on the training data, but a smaller error than *h* when tested on the entire dataset. The most straightforward way of tackling overfitting is to pre prune the decision tree by not allowing it to grow to its full size. Establishing a non-trivial termination criterion such as a threshold test for the feature quality metric can do that. Decision tree classifiers usually employ post-pruning techniques that evaluate the performance of decision trees, as they are pruned by using a validation set. Any node can be removed and assigned the most common class of the training instances that are sorted to it

Even though the divide-and-conquer algorithm is quick, efficiency can become important in tasks with hundreds of thousands of instances. The most time consuming aspect is sorting the instances on a numeric feature to find the best threshold *t*. This can be expedited if possible thresholds for a numeric feature are determined just once, effectively converting the feature to discrete intervals, or if the threshold is determined from a subset of the instances. Elomaa & Rousu (1999) stated that the use of binary discretization with C4.5 needs about the half training time of using C4.5 multisplitting.

SUPPORT VCTOR MACHINE

Support vector machine (SVM) is gaining popularity for its ability to classify noisy and high dimensional data. SVM is a statistical learning algorithm that classifies the samples using a subset of training samples called support vectors. The idea behind SVM classifier is that it creates a feature space using the attributes in the training data. It then tries to identify a decision boundary or a hyper-plane that separates the feature space into two halves where each half contains only the training data points belonging to a category.

LINEAR SVM

Given some training data $\mathcal{D}$, a set of *n* points of the form

$$\mathcal{D} = \left\{ (\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\} \right\}_{i=1}^{n} \tag{3}$$

(3)

Where the y_i is either 1 or −1, indicating the class to which the point belongs. Each is a *p*-dimensional real vector. We want to find the maximum-margin hyperplane that divides the points having $y_i = 1$from those having $y_i = -1$. Any hyperplane can be written as the set of points satisfying

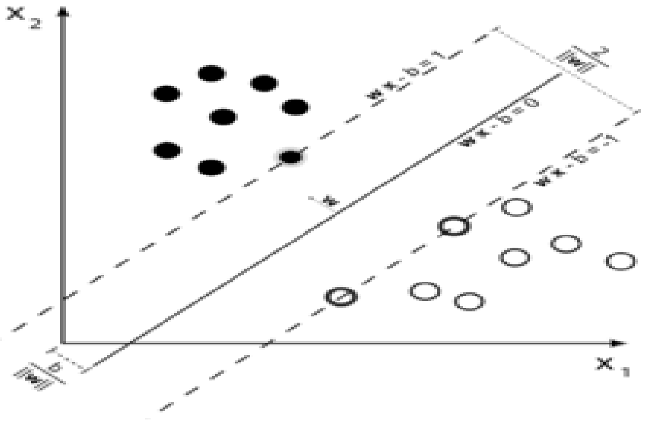

Fig. 4 Linear SVM

Maximum-margin hyperplane and margins for an SVM trained with samples from twoclasses. Samples on the margin are called the support vectors.

$$\mathbf{w} \cdot \mathbf{x} - b = 0, \qquad (4)$$

Where $\cdot$ denotes the dot product and $\mathbf{w}$ the normal vector to the hyperplane. The parameter $\frac{b}{\|\mathbf{w}\|}$ determines the offset of the hyperplane from the origin along the normal vector .

If the training data are linearly separable, we can select two hyperplanes in a way that they separate the data and there are no points between them, and then try to maximize their distance. The region bounded by them is called "the margin". These hyperplanes can be described by the equations

$$\mathbf{w} \cdot \mathbf{x} - b = 1 \qquad (5)$$

and

$$\mathbf{w} \cdot \mathbf{x} - b = -1. \qquad (6)$$

3. PROPOSED METHOD :

Step 1: Load the dataset. Initialize the size and position of each particles xij with velocity vijarbitrarily.
Update pbest (local best) and gbest (global best) of each particle. Go to Step 2.
Step 2: Identify parameters for each particle with lower and upper bounds of velocity, size,number of iterations. Go to Step 3.
Step 3: In each particle, if the current number of iteration < iteration numbers or gbest keepsno changes less than 45 number of iterations, then go to Step 4, otherwise go to Step 10.
Step 4: In each swarm, if current number of particle < particle size, go to Step 5, otherwise,go to Step 9.
Step 5: In each particle, acquire the gbest and pbest for each particle and keep updates itsposition and velocity. Go to Step 6.
Step 6: Restrict position and velocity of each individual. Go to Step 7.

Step 7: Each particle calculates its fitness Eq (4.5) and up-dates pbest and gbest.

$$fitness_i = \theta_a \times accuracy_i + \theta_b \times \left[\frac{\sum_{j=1}^{N_b} F(FS(i))}{\sum_{k=1}^{N_b} F(k)} \right]. \qquad (7)$$

In Eq. (4.5), θa is the weight for SVM classification accuracy rate, $accuracy_i$ is the classification accuracy rate for the selected features, θb the weight for the score of selected features, F(FS(i)) the function for calculating the score of the current features, and the total score of the selected features and all features respectively are

$$\sum_{k=1}^{N_b} F(k) \text{ and } \sum_{j=1}^{N_b} F(FS(i))$$

Step 8: current particle number = current particle number + 1. Go to Step 4.
Step 9: current iteration number = current iteration, number + 1. Go to Step 3.
Step 10: Execute multi-swarm

collection rule, and exit.

4. RESULT ANALYSIS:

Using the decision tree depicted in Figure 3.2.as an example, the instance {at1 = a1, at2 = b2, at3 = a3, at4 =b4} would sort to the nodes: at1, at2, and finally at3, which would classify the instance as being positive (represented by the values −Yes‖).The problem of constructing optimal binary decision trees is an NPcomplete problem and thus theoreticians have searched for efficient heuristics for constructing near-optimal decision trees. The feature that best divides the training data would be the root node of the tree. There are numerous methods for finding the feature that best divides the training data such as information gain (Hunt et al., 1966) and gini index (Breiman et al., 1984). While myopic measures estimate each attribute independently, ReliefF algorithm (Kononenko, 1994) estimates them in the context of other attributes. However, a majority of studies have concluded that there is no single best method (Murthy, 1998). Comparison of individual methods may still be important when deciding which metric should be used in a particular dataset. The same procedure is then repeated on each partition of the divided data, creating sub-trees until the training data is divided into subsets of the same class.
Decision trees can be translated into a set of rules by creating a separate rule for each path from the root to a leaf in the tree (Quinlan, 1993). However, rules can also be directly induced from training data using a variety of rule-based algorithms. Furnkranz (1999) provided an excellent overview of existing work in rule-based methods.

Classification rules represent each class by disjunctive normal form (DNF). A k-DNF expression is of the form: $(X_1 \wedge X_2 \wedge \ldots \wedge X_n) \vee (X_{n+1} \wedge X_{n+2} \wedge \ldots X_{2n}) \vee \ldots \vee (X_{(k-1)n}+1 \wedge X_{(k-1)n+2} \wedge \ldots \wedge X_{kn})$, where k is the number of disjunctions, n is the number of conjunctions in each disjunction, and X_n is defined over the alphabet $X_1, X_2, \ldots, X_j$ U $\sim X_1, \sim X_2, \ldots, \sim Xj$. The goal is to construct the smallest rule-set that is consistent with the training data. A large number of learned rules is usually a sign that the learning algorithm is attempting to −remember‖ the training set, instead of discovering the assumptions that govern it.

There are numerous other rule-based learning algorithms. Furnkranz (1999) referred to most of them. The PART algorithm infers rules by repeatedly generating partial decision trees,thus combining the two major paradigms for rule generation − creating rules from decision trees and the separate-and-conquer rule learning technique. Once a partial tree has been build,a single rule is extracted from it and for this reason the PART algorithm avoids post processing (Frank and Witten, 1998).

5. CONCLUSION

In this study, a novel multi-swarm MSPSO algorithm is proposed to solve discrete problem, an efficient objective function of which is designed by taking into consideration classificationaccuracy rate and F-score. In order to describe the competition among the swarms, we introduced the mechanism for survival of the fittest. To further settle the feature selection problem, we put forward the IFS approach, in which both the SVM parameter optimization and the feature selection are dynamically executed by MSPSO algorithm, then, SVM model performs the classification tasks using the optimal parameter values and the subset of features.

6. REFERENCES

[1] Liu et al "An Improved Particle Swarm Optimization for Feature Selection ", Journal of Bionic Engineering, pp.191- 200, May 2011.

[2] Jiuzhong et al "A Multi-Swarm Self-Adaptive and Cooperative Particle Swarm Optimization" , Engineering Applications of Artificial Intelligence, pp.958-967,May 2011.

[3] Tinghua Wang Et al, ―Feature selection for SVM via optimization of kernel polarization with Gaussian ARD kernels‖ , Journal Expert Systems with Applications: An International Journal archive Volume 37 Issue 9, September, 2010 pp 6663-6668.

[4] Ching-Yi Cheo etal, ‖proposed Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis‖, lntematianai Conference on Networking, SensingControl.Proceedings of the 2004 IEEE Taiwi. Taiwan. March 21-23, 2004

[5] Ying Li, Jiaxi Liang, Jie Hu ,―Ying Li, Jiaxi Liang, Jie Hu, has proposed A Multi-swarm Cooperative Hybrid Particle Swarm Optimizer ―.2010 Sixth International Conference on Natural Computation (ICNC 2010).

[6] Chung-Jui Tu, Li-Yeh Chuang, Jun-Yang Chang, and Cheng-Hong Yang,‖ Feature Selection using PSO-SVM‖, IAENG International Journal of Computer Science, 33:1, IJCS_33_1_18.

[7] Jing Jie etal,‖ Multi-swarm Particle Swarm Optimization Based on Mixed Search Behavior‖, IEEE , 978-1-4244-5046-6/10,2010

[8] CHULMIN YUN, BYONGHWA OH, JIHOON YANG AND JONGHO NANG,‖ Feature Subset Selection Basedon Bio-Inspired Algorithms―,JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 1667-1686 (2011).

[9] Yuanning Liu, ―An Improved Particle Swarm Optimization for Feature Selection‖,Science direct, Journal of Bionic Engineering 8 (2011)

[10] Ching-Yi Cheo,‖ Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis‖, Proceedings of the 2004 IEEE Taiwi. Taiwan. March 21-23, 2004 lntematianai Conference on Networking, Sensing Control, 0-7803-81 93-9/04,2004

[11] Xiangyang Wang etal,‖ Feature Selection based on Rough Sets and Particle Swarm Optimization ―,Institute of Automation, Shanghai Jiao Tong University, Shanghai 2000,30

[12] Guyon I, Elisseeff A. An introduction to variable and feature selection. Journal of Machine Learning Research, 2003, 3, 1157–1182.

[13] Kennedy J, Eberhart R. Particle swarm optimization. Pro-ceedings of the IEEE International Conference on Neural Network, Perth, Australia, 1995, 1942–1948.

[14] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Sym-posium on Micro Machine and Human Science, Nagoya, Japan, 1995, 39–43.