I.J. Information Technology and Computer Science, **2012, 5, 16-24**
Published Online May 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2012.05.03

Feature Selection using a Novel Particle Swarm Optimization and It's Variants

R. Parimala
Research Scholar, National Institute of Technology, Tiruchirappalli
Email: rajamohanparimala@gmail.com

Dr. R. Nallaswamy
Professor, Department of Mathematics, Tiruchirappalli
Email: nalla@nitt.edu

Abstract— **Feature selection has been keen area of research in classification problem. Most of the researchers mainly concentrate on statistical measures to select the feature subset. These methods do not provide a suitable solution because the search space increases with the feature size. The FS is a very popular area for applications of population-based random techniques. This paper suggests swarm optimization technique, binary particle swarm optimization technique and its variants, to select the optimal feature subset. The main task of the BPSO is the selection of the features used by the SVM in the classification of spambase data set. The results of our experiments show a very strong relation between number of features and accuracy. Comparison of the optimized results and the un-optimized results showed that the BPSO-MS method could significantly reduce the computation cost while improving the classification accuracy.**

Index Terms—**Feature selection, Support Vector Machine, Particle Swarm optimization**

1. Introduction

The Particle Swarm Optimization algorithm (shortened as PSO) is a novel population-based stochastic search algorithm. It is an alternative solution to the complex non-linear optimization problem. PSO is an evolutionary computation technique developed by Dr. Ebernet and Dr. Kennedy in 1995 inspired by social behaviour of bird flocking or fish schooling [9]. While searching for food, the birds either scattered or go together before they find the place where they can find the food. Because they are transferring the information, especially the good information while searching the food from one place to another, conducted by the good information, the birds will eventually flock to the place where food can found. As far as particle swam optimization algorithm concerned, solution swarm compared to the bird swarm, the birds moving from one place to another is equal to develop the solution swarm, good information is equal to the most optimist solution, and the food is equal to the most optimist solution during the whole course. PSO is particularly attractive for feature selection in that particle swarms will discover the best feature combinations as they fly within the problem space. PSO has strong search ability in the problem space and can discover optimal solutions quickly.

The PSO is initialize with a population of random solutions and searches for optima by updating generations, In PSO, the potential solutions, called particles, are "flown" through the problem space by following the current optimum particles. The members of entire population upheld through the search procedure so information socially shared among individuals to direct the search towards the best solution in the search space [10.11]. In last few years, the PSO became frequently applied in wrappers [20, 21]. The BPSO is also a very common search technique in feature selection studies. PSO as a novel computational intelligence technique has succeeded in many continuous problems. But in discrete or binary version there are still some difficulties. In binary PSO, each particle represents its position in binary values which are 0 or 1. Each particle's value can from one to zero or conversely. In binary PSO the velocity of a particle defined as the probability that a particle might change its state to one. The proposed novel binary PSO algorithm called and applied it on a wrapper feature selection for SVM classifier. The performance of the proposed algorithm is evaluated using several UCI machine learning Spambase dataset.

2. Binary Particle Swarm Optimization (BPSO) and its variants

Standard PSO influenced the different parameters, namely dimension of the problem, number of individuals and inertia weight [13, 14]. Two variants of BPSO algorithm stated as BPSO with a local neighbourhood and BPSO with a global neighbourhood. According to the global neighbourhood, each particle move towards its best previous position and towards the best particle in the whole swarm, called gbest. Local variant called Lbest, each particle move towards its best previous position and towards the best particle in its restricted neighbourhood. Initialize the swarm population. The binary particle swarm optimization say μ- triples ($X^{(k)}$, $X^{(k)*}$,$V^{(k)}$), $1 \leq k \leq \mu$, as particles. Let $X_{ij}^{(k)}$

=[p0,p1,p2,p3..pn], pi, the position of the particle can take either 0 or 1. 0 represent absence of feature and 1 represent the presence of feature.

Selecting good BPSO parameters has been the subject of research. Pedersen et al (2000) presented a simple way of tuning the BPSO parameters [15,16]. The technique for tuning PSO parameters called meta-optimization. The inertia weight employed to control the impact of the previous history of velocities on the current one. At time t update velocity from the previous velocity to the new velocity.

$$V_{ij}(t+1) = w\,V_{ij}(t) + c_1 r_1 (X_{ij}^{p}(t) - X_{ij}(t)) + C_2 r_2 (X_{ij}^{g}(t) - X_{ij}(t))$$

$X_{ij}(t+1)$ is the n-dimensional position vector of particle i at iteration k,

$V_{ij}(t+1)$ is the n-dimensional velocity vector of particle i at iteration k,

$X_{ij}^{p}(t)$ is the n-dimensional personal best of particle i found through iteration k,

$X_{ij}^{g}(t)$ is the n-dimensional social best of particle i found through iteration k:

The sum of the previous position with the new velocity determine the new position

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (10)$$

$X_{ij}(t+1) = 1$, if $\frac{1}{1+\exp(-x_{ij})} < u[0,1]$

=0, otherwise.

where w stands for the inertia weight between 0 and 1 which simulates friction, r_1 and r_2 are the random numbers which to keep up diversity of the population, uniformly distributed in the interval [0, 1] for the j^{th} dimension of i^{th} particle. C_1 is a positive constant, called as coefficient of the self-recognition part. C_2 is also a positive constant called as coefficient of the social part; a particle decides where to move next, considering its own experience the memory of the best past position and its most successful particle in the swarm. The parameter w regulates the trade-off between global and local exploration abilities of the swarm. A large inertia weight helps global exploration while a small one helps global local exploration. A suitable value for the inertia weight w usually provides balance between global and local exploration abilities. This results in reduce the number of iterations needed to find the optimum solution. The parameters $C_1=C_2=2$ can set as default values [9]. Some experiment result show that $C_1=C_2=1.49$ might give even better results swarm size value might be 20. Each particle performance measured according to the fitness function.

2.1. Neighborhood topologies

In PSO, individuals, referred to as particles, "flown" through hyperdimensional search space. The position of particles changes within the search space based on the social-psychological tendency of individuals to copy the success of other individuals. The changes to a particle within the swarm therefore influenced through the experience, or knowledge, of its neighbors. The search behavior of a particle affected by that of other particles within the swarm therefore PSO is the symbiotic cooperative algorithm. The result of modeling this social behavior is the search process is particles stochastically return toward previously successful regions in the search space. The PSO based on the neighborhood principle as social network structure. For the global best PSO, the neighbourhood for each particle is the entire swarm. The social networking employed by gbest PSO reflects the star topology, where the social part of the velocity equation reflects the information got from the entire swarm [26]. The local best PSO, lbest, uses a ring social network topology, where smaller neighbourhoods defined for each particle [11]. The social part reflects the information exchanged within the neighborhood of the particle.

Various types of neighborhood topologies explored and presented in literature.

2.1.1 Ring Topology

Signals travel around the loop in one direction pass through each node acting like a repeater to the signals and send it on to the nest particle. A communication delay is directly proportional to the number of nodes in the network. This slow propagation will enable the particles to explore more areas in the search space and thus decreases the chance of premature convergence.

2.1.2 Star Topology

Signals from each station rebroadcast them to another

2.1.3 Hybrid Topology

In Hybrid topology (or model) star, ring and Von Neumann topologies combined together in the same algorithm. For each generation, the particle will analyze its next position using all different topologies. Particle will select the topology with the smallest fitness value and will update its velocity and position according to it.

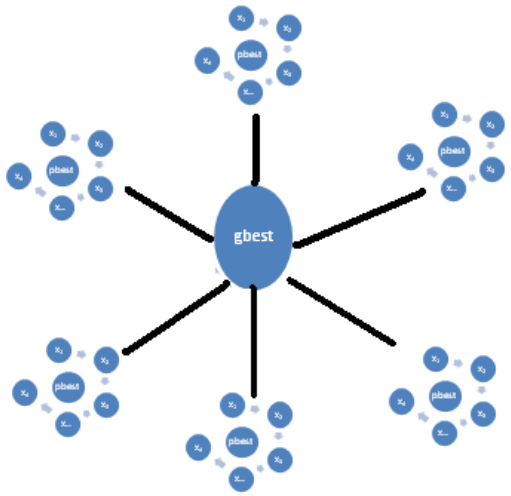

Fig2: Network Topology of PSO

2.2. Binary Particle swarm optimization using Average Velocity (BPSO-AV)

Using stereo metric digital photography and computer vision techniques, Roman Physicist measured in real time the positions and velocities of individual starlings in flocks of various sizes. They reported that in a flock of birds, despite the absence of any central coordination, a coordinated emerges. This collective behavior for the flock follows from simple rules followed by the individual birds. The Roman study shows the clear orchestration of the flock is more robust and more profound than expected. It appears the behavior of the flock as a whole is independent of the size of the flock. More in particular, two birds 1 m apart in a 10 m wide flock move as coordinated as two birds 10 m apart in a 100 m wide flock. Even more astonishing, it appears that this behavior can extrapolate down to two birds: a flock of arbitrary size (the researchers measured flocks from 122 to 4268 birds) behaves similar to a 'flock' of two birds.

Key to this discovery is to apply a Galilean perspective to the flock dynamics. As the velocities of the individual birds known at any given moment during the measured flight of the flock the researchers could draw relative velocities as well. This allowed the researchers in their analysis 'to fly with the flock'. At each time they derived the average velocity of the birds in the flock, and subtracted that figure from the velocities of the individual birds.

The average of absolute value of all velocity of all particles can be used as an index to understand all the particles in the swarm.

$V_{avg} = \frac{1}{m.n}\sum_{i=1}^{m}\sum_{j=1}^{n}\left|V_{ij}\right|$, m is the size of the swarm, and a particle with n dimensionality.

The value V_{avg} can express the activity of the swarm. If the values of the parameter are not suitable, the absolute value of velocity can increase or decrease rapidly. The average velocity gradually deceases, a good solution is got, and the average velocity gradually increases, and the search ends in failure.

The velocities update equation as

$$V_{ij}(t+1) = wV_{ij}(t) - avg(V_{ij}(t)) + c_1 r_1 (X_{ij}^{p}(t) - X_{ij}(t)) + C_2 r_2 (X_{ij}^{g}(t) - X_{ij}(t))$$

2.3. Binary Particle swarm optimization using Optimum Velocity (BPSO-Opv)

The Particle Swarm Optimization algorithm composed of a collection of particles that move around the search space. It influences their own best past location and the best past location of the whole swarm and the optimum best past location of the whole swarm. Three variants of BPSO algorithm, namely BPSO with a local neighbourhood, BPSO with a global neighbourhood and BPSO with optimal neighbourhood proposed. According to the global neighbourhood, each particle move towards its best previous position and towards the best particle in the whole swarm, called gbest. Optimal variant called Obest, each particle move towards its best previous position and towards the best particle in its restricted neighbourhood. Optimal variant called Obest, each gbest move towards its optimum position. Assume that with the initial velocity, the global particle move towards the optimum position in the swarm is called Obest.

Update the particle's velocity in each iteration using:

$$V_{ij}(t+1) = wV_{ij}(t) + c_1 r_1 (X_{ij}^{p}(t) - X_{ij}(t)) + C_2 r_2 (X_{ij}^{g}(t) - X_{ij}(t)) + C_3 r_3 (X_{ij}^{og}(t) - X_{ij}(t))$$

The values of ***c1, c2*** and ***c3*** control the weight balance of personal best and global best and optimum best particles. It used to decide the particle's next movement velocity. At every generation, the particle's new location is calculated by adding the particle's current velocity to its location,

2.4. Binary Particle swarm optimization using Gravitational Search Algorithm(BPSO-GS)

Heuristic algorithm mimics physical or biological processes. Each particle has mass m, position and velocity. The parameters Vmax and Fmax restricted the maximum force exerted on the particle and the velocity of the particle, respectively. GSA is a new multiagent optimization algorithm; inspired from the general gravitational law [11].The algorithm is based on the movement of some particles under the effect of the gravitational forces, applied by the others. Using Newton's Law, the force between two particles i and j is directly proportional to product of their masses and inversely proportional to the square of the distance between them.

$F = G\frac{m_i m_j}{r^2}$. Each individual mass is inversely proportional to its objective function. Each individual is driven by the total force exerted on it. The sum of the previous position and the new velocity determine its new position.

The velocity as

$$V_{ij}(t+1) = wV_{ij}(t) + c_1 r_1 (X_{ij}^{p}(t) - X_{ij}(t)) + C_2 r_2 (X_{ij}^{g}(t) - X_{ij}(t)) + H\frac{F_i}{m_i}$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1)$$

2.5. Binary Particle swarm optimization with Time Varying Parameters (BPSO-TVP)

2.5.1. Inertia Weight

Different inertia weights *w*1 under different maximum velocities (Vmax) allowed have been chosen for simulation. The inertia weight w1 is employed to control the impact of the previous history of velocities on the current velocity, thus to influence the trade-off between global (wide-ranging) and local (nearby) exploration abilities of the "flying points". A larger inertia weight w1 helps global exploration (searching new areas) while a smaller inertia weight helps local exploration to fine-tune the current search area. Suitable selection of the inertia weight w1 can provide a balance between global and local exploration abilities and thus need less iteration on average to find the optimum.

The time varying inertia weight that is linearly reduced during the iterations to improve the computational efficiency introduced as

tw<-(max(w1)-mean(w1))*(maxiter-j)/maxiter+mean(w1)

w1<-w1+(w1-tw*wmin)/(maxiter-j)

2.5.2. Acceleration Coefficients

The time-varying acceleration coefficients introduced efficiently control the search and convergence to the global solution.

c1<-(c1max-c1min)*(maxiter-j)/maxiter+c1min

c2<-(c2min-c2max) + (maxiter-j)/maxiter+c2max

In time varying pso and average velocity pso, the velocity is zero and the particle in the stagnation state is finding randomly. Therefore, add a mutation operator to PSO should improve its global search capacity and thus improve its performance. Particle's position mutated to increase the diversity of the algorithm and to prevent premature convergences.

2.6. Binary Particle swarm optimization with Monotonic Search (BPSO-MS)

In general, the Branch & Bound algorithm, starting the search with all the *D* features and then

applying a backward elimination feature strategy, until they obtain *d* optimal features ($d < D$). Additionally, they use a monotonic subset feature evaluation criterion: i.e., when augmenting (subtracting) one feature to the feature subset, the criterion value function always increases (decreases). The monotonicity property allows pruning unnecessary sub-trees. In out BPSO-MS, starting the search with randomly chosen features and then evaluate global best particle using a monotonic subset feature evaluation criteria. The time complexity is less, compared to Branch and bound algorithm.

3. Support Vector Machine

Support Vector Machines (SVMs) are a machine learning model proposed by V. N. Vapnik [2]. The basic idea of SVM is to find an optimal hyperplane to separate two classes with the largest margin from pre-classified data. After this hyperplane determined, used for classifying data into two classes based on which side they located. By applying proper transformations to the data space before computing the separating hyperplane, SVM can extend to cases where the margin between two classes is non-linear.

3.1. Maximal Margin Hyperplanes

Machine Learning algorithm has produced a model of the training data, used to classify new un-labeled documents automatically. SVM is a new paradigm of learning system. Since 1990s SVM has been a promising tool for data classification [3][4]. This introduction to Support Vector Machines (SVMs) based on [1], [2] and [3]. Support vector machines (SVMs) [5][6] are of great interest to theoretical and applied researchers and they have strong connections to computational learning theory. The basic idea is easiest to understand, when we have a linearly separable two-class problem. The resulting classifier called the maximal margin classifier [6][7]. The idea is to search the optimal separating hyperplane which has the maximal margin of separation between the training vectors from the two classes, so maximal margin classifiers estimate directly the decision boundary[2]. Being a separating hyperplane means the training vectors from the two classes lie on different sides of the hyperplane, and having maximal margin means that distance from the hyperplane to the nearest training vector is maximal. The support vectors are those training vectors which lie nearest to the optimal hyperplane. This optimization problem formulated as a quadratic programming problem. In real applications, the training data is usually not linearly separable and then the maximal margin hyperplane does not exist. A solution is to seek the so-called soft-margin hyperplane instead. Also this leads to a quadratic program. Since interpret of SVM classifiers leads to standard convex optimization problems, no

complications with local minima as there are with MLPs. These quadratic programs solved either by general purpose quadratic program solvers or by techniques developed specially for SVMs.

If the training data are linearly separable then there exists a pair $(\mathbf{w}, b)$

$$\begin{aligned} &\mathbf{w}^T\mathbf{x}_i + b \geq 1, \text{for all}\, \mathbf{x}_i \in P \\ &\mathbf{w}^T\mathbf{x}_i + b \leq -1, \text{for all}\, \mathbf{x}_i \in N \end{aligned} \quad (1)$$

with the decision rule given by

$$f_{\mathbf{w},b}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + b) \quad (2)$$

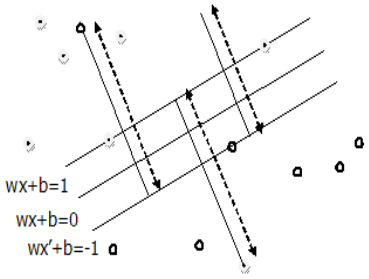

Figure 1. Optimal separating hyperplane for Binary classification problem.

$\mathbf{w}$, termed as the weight vector and b the bias (or $-b$ is termed the threshold). The inequality constraints (1) can combined to give

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \text{for all}\, \mathbf{x}_i \in P \cup N \quad (2)$$

3.2. Support Vector Machines

Given a training set of instance-label pairs (x_i, y_i), i =1, 2, 3 …ℓ where $x_i \in R^n$. The class label of the i[th] pattern is meant by $y_i \in \{1, -1\}^t$. Nonlinearly separable problem are often solved by mapping the input data samples x_i to a higher dimensional feature space $\phi(x_i)$. The classical maximum margin SVM classifier aims to find a hyperplane of the form $w^t\phi(x) + b = 0$, which separates patterns of the two classes. So far we have restricted ourselves to the case where the two classes are noise-free. In case of noisy data, forcing zero training error will lead to poor generalization. To take account of some data points misclassified, we introduce a vector of slack variables $\Xi = (\xi_1, \ldots, \xi_l)^T$ that measure for violation of the constraints (2). The problem can then be written

$$\underset{w,b,\xi}{\text{Minimize}} \frac{1}{2} w^t w + C\sum_{i=1}^{n} \xi_i \quad (4)$$

subject to the constraints

$$\begin{aligned} &y_i\left(w^t\phi(x_i) + b\right) \geq 1 - \xi_i \\ &\xi_i \geq 0, i = 1, 2, 3 \ldots\ldots \ell, \end{aligned} \quad (5)$$

The solution to (4)-(5) yields the soft margin classifier, so termed because the distance or margin between the separating hyperplane $w^t\left(\phi(x) + b\right) = 0$ usually determined by considering the dual problem, given by

$$L(\mathbf{w}, b, \alpha, \Xi, \Gamma) = \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^{l} \alpha_i [y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) - 1 + \xi_i] - \sum_{i=1}^{l} \gamma_i \xi_i + C\sum_{i=1}^{l} \xi_i$$

where $\Lambda = (\alpha_1, \ldots, \alpha_l)^T$ as before, and , $\Gamma = (\gamma_1, \ldots, \gamma_l)^T$ are the Lagrange multipliers corresponding to the positivity of the slack variables. The solution of this problem is the saddle point of the Lagrangian given by minimizing L with respect to $\mathbf{w}, \Xi$ and b , and maximizing with respect to $\Lambda \geq 0$ and $\Gamma \geq 0$. Differentiating with respect to $\mathbf{w}$, b and Ξ and setting the results equal to zero.

We get

$$\frac{\partial L(\mathbf{w}, b, \alpha, \Xi, \Gamma)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} \alpha_i y_i \phi(\mathbf{x}_i) = 0 \,,$$

$$\frac{\partial L(\mathbf{w}, b, \alpha, \Xi, \Gamma)}{\partial b} = -\sum_{i=1}^{l} \alpha_i y_i = 0 \,,$$

and

$$\frac{\partial L(\mathbf{w}, b, \Lambda, \Xi, \Gamma)}{\partial \xi_i} = C - \alpha_i - \gamma_i = 0 .$$

$$\underset{\alpha}{Minimize} \frac{1}{2}\sum_{i=1}^{\ell}\sum_{j=1}^{\ell} y_i y_j \alpha_i \alpha_j k(x_i, x_j) - \sum_{i=1}^{\ell} \alpha_i \quad (6)$$

subject to

$$\sum_{i=1}^{\ell} \alpha_i y_i = 0, \text{ and } 0 \leq \alpha_i \leq C, i = 1, 2, 3 \ldots\ldots \ell \quad (7)$$

Here, $\alpha_i, i = 1, 2, 3, \ldots\ldots \ell$ denotes the Lagrange multipliers and the matrix $K(x_i, x_j) = \phi(x_i).\phi(x_j)$ termed as Kernel matrix [17, 18]. Training vector x_i is mapped into a higher dimensional feature space and build an optimal hyperplane. SVM also restrict the choice of Kernel. The Quadratic programming is a convex problem; therefore, it guarantees that global optimization with corresponding Kernel. SVM uses training data as Support Vectors and uses Lagrange multipliers to represent the Support Vectors. The classifier can built using the decision function in the form

$$y(x) = sgn\left[\sum_{k=1}^{\ell} \alpha_k y_k K(x, x_k) + b\right]$$

4. Methods

4.1. Direct Use of SVM

The penalty factor for training the SVM and PSO-SVM set to 5. RBF kernel function is a universal kernel function; after selection of the relevant parameters, it can apply to arbitrary distributive samples. In conclusion, RBF kernel function generally applied in the Support Vector Machine [6]. The generalization ability of SVM algorithm depends on a set of parameters.

Take RBF kernel function as the kernel function. The parameters needs to optimized are: RBF kernel parameter and the estimated accuracy. Use the 10-fold method to estimate the generalization ability. The original data set was randomly divided into a two-third of a set (training set) and one-third of a set (testing set). The basic step is stated as follows:

1. Input the sample training set, and set a group of parameters {C, cross}.
2. Train SVM based on the parameters. Calculates the cross validation error and obtains its object.
3. Test the SVM using object obtained from step 2.
4. Repeat the above step 25 times and find the average testing accuracy.

4.2. Implementation of PSO-SVM

The procedure for describing proposed PSO-SVM is as follows:

1. Initialize PSO with population size, inertia weight.
2. Set cognitive and social learning rate as 2.
3. Set the number of particles and its dimension.
4. Train SVM on particle.
5. Evaluate the fitness value of each particle. Take the cross validation error of the SVM training set as fitness value.
6. Compare the fitness values and calculates the local best and global best.
7. Update the inertia weight, velocity and position of the each particle.
8. Repeat the step 4-6 until a value of the fitness function converges or the number of iteration reached.
9. After converging, the global best object is fed in to SVM classifier for testing.

5. EXPERIMENT AND ANALYSIS

Spam defined as unsolicited email messages and the goal of spam classification is to distinguish between spam and legitimate email messages. Many researchers have been trying to separate spam from legitimate emails using machine learning algorithms based on statistical learning methods. Most text classification approaches use supervised learning for building a classification system. Several solutions have proposed to overcome the spam problem. Among the proposed methods, much interest has focused on the machine learning techniques in spam mail classification. A data set collected at Hewlett-Packard Labs that classifies 4601 e-mails as spam or non-spam. In addition to this class label, there are 57 variables indicating frequency of certain words and characters in the e-mail. We have used the spam data set for training and testing the spam e-mail classifier. The performance measure of BPSO and its variants are given in Table.1.

5.1. Result and Discussion

The accuracy (acc) is the percentage of total cases correctly classified [19]**.** We measured the classification accuracy of data sets with full features first using the SVM classifier by 10 fold Cross validation. We measured the accuracy of datasets and calculate its performance using paired t-test. Normally say that a P value of .05 or less is significant. Except feature selection BPSO-TVP method on spambase data set, all other methods are significant. Table I. shows the performances of average accuracy of Gravitational search perform poorly. Although direct use of SVM yields the highest classification accuracy with no feature lessened and hence the classification result is equal to original result. The BPSO with monotonic search give optimal feature subset. The BPSO with optimum velocity got the next highest classification accuracy. The proposed method can solve as pre-processing tool and help to optimize the feature selection process, which leads to an increase in classification accuracy. A good feature selection method lessens the number of features and improves accuracy. The list of method has the highest classification accuracy listed in Table II.

Table II. List of Methods in the order of Accuracy based on Coefficient of variation.

1	BPSO with Monotonic Search
2	BPSO with Average Velocity
3	BPSO with Optimum Velocity
4	Standard BPSO
5	BPSO-Time varying parameter
6	Without BPSO
7	BPSO-Time varying parameter with Mutation
8	BPSO – Average Velocity with mutation
9	BPSO- Gravitational search

Table I. Performance Measure of BPSO and its variants

Method	Average		Maximum		Minimum		Variance		Coefficient of variation	
	#f	Acc	#f	Acc	#f	Acc	f	Acc	f	Acc
without PSO	57	93.03	57	93.15	57	92.94	0	0.0119	0	0.012
Standard BPSO	21	90.70	23	92.50	20	88.91	1.3667	0.0106	5.567	0.011
BPSO - Opv	21	90.41	28	91.00	16	90.41	14.667	0.0010	18.367	0.008
BPSO-AV	22	91.13	25	94.26	20	89.37	04.500	0.00039	9.6420	0.00414
BPSO– AVM	37	91.76	44	93.41	32	90.61	24.920	1.5529	13.491	1.361
BPSO-TVP	42	92.72	44	93.70	39	91.50	08.330	0.0001	6.8730	0.012
BPSO-TVPM	34	91.77	39	92.50	32	90.41	07.950	0.5693	8.2940	0.822
BPSO-GS	20	90.13	24	91.13	13	89.30	24.920	1.5593	19.732	1.385
BPSO-MS	16	89.43	18	89.75	15	89.23	1.567	3.60E-06	7.8237	0.00212

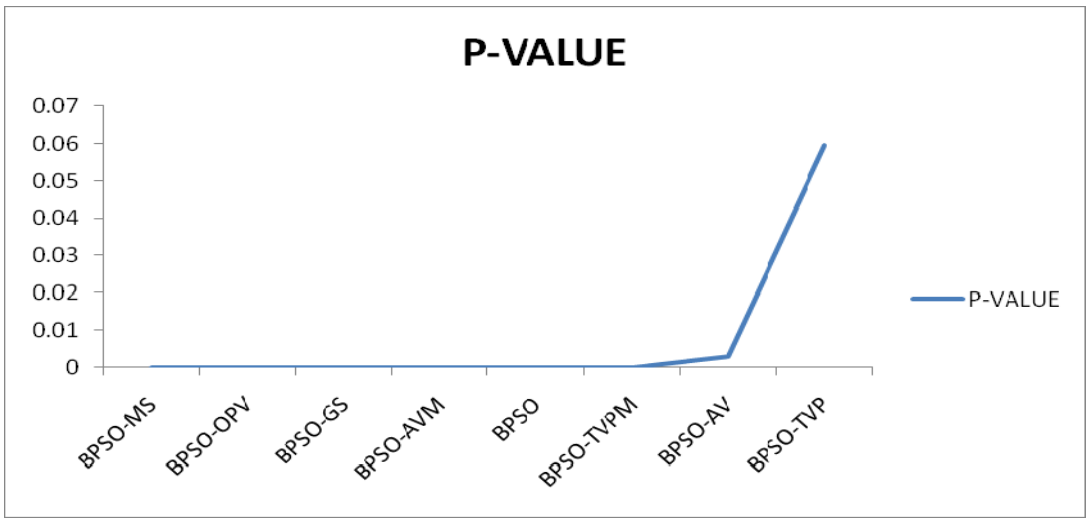

Fig1: Performance Measure using Paired t-test

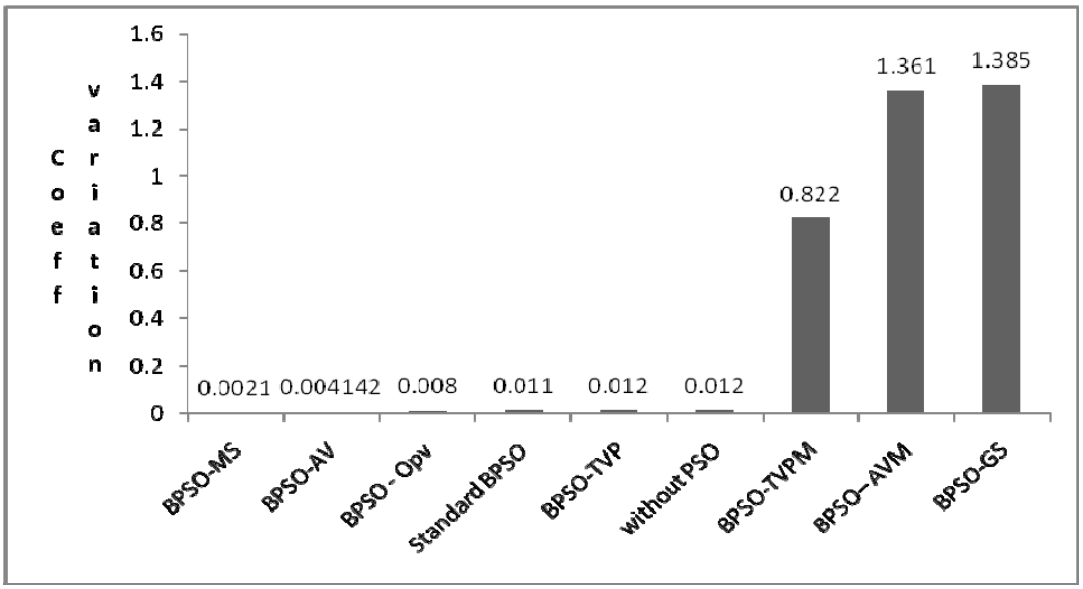

Fig1: Performance Measure using Coefficient of Variation

5.2. Used Environment and Libraries

Within the last year's tm has gained interest from a various researchers and users of different backgrounds R is a programming language and software environment for statistical computing and graphics. R is more than a programming language. It is an interactive environment for doing statistics. We find it more helpful to think of R as having a programming language than being a programming language. The R language is the scripting language for the R environment. An R interface has added to the popular data mining software Weka which allows for the use of the data mining capabilities in Weka and statistical analysis in R. kernlab for kernel learning provides ksvm and is more integrated into R so different kernels can easily explored [27,28]. The machine used was an Intel Core 2 Duo E7500 @ 2.93GHz with 2GB RAM.

6. Conclusion

The BPSO based Feature selection method applied to the whole feature space, select the best feature subset. Also we noted the classifier accuracy is decreased significantly when the mutation applied. The inertia parameter, acceleration coefficient, position updating tactics and the fitness function have been important. From the result got we conclude that BPSO has powerful exploration ability, it is a gradual searching process that approaches optimal solution. Using the proposed BPSO-SVM-based feature selection scheme, feature dimensionality is reduced and classification performance of the SVM classifier is greatly enhanced.

The coefficient of variation defined on number of features decides the search manner of BPSO method.

References

[1] C.J.C. Burges. A tutorial on support vector machines for pattern recognition". *Data Mining and Knowledge Discovery*, 2(2): 955-974, 1998.

[2] V. N. Vapnik. *The nature of Statistical Learning Theory*. Springer, Berlin, 1995.

[3] N. Cristianini, and J. Shawe-Taylor, "Support Vector and Kernel Methods, Intelligent Data Analysis: An Introduction Springer – Verlag", 2003.

[4] N.Cristianini, and J. Shawe-Taylor, "An introduction to support vector machines, Cambridge, UK: Cambridge University Press", 2004.

[5] B. Schölkopf. C.J.C. Burges, and A.J. Smola,"Advances in Kernel Methods: Support Vector Learning", MIT Press, (Eds.), 1998.

[6] A.J. Smola and B. Scholkopf, "Learning with kernels: Support Vector Machines, regularization, optimization, and beyond", Cambridge, MA: MIT press.

[7] R.C. Eberhart, and J. Kennedy. "A new optimizer using particle swarm theory", Proceedings of the sixth international symposium on micro machine and human science pp. 39-43, IEEE service center, Piscataway,NJ, Nagoya, Japan, 1995.

[8] R.C. Eberhart, and Y. Shi. "Particle swarm optimization: developments, applications and resources". Proc. congress on evolutionary computation 2001 IEEE service center, Piscataway, NJ., Seoul, Korea., 2001.

[9] Y. Shi, and R.C. Eberhart, "Parameter selection in particle swarm optimization", volutionary Programming VII: Proc. EP 98 pp. 591-600. Springer-Verlag,, New York, 1998.

[10] M.Carvalho, and T.B. Ludermir, "Particle swarm optimization of neural network architectures and weights", In Proc. of the 7th int. conf. on hybrid intelligent systems, (pp. 336_339), 2007.

[11] M. Meissner, M. Schmuker, and G. Schneider, "Optimized particle swarm optimization (OPSO) and its application to artificial neural network training", BMC, Bioinformatics, 7, 125, 2006.

[12] J. Yu, L. Xi, and S. Wang, "An improved particle swarm optimization for evolving feed forward artificial neural networks", Neural Processing Letters, 26(3), 217_231, 2007.

[13] J. Salerno. "Using the particle swarm optimization technique to train a recurrent neural model", IEEE International Conference on Tools with Artificial Intelligence, 45_49, 1997.

[14] M. Settles, B. Rodebaugh, and T. Soule,"Comparison of genetic algorithm and particle swarm optimizer when evolving a recurrent neural network", Lecture notes in computer science (LNCS): Vol. 2723, Proc. of the genetic and evolutionary computation conference, pp. 151_152, 2003.

[15] M.E.H. Pedersen, "Tuning and Simplifying Heuristical Optimization",PhD Dissertation, University of Southampton, 2010.

[16] M.E.H. Pedersen, and A.J. Chipperfield, Simplifying particle swarm optimization, Applied Soft Computing 10 (2) (2010) 618–628.

[17] Karatzoglou, A., Smola, A., Hornik, K,, Zeileis, A., 2005, "kernlab – Kernel Methods.", R package, Version 0.6-2., Available from http://cran.R-project.org.

[18] Alexandros Karatzoglou and Ingo Feinerer, Kernel-based machine learning for fast text mining in R. *Computational Statistics & Data Analysis*, 54(2):290-297, February 2010.

[19] C. J. van Rijsbergen., 1979," Information Retireval". Butterworths, London.

[20] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee. Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Systems with Applications, 35:1817-1824, 2008. 12, 14, 16, 24

[21] M. A. Esseghir, G. Goncalves, and Y. Slimani. Adaptive particle swarm optimizer for feature selection, In Proceedings of the 11th International Conference on Intelligent Data Engineering and

Automated Learning, IDEAL 2010, pages 226{233. Springer-Verlag, 2010. 13, 16

Author Biographies

R. Parimala graduated with M.Sc., of Applied Science at the National Institute of Technology, (formerly Regional Engineering College)Tiruchirapalli in 1990. She received her M.Phil., Computer Science at Mother Teresa University, Kodaikanal in 1999. She started teaching in 1999 at National Institute of Technology and is currently working as Assistant Professor in Department of Computer Science, Periyar E.V.R. College (Autonomous), Tiruchirapalli. Presently, she is a P.hD., Research Scholar in National Institute of Technology, Tiruchirappalli. Her area of research interests include Neural Networks, Data mining and Optimization Techniques.

Dr. R. Nallaswamy received his degree M.Sc., Applied Mathematics and Ph.D from Indian Institute of Technology, Kanpur, Currently, he is working as Head & Professor at Department of Mathematics, National Institute of Technology, Tiruchirappalli, He is a member of ISTE and ISMMS. He published more than 12 papers in International Journal. His area of research interest includes Bio Mathematics, Applied Statistics, Optimization Techniques and Soft Computing.

