International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 01, January 2017
www.ijates.com
ijates
ISSN 2348 - 7550

A BINARY PSO-ACO HYBRID ALGORITHM FOR FEATURE SUBSET SELECTION

Neeta Agarwal

Applied Sciences , BBDIT Ghaziabad , (India)

ABSTRACT

Feature Selection is the process of selecting a subset of features available, allowing a certain objective function to be optimized, from the data containing noisy,irrelevant and redundant features. This paper presents a novel feature selection method that is based on hybridization of ACO with a binary PSO to obtain excellent properties of two algorithms by synthesizing them and aims at achieving similar or better results than PSO-based feature selection and ACO-based feature selection. The fundamental idea of this hybrid approach is that the PSO is employed at the beginning of the searching process,as PSO has the ability to eplore the search space .The best feature subset having best gbest that minimizes the classification error is selected.The searching process is switched to the ACO algorithm; the best subsets found by PSO is the initial population of ACO.

From the features of the gbest having m features each ant start with randomly produce m-p feature subset .ACO works as a local search,wherein,ants apply pheromone-guided mechanism to refine the positions found by particles in PSO stage.In PSACO ,a simple pheromone-guided mechanism of ACO is proposed to apply as local search.Thus ACO helps PSO process for rapidly and effectively attaining the optimal and near optimal solution(subset). Proposed algorithm is applied to a biometric feature selection problem of left index knuckle dataset.

Keywords – Ant Colony Optimization , Binary PSO , Feature selection , Hybrid Optimization.

I. INTRODUCTION

Feature Selection is extensive and spread across many fields.In Biometrics, to establish an identity based on the physical attributes of an individual, images of these attributes are used as patterns(e.g.fingerprint,face,iris,hand,knuckle etc.).Different features are extracted to characterize the images. The end result of the extraction task is a set of features, commonly called a feature vector, which constitutes a representation of the image. Only a small subset of these features are necessary in practice for building an accurate model for identification.

A reduced feature subset requires fewer patterns in the training procedure of a classification algorithm. In addition, the training procedure would take less time due to fewer features and also better accuracy of classifier is achieved to improve the prediction accuracy.

In real-world problems, manual feature selection is often impossible to achieve due to the large number of features. Therefore, feature selection is necessary in these problems. The goal of feature selection is to find those features that may neither affect the target in any way (called irrelevant features) nor add anything new to the target (called redundant features) and exclude them.

A feature selection algorithms consist of the following four components

1.1 Starting point in the feature space

search for feature subsets could start with (i) no features, (ii) all features, or (iii) random subset of features. In the first case, the search proceeds by adding features successively, while in the second case, features are successively removed. When starting with a random subset, features could be successively added/removed, or reproduced by a certain procedure.

1.2 Search procedure

Three main methods based on their subset search strategies:*exhaustive*, *heuristic*, and *random* search strategy.Ideally, the best subset of features can be found by evaluating all the possible subsets, which is known as *exhaustive search*. However, this becomes prohibitive as the number of features increases, where there are $2^{N}-1$ possible combinations for N features. Sequential selection approaches such as sequential forward selection (SFS) and sequential backward selection(SBS) belong to *heuristic search* category. SFS starts with an empty set and gradually add the best features to it until an optimal subset is achieved. While SBS starts with the complete feature set and gradually discard the worst features until an optimal subset is retained. The disadvantage of SFS and SBS is that the features once selected/removed, cannot be discard/reselect later. Sequential forward/-backward floating selection approaches proposed to flexibly add and remove features.Among many methods which were proposed for FS problem, evolutionary optimization algorithms such as genetic algorithm (GA) , particle swarm optimization (PSO) , and ant colony optimization (ACO), have attracted a lot of attention. Typically evolutionary algorithms are *random search* techniques; these algorithms attempt to achieve better solutions by using knowledge from previous iterations.

1.3 Evaluation function

An important component of any feature selection method is the evaluation of feature subsets. Evaluation functions measure how good a specific subset can be in discriminating between classes, and can be divided into two main groups: filters and wrappers. Filters operate independently of any learning algorithm, where undesirable features are filtered out of the data before learning begins . On the other hand, performance of classification algorithms is used to select features for wrapper methods .

1.4 Criterion for stopping the search

Feature selection methods must decide when to stop searching through the space of feature subsets. Some of the methods ask the user to predefine the number of selected features. Other methods are based on the evaluation function, like whether addition/deletion of any feature does not produce a better subset, or an optimal subset according to some evaluation strategy is obtained.

In this work, we will mainly be concerned with the second component, which is the search procedure that utilizes the PSO-ACO hybrid optimization algorithm.

Feature selection is an optimization problem, since the aim is to obtain any subset that minimizes a particular measure (classification error, for instance). An optimization problem can be solved through stochastic algorithms. In the present work, an approach based on particle swarm optimization (PSO) algorithm and Ant colony optimization(ACO) algorithm is proposed.

II. EVOLUTIONARY OPTIMIZATION TECHNIQUES

2.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic optimization technique. PSO simulates the social behavior of organisms, such as bird flocking and fish schooling, to describe an automatically evolving system. In PSO, each single candidate solution is "an individual bird of the flock", that is, a particle in the search space. Each particle makes use of its individual memory and knowledge gained by the swarm as a whole to find the best solution . All of the particles have fitness values, which are evaluated by a fitness function to be optimized, and have velocities which direct the movement of the particles. During movement, each particle adjusts its position according to its own experience, as well as according to the experience of a neighboring particle, and makes use of the best position encountered by itself and its neighbor. The particles move through the problem space by following a current of optimum particles. The initial swarm is generally created in such a way that the population of the particles is distributed randomly over the search space. At every iteration, each particle is updated by two "best" values, called *pbest* and *gbest*. Each particle keeps track of its coordinates in the problem space,which are associated with the best solution (fitness) the particle has achieved so far. This fitness value is stored, and called *pbest*. When a particle takes the whole population as its topological neighbor, the best value is a global “best” value and is called *gbest*. The pseudo code of the PSO procedure is given below.

Initialize population

While (number of generations, or the stopping criterion is not met)

For p = 1 to number of particles

If the fitness of X_p is greater than the fitness of $pbest_p$ then Update $pbest_p = X_p$

For k € Neighborhood of X_p

If the fitness of X_k is greater than that of $gbest$ then Update $gbest = X_k$

Next k

For each dimension d :

$$v_{pd}^{new} = w \times v_{pd}^{old} + c_1 \times rand_1 \times (pbest_{pd} - x_{pd}^{old}) + c_2 \times rand_2 \times (gbest_d - x_{pd}^{old}) \quad (1)$$

$$x_{pd}^{new} = x_{pd}^{old} + v_{pd}^{new}$$

Next d

Next p

Next generation until stopping criterion.

Eq. (1) is used to calculate the new velocity of the particle according to its previous velocity and the distances of its current position from its own previous best experience (position) and the group’s best experience. Then, the particle flies towards a new position according to Eq. (1). The performance of each particle is measured according to a predefined fitness function. v_{pd}^{new} and v_{pd}^{old} are the particle velocities, x_{pd}^{old} is the current particle position (solution), and x_{pd}^{new} is the updated particle position (solution). The values

$pbest_{pd}$ and $gbest_d$ are defined as stated above. The two factors $rand_1$ and $rand_2$ are random numbers between (0, 1), whereas c_1 and c_2 are acceleration factors, usually $c_1 = c_2 = 2$. Particle velocities of each dimension are tried to a maximum velocity V_{max} . If the sum of velocities causes the total velocity of that dimension to exceed V_{max}, then the velocity of that dimension is limited to
V_{max} . V_{max} is a user-specified parameter.

The values of *w*, c1 and c2 control the impact of previous historical values of particle velocities on its current one. A larger value of **w** leads to global exploration, whereas smaller values results with a fine search within the solution space. Therefore, suitable selection of x; c1 and c2 provides a balance between the global and local search processes . Note that the terms $(pbest_{pd} - x_{pd}^{old})$ and $(gbest_d - x_{pd}^{old})$ in Eq.(1) are called the cognition and social terms, respectively. The cognition term takes into account only the particle's own experience, which represents the private thinking of the particle itself. whereas the social term signifies the interaction between the particles.

In order to apply the idea of PSO for feature selection , an adapt form of the general PSO Binary PSO is used. This will be the objective of the following subsections.

2.1.1 Binary PSO

The Original PSO is basically developed for continuous optimization problems such that the population of the particles is distributed randomly over the search space.It has been later extended to discretee valued population,called Binary PSO which is well adapted to the feature selection context.The particle's positions is represented as binary bit strings of length N,where N is the total dimension of the feature set. Every bit in the string represents a feature, the value '1' means the corresponding feature is selected while '0' means not selected. Each position is an attribute subset.Each particle velocity is updated according to the following equations:

$$v_{pd}^{new} = w \times v_{pd}^{old} + c_1 \times rand_1 \times (pbest_{pd} - x_{pd}^{old}) + c_2 \times rand_2 \times (gbest_d - x_{pd}^{old}) \qquad (2)$$

Particle's position is updated using a sigmoid transformation of the velocity component:

$$s(v_{pd}^{new}) = \frac{1}{1 + e^{-v_{pd}^{new}}} \qquad (3)$$

$$\text{If} \quad rand < s(v_{pd}^{new}) \quad \text{then} \quad x_{pd}^{new} = 1 \quad \text{else} \quad x_{pd}^{new} = 0$$

Where v_{pd}^{new} denotes the particle velocity obtain from the above equation,function $s(v_{pd}^{new})$ is a sigmoid transformation and rand is a random number selected from the uniform distribution (0,1). If $s(v_{pd}^{new})$ is larger than random number then its position values is represented as '1', meaning that this feature is selected as a required feature for the next stage.Else its position value is represented as '0' indicating that this feature is not selected as required feature for the next stage.

2.1.2 Fitness function

The feature selection relies on an appropriate formulation of the fitness function. The main objective of the closed identification fitness function is to maximize the recognition rate.Given the test sample,we compute its distance against all the samples in the reference dataset to obtain the match scores.Then,we select the sample from the reference data set with the lowest distance value and we check whether it belongs to the same class(same person)as the testing sample.In this case, it is a success or else a failure.We repeat this for all testing samples and count the number of success and failures. Recognition rate is calculated as follows:

$$\mathrm{Re}\,cognitionRate = \frac{N_{success}}{N_{testSamples}}$$

where $N_{testSamples}$ is the total number of test samples in the whole database and $N_{success}$ is the number of success in the whole database.

Finally,we define we define the fitness function as being the error rate given by

$$\text{Error Rate} = (\text{ 1- Recognition Rate})^2$$

2.1.3 Velocity limitation (maximum velocity $V_{\max}$)

The maximum velocity $\boldsymbol{V}_{\mathbf{max}}$ controls the global exploration ability of a particle swarm.A large value of $\boldsymbol{V}_{\mathbf{max}}$ facilitates the global exploration,while a smaller value encourages local explora-tion. In the binary version of PSO,the value of $\boldsymbol{V}_{\mathbf{max}}$ limits the probability that bit x_{pd} takes value '0' or '1' and therefore the use of high $\boldsymbol{V}_{\mathbf{max}}$ value in binary PSO will decrease the range explored by the particle. In our experiments,we tried different values of $\boldsymbol{V}_{\mathbf{max}}$ and finally selected $\boldsymbol{V}_{\mathbf{max}}= 6$, as it allows the particle to reach near optimal solutions.

2.1.4 Inertia weight and acceleration constant

The weight of inertia is an important parameter as it provides the particles with a degree of memory capability.In our experi-ments, the inertia weight is decreased along with the iterations, varying from 0.95 to 0.4 according to the following equation :

$$\text{weight} = (\text{weight} - 0.4) \times ((\text{MAXITER} - \text{iter}) / \text{MAXITER}) + 0.4$$

where MAXITER is the maximum iteration and iter is the current iteration. In our work,we choose MAXITER as 10. Even though the choice of acceleration constants C1 and C2 is not so critical for the convergence of PSO,a suitable chosen value may lead to faster convergence of the algorithm.A default value is C1=2 and C2=2. In our experiments,we choose C1=2 and C2=2.5 as it yields better convergence.

2.1.5 Population size

The population size ,i.e. the number of particles in the swarm, plays an important role,as it is not only influences the perfor-mance but also the computation cost.In present work, we fixed the population size as 10 since any further increase in this value did not provide a significant improvement in the performance.

2.2 Ant Colony Optimization

Ant colony optimization (ACO) was introduced as a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization (CO) problems .The ability of real ants to find shortest routes is mainly due to their depositing of pheromone as they travel; each ant probabilistically prefers to follow a direction rich in this chemical. The pheromone decays over time, resulting in much less pheromone on less popular paths. Given that over time the shortest route will have the higher rate of ant traversal, this path will be reinforced and the others diminished until all ants follow the same, shortest path (the "system" has converged toa single solution) .In general, an ACO algorithm can be applied to any combinatorial problem as far as it is possible to define:

Appropriate problem representation : The problem must be described as a graph with a set of nodes and edges between nodes.

Heuristic desirability (η) of edges : A suitable heuristic measure of the "goodness" of paths from one node to every other connected node in the graph.

Construction of feasible solutions : A mechanism must be in place whereby possible solutions are efficiently created.

Pheromone updating rule : A suitable method of updating the pheromone levels on edges is required with a corresponding evaporation rule. Typical methods involve selecting the n best ants and updating the paths they chose.

Probabilistic transition rule : The rule that determines the probability of an ant traversing from one node in the graph to the next.

2.3 ACO for Feature Selection

The feature selection task may be reformulated into an ACO-suitable problem. ACO requires a problem to be represented as a graph. Here nodes represent features, with the edges between them denoting the choice of the next feature.The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion.

The feature selection representation exploited by artificial ants includes the following:

n features that constitute the original set, $F = \{f_1, \ldots, f_n\}$.

A number of artificial ants to search through the feature space (*na* ants).

T_i, the intensity of pheromone trail associated with feature f_i, which reflects the previous knowledge about the importance of f_i.

For each ant j, a list that contains the selected feature subset, $S_j = \{s_1, \ldots, s_m\}$.

A classification algorithm is used to estimate the performance of subsets . In the first iteration, each ant will randomly choose a feature subset of m features. Only the best k subsets, $k < na$, will be used to update the pheromone trial and influence the feature subsets of the next iteration. In the second and following iterations, each ant will start with $m - p$ features that are randomly chosen from the previously selected k-best subsets, where p is an integer that ranges between 1 and $m - 1$. In this way, the features that constitute the best k subsets will have more chance to be present in the subsets of the next iteration. However, it will still be possible for each ant to consider other features as well. For a given ant j, those features are the ones that achieve the best compromise between pheromone trails and local importance with respect to S_j, where S_j is the subset that consists of the features that have already been selected by ant j. The Updated Selection Measure (*USM*) is used for this purpose and defined as:

ijates
ISSN 2348 - 7550

$$USM_i^{S_j} = \begin{cases} \dfrac{(\tau_i)^{\eta}(LI_i^{S_j})^{\kappa}}{\sum_{g \notin S_j}(\tau_g)^{\eta}(LI_g^{S_j})^{\kappa}} & \text{if } i \notin S_j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where LI_i^{Sj} is the local importance of feature f_i given the subset S_j. The parameters η and κ control the effect of pheromone trail intensity and local feature importance respectively. LI_i^{Sj} is measured using the coorelation measure and defined as:

$$LI_i^{S_j} = \frac{r(c; f_i)}{\sum r(f_i; S_j)} \quad (5)$$

Where $r(c; f_i)$) is the coorelation information between the "class labels" and f_i, $\sum r(f_i; S_j)$ is the sum of coorelation between the feature f_i and the features of subset S_j. The numerator of (5) provide an indication of how predictive of the class a set of features are; the denominator of how much redundancy there is among the features.A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other.The bias of the local importance is toward subsets that contain features that are highly correlated with the class and uncorrelated with each other.Irrelevant features should be ignored because they will have low correlation with the class.Redundant features should be screened out as they will be highly correlated with one or more of the remaining features.

The pheromone trail intensity is updated according to the following formula:

$$\tau_i = \rho\tau_i + \Delta\tau_i$$

$$\Delta\tau_i = \begin{cases} \dfrac{\max\limits_{g=1:k}(error_g) - error_j}{\max\limits_{h=1:k}(\max\limits_{g=1:k}(error_g) - error_h)} & \text{if } f_i \notin S_j \\ 0 & \text{otherwise} \end{cases}$$

where ρ is a constant such that (1 - ρ) represents the evaporation of pheromone trails.

III. PROPOSED SEARCH ALGORITHM

In this work one of the way to hybridize PSO and ACO for biometric feature selection of left index knuckle is discussed.Particle swarm optimization algorithm is improved using an Ant colony approach.This algorithm called PSACO is based on the comman characteristics of both PSO and ACO algorithms.The implementation of PSACO algorithm consists of two stages.In the first stage PSO is applied ,while ACO is implemented in the second stage.ACO works as a local search,wherein,ants apply pheromone-guided mechanism to refine the positions found by particles in PSO stage.In PSACO ,a simple pheromone-guided mechanism of ACO is proposed to apply as local search.

The ACO algorithm handles ants equal to number of particles in PSO.Each ant generate a solution(feature subset) around the global best-found position among all particles .For each solution(subset) generated by ants subset value is calculated using evaluating function and compare with the current positions of particles (feature subset) generated by PSO.Current position of particles are replaced by the solution giving optimum value(minimizes the fitness function).This simple pheromone –guided mechanism considers,there is highest density of trail at global best solution of swarm at any iteration in each stage of ACO implementation and all ants search for better solutions in the neighbourhood of global best solution.Thus ACO helps PSO process for rapidly and effectively attaining the optimal and near optimal solution(subset).

The performance of selected feature subsets is measured by invoking an evaluation function with the corresponding reduced feature space and measuring the specified classification result. The best feature subset found is then output as the recommended set of features to be used in the actual design of the classification system.

IV.IMPLEMENTATIONS DETAILS

The process of implementing PSACO algorithm is as follow:

4.1. Initialization:

For particle swarm optimization:

Define the maximum number of iterations.

Define number of particles.

Initialize the particle's positions(subset) as binary strings of length N(dimension of feature set) by randomly generating '0' and '1'depending how many features we want in a subset.

Initialize the particle's velocity as zero and *pbest* as the initial position's of particles.

Evaluate the subset of each particle using a Euclidean distance classifier i.e. estimate the error rate of the classification results obtained by classifying the features of the subset.Assign the subset with minimum error rate as the global best *gbest*.

For Ant colony optimization:

Number of ants are equal to number of particles.

Set $\tau_i = cc$ and $\Delta\tau_i = 0$, ($i = 1, \ldots, n$), where cc is a constant and $\Delta\tau_i$ is the amount of change of pheromone trial quantity for feature f_i.

Define k, where the k-best subsets will influence the subsets of the next iteration.

Define p, where $m - p$ is the number of features each ant will start with in the second and following iterations.

For i= maximum number of iterations

4.2. In the first iteration

For j =1 to nop (no.of particles)

Find all the particle's new velocity v_j and positions x_j using relations (2) and (3).

Apply the ACO using *gbest* having m features(no.of '1')

For j = 1 to noa (number of ants)

From the features of the *gbest* having m features randomly produce m-p feature subset for the ant j and store it in S_j.

4.3. Evaluate the selected subset of each ant using a chosen classification algorithm:

For j = 1 to *noa*,

Estimate the *Error rate* of the classification results obtained by classifying the features of S_j.

Sort the subsets according to their *Error rate* . Update the minimum *Error rate* (if achieved by any ant in this iteration), and store the corresponding subset of features.

4.4. Using the feature subsets of the best k ants, update the pheromone trail intensity and initialize the subsets for next iteration:

For j = 1 to k

$$\Delta\tau_i = \begin{cases} \dfrac{\max\limits_{g=1:k}(error_g) - error_j}{\max\limits_{h=1:k}(\max\limits_{g=1:k}(error_g) - error_h)} & \text{if } f_i \notin S_j \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\tau_i = \rho\tau_i + \Delta\tau_i$$

where ρ is a constant such that (1 - ρ) represents the evaporation of pheromone trails.

4.5. Select the remaining p features for each ant

For $mm = m - p + 1$ to m,

For j = 1 to *noa*,

Given subset S_j, Choose feature f_i that maximizes USM_i^{Sj}.

$S_j = S_j \cup \{f_i\}$.

Replace the duplicated subsets, if any, with randomly chosen subsets.

4.6. Estimate the *Error rate* of the classification results obtained by classifying the features of new S_j.compare these rates with the *Error rate* of position's (subsets) found by particles.

For j =1 to noa

If *Error rate* (S_j) < *Error rate*(x_j)

then $x_j = S_j$

4.7. Compare the particle's *Error rate*(x_j) with particle's *pbest*. If the current *Error rate* is better than *pbest*, then set *pbest* value equal to the current value and the *pbest* location equal to the current location x_j .

4.8.Find minimum of all *Error rate*(x_j).If min(*Error rate*(x_j)) is better than *gbest* than reset *gbest.*

4.9. If the number of iterations is less than the maximum number of iterations, or the desired *Error rate* has not been achieved, goto step 2.

V. EXPERIMENTAL RESULTS

The feature selection using Hybrid PSO-ACO is tested on LeftIndex Knuckle features consisting of 531 features of 165 users.For every user there is 4 training and 2 testing samples.

The parameters of the PSO algorithm are assigned the following values: Population size(no. of paticles) = 10, number of iterations = 10, acceleration constants C1=2 and C2=2.5 , Inertia weight *w* is1.2. The obtained strings are constrained to have the number of '1's matching a predefined number of desired features.

The parameters of the ACO algorithm are assigned the following values:

$\eta = \kappa = 1$, which basically makes the trail intensity and local measure equally important.

The number of ants, *noa* = 10 which is equal to number of particles in PSO.

$k = 4$. Thus, only the best *noa*/3 ants are used to update the pheromone trails and affect the feature subsets of the next iteration.

$m - p = \max(m - 5, \text{round}(0.65 \times m))$, where p is the number of the remaining features that need to be selected in each iteration. It can be seen that p will be equal to 5 if $m \geq 13$. The rational behind this is that evaluating the importance of features locally becomes less reliable as the number of selected features increases. In addition, this will reduce the computational cost especially for large values of m.

The initial value of trail intensity $cc = 1$, and the trail evaporation is 0.25, i.e., $\rho = 0.75$.

The Genuine and Imposter scores are calculated based on Euclidean Distance.There are 330 (165*2) genuine scores and 54120 (165*164*2) imposter scores.These scores are then compared with the threshold(varying with the step size of 0.01)and error rates are calculated.If the genuine score exceed the threshold,then it contributes to the false rejection rate of the person.False rejection rate(FRR) is interpreted as the number of rejected verification attempts to the number of verification attempts for a qualified person. If the imposter score is less than the threshold,then it contributes to the false acceptance rate(FAR) of the person.FAR is the ratio of the number of the successful fraud attempts to the total number of fraud attempts against a person is termed as FAR.Using these error rates,the receiver operating characteristic (ROC) which depicts the performance of an authentication system is plotted.The ROC plot is drawn FAR vs.GAR(GAR equals 100-FRR is the genuine acceptance rate),with varying threshold values.From this plot the threshold that yields the highest GAR corresponding to the lowest FAR needs to be selected.

Table shows the performance of algorithm in terms of GAR at the value of FAR = 0.03%

Method	GAR(%) (for FAR=0.03%)
Without Feature selection	68
Hybrid PSO-ACO	72

Figure compare the performance of the verification results with feature selection(277 features)using hybrid PSO-ACO and without performing feature selection.

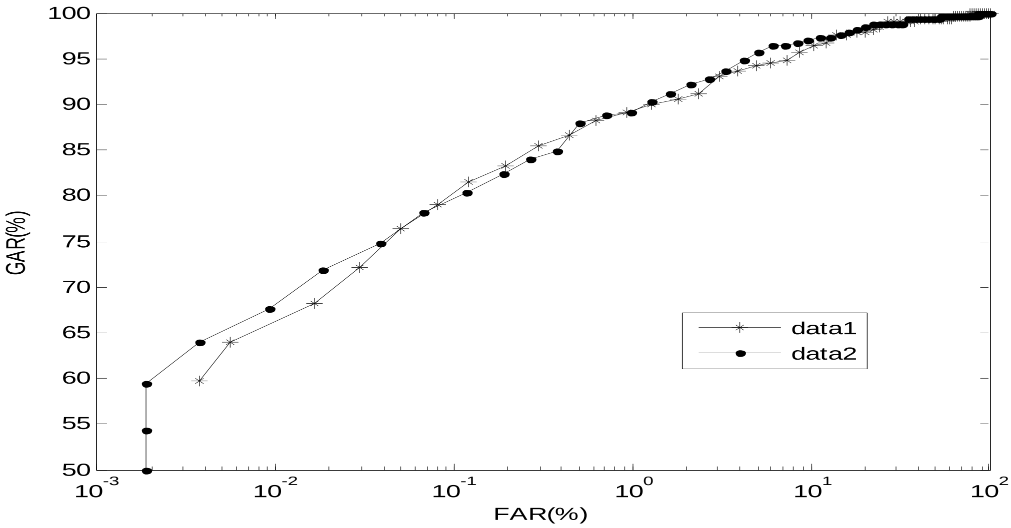

Fig.ROC

The performance of the whole feature set, which consist of 531features, is indicated by the horizontal dash-dotted line in the figure. The PSO-ACO achieved with 4% improvement in the performance using 277 features only.

VI. CONCLUSION

In this work a feature selection search procedure based on the Hybrid PSO-ACO is given.The proposed algorithm utilizes both the local importance of features and overall performance of the subsets to search through the feature space for optimal solutions.The implementation of the proposed approach on leftIndex Knuckle gives a reduction of 47% features with 4% improvement in the performance.In coparison with PSO however not much improvement in the performance but it converges faster than PSO computation time is less for the reduction of same number of features.

VII. FUTURE WORK

This approach will be improved using reinforcement to show better performance over PSO and ACO.Try to hybridized ACO with PSO i.e. updation of trail by applying PSO.

To test this hybrid approach other classification problems will be considered in the future.

This feature selection approach will be implementaed to select the features from concatenated feature vectors originated from different biometric modalities .

REFERENCES

[1] Ioan Cristian Trelea,The Particle Swarm Optimization algorithm:Convergence analysis and parameter selection, SAMS Publishers, Indiana USA.

[2] Shelokar P.S., Siarry P., Jayaraman V.K., Kulkarni B.D., Particle Swarm and Ant Colony algorithms hybridized for improved continuous optimization.Comput Mech 1994; 14:277–87.

[3] Shuang B., Chen J., Li Z., Study on Hybrid PS-ACO algorithm , M.Tech. Thesis, IIT Delhi.

[4] Hamidreza Rashidy Kanan, Karim Faez and Mehdi Hosseinzadeh, Face Recognition System Using Ant ColonyOptimization-Based Selected Features,CISDA 2007,

[5] Ahmed Al-Ani,Ant colony optimization for feature subset selection,World academy of science.

[6] R. Raghavendra , BernadetteDorizzi , AshokRao , G.HemanthaKumar , Designing efficient fusion schemes for multimodal biometric systems using face and palmprint , Pattern Recognition 44 (2011) 1076–1088