Multiple Features Subset Selection using Meta-Heuristic Function
L. S. Maurya
Information Technology Department
SRMSCET, Bareilly (U.P.) India
ismaurya@yahoo.com

Susheel Jain
M.TECH. Scholar (Software Engineering)
SRMSCET, Bareilly (U.P.) India
Jain_susheel65@yahoo.co.in

ABSTRACT
This paper is being presented on Multiple Features Subset Selection Using Meta-heuristic Function. Classification problems require selection of a subset of attributes or features from a much larger dataset to represent the patterns to be classified. Many excellent multiple feature selection method such as Hill Climbing (HC), Simulated Annealing (SA), Genetic Algorithms (GAs), Tabu Search (TS) has been prevalent amongst research community. However, these approaches still encounter when the multiple feature of dataset is available and it need to choose those attribute which is best amongst the available features.

So, in this paper basically considering the issue of multiple features are analyzed as well as implemented and tested on different dataset. The experiment is being conducted abalone dataset. It may conclude that the algorithm shows highest accuracy amongst all other method which is being used in this paper, Therefore we are focusing on the classification of multiple features.

1. Introduction
Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. These tools can include statistical models, mathematical algorithms, and machine learning methods (algorithms that improve their performance automatically through experience, such as neural networks or decision trees). Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction [1].

Two common data mining techniques [2], [3] for finding hidden patterns in data are clustering and classification analyses. There are a variety of algorithms used for clustering, but they all share the property of iteratively assigning records to a cluster, calculating a measure (usually similarity, and/or distinctiveness), and re-assigning records to clusters until the calculated measures don't change much indicating that the process has converged to stable segments. Classification is a different technique than clustering. Classification is similar to clustering in that it also segments customer records into distinct segments called classes. But unlike clustering, a classification analysis requires that the end-user/analyst know ahead of time how classes are defined.

In the process of classification [3], it finds that the normal classification algorithm is very slow in term of processing time as well as the classification rate is very low. Now these are applying some existing methodology for the improvement of classification of K.N.N algorithm such as MFS [4], MDF [5] and FC-MMNC [6]. But all these methods of classification not up to the mark for the different-different dataset to increase accuracy and decrease errors, Such as abalone, iris, cancer, aids, and internet advertisement. Because classification required a search space for better classification of multiple feature dataset. Now the proposed a new technique for multiple feature classification of data. For this work adopt meta-heuristic algorithm such ant colony optimization with multiple feature subset classification in order to improve accuracy of classification.

This paper is organized in such a manner that section 2 describes existing Classification techniques
followed by MFS, MDF and FC-MMNC and heuristic and meta-heuristic algorithms in section 3. In section 4 and 5 define implementation environment and methods is described with results and conclusion of the paper describe in section 6.

2. EXISTING CLASSIFICATION TECHNIQUES

2.1 THE MFS ALGORITHM

This technique proposed by Stephen D. Bays [4] for classification from multiple feature subsets (MFS) which combines multiple $k$-NN classifiers. The algorithm for nearest neighbor classification from multiple feature subsets (MFS) is simple and can be stated as:

Using simple voting, combine the outputs from multiple NN classifiers, each having access only to a random subsets of features. It selects the random subset of features by sampling from the original set. We use two different sampling functions: sampling with replacement, and sampling without replacement. In sampling with replacement, a feature can be selected more than once which is equivalent to increasing its weight.

Each of the NN classifiers uses the same number of features. This is a parameter of the algorithm which we set by cross-validation performance. Each time a pattern is presented for classification, we select a new random subset of features for each classifier.

2.2 FC-MNNC ALGORITHM

FC-MNNC (Feature subset Clustering for Multiple Nearest Neighbor Classifiers) proposed by Li-Juan Wang, Xiao-Long Wang [6] which was based on feature subset clustering for combining multiple $k$-NN to obtain better performance than that using a single $k$-NN. Below given figure demonstrated FC-MNNC system.

This method uses the concept of feature subset clustering which is used to improve the combination performance of multiple classifiers. Feature subset selection is similar to feature selection, both of which use a part of feature to classify unknown patterns. Feature subset clustering can be used to improve the combination performance of multiple classifiers, where the feature set is clustered into different feature subsets, not discarding the irrelevant features. Because feature selection cannot deal with multiple decisions, it only uses a part of features and discards the irrelevant features to classify. In FC-MNNC method, GA is used for clustering features to form different feature subsets according to the combination classification accuracy. The final decision is aggregated by majority voting rule [7], which is a simple and efficient technique.

2.3 MDF Algorithm:

MDF method proposed by Takahiro Yamada, Kyohei Yamashita [5] to combine $k$-NN classifiers based on different distance functions with weights.

Below Given figure shows the basic framework of the proposed new system. We call this system $k$-nearest neighbor classification by combining multiple different distance functions (D$k$NN). First, the D$k$NN inputs several distance functions. Then, it uses each distance function to generate $k$ nearest samples in the training data. It combines the all $k$ nearest samples and determines the class of unknown object based on the simple voting.
3. HEURISTIC AND METAHEURISTIC ALGORITHMS:

Heuristic algorithms are usually experience-based, with no specific pre-defined rules to apply. Simply, they are a ‘common-sense’ approach to problem solving. On the other hand, an important subclass of heuristics is meta-heuristic algorithms, which are general purpose frameworks that can be applied to solve wide range of problems, with just minor changes to the basic algorithm definition. Many meta-heuristic techniques try to mimic biological, physical or natural phenomena drawn from the real-world.

Combinatorial optimization (CO) problems can generally be defined as problems that require searching for the best solution among a large number of finite discrete candidate solutions. In solving a problem using a heuristic or a meta-heuristic algorithm, the search for a good problem solution is usually divided into two phases: solution construction and solution improvement. Solution construction refers to the process of creating one or more initial feasible solutions that will act as a starting point from which the search progresses. In this phase, a construction heuristic usually starts from an empty solution and gradually adds solution components to the partially constructed solution until a feasible solution is generated. On the other hand, solution improvement tries to gradually modify the starting solution(s), based on some predefined metric, until a certain solution quality is obtained or a given amount of time has passed.

Figure 3.1 search space for a minimization function

Within the context of searching for a good quality solution, we usually use the term search space to refer to the state space of all feasible solutions/states that are reachable from the current solution/state. To find a good quality solution, a heuristic or a meta-heuristic algorithm moves from one state to another, through a process that is often called Local Search (LS). During LS, the new solution is usually generated within the neighborhood of the previous solution. A neighborhood N(x) of a solution x is a subset of the search space that contains one or more local optima, the best solutions in this neighborhood. The transition process from one candidate solution to another within its neighborhood requires a neighborhood move that changes some attributes of the current solution to transform it to a new solution x’. A cost function f(x) is used to evaluate each candidate solution x and determine its cost compared to other solutions in its neighborhood. The best solution within the overall search space is called the globally optimal solution, or simply the optimum solution. Figure shows a search space for a minimization function, i.e., our goal is to obtain the global minimum for this function. In this figure three points (solutions) are local optima within their neighborhoods, A, B and C. The global optimum among the three points is point C. The most famous meta-heuristic techniques are: Hill Climbing (HC), Simulated Annealing (SA) [8], Genetic Algorithms (GAs) [11], Ant Colony Optimization (ACO) [12] and Tabu Search [10].

4. PROPOSED WORK

Ant Colony Optimization (ACO) it is 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 overall process of ACO feature selection can be seen in figure.

For a given classification task, the problem of feature selection can be stated as follows: given the original set, F, of n features, find subset S, which consists of m features (m < n, S ⊆ F), such that the classification accuracy is maximized.

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

- n features that constitute the original set,\n  \[ F = \{ f_1, \ldots, f_n \} \]
- A number of artificial ants to search through the feature space (na ants),\n- \( \tau_i \), 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 \( S_j = \{ S_1, \ldots, S_n \} \)
We propose to use a hybrid evaluation measure that is able to estimate the overall performance of subsets as well as the local importance of features. A classification algorithm is used to estimate the performance of subsets (i.e., wrapper evaluation function). On the other hand, the local importance of a given feature is measured using the correlation based evaluation function, which is a filter evaluation function.

Figure 4.1 Architecture of ACO for feature selection

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 trail 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 feature 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:

\[
USM_{ij}^{S_j} = \begin{cases} 
(\tau_i)^{\alpha} (LI_i^{S_j})^\beta & \text{if } i \notin S_j \\
\sum_{i \in S_j} (\tau_i)^{\alpha} (LI_i^{S_j})^\beta & \text{otherwise}
\end{cases}
\]

Where \( LI_i^{S_j} \) is the local importance of feature \( f_i \) given the subset \( S_j \) the parameters \( \alpha \) and \( \beta \) control the effect of pheromone trail intensity and local feature importance respectively. \( LI_i^{S_j} \) is measured using the correlation measure and defined as:

\[
LI_i^{S_j} = \frac{|C_{IR}|}{\sum_{f \in S_j} |C_{IS}|}
\]

Where \( C_{IR} \) is the absolute value of the correlation between feature \( i \) \( (f_i) \) and the response (class) variable \( R \), and \( C_{IS} \) is the absolute value of the inter-correlation between feature \( i \) \( (f_i) \) and feature \( s \) \( (f_s) \) that belongs to \( S_j \).

Below are the steps of the algorithm:

1. Initialization:
   - Set \( t_i = CC \) and \( \Delta T_i = 0, (i=1\ldots n) \) where \( CC \) is a constant and \( \Delta T_i \) is the amount of change of pheromone trail quantity for feature \( f_i \).
   - Define the maximum number of iterations.
   - 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 that each ant will start with in the second and following iterations.

2. If in the first iteration,
   - For \( j = 1 \) to \( na \),
     - Randomly assigns a subset of \( m \) features to \( S_j \).
     - Go to step 4.

3. Select the remaining \( p \) features for each ant:
   - For \( mm = m - p + 1 \) to \( m \),
   - For \( j = 1 \) to \( na \),
   - Given subset \( S_j \), choose feature \( f_i \) that maximizes \( USM_{ij}^{S_j} \)
     - \( S_j = S_j \cup \{ f_i \} \) Replace the duplicated subsets, if any, with randomly chosen subsets.

4. Evaluate the selected subset of each ant using a chosen classification algorithm:
   - For \( j = 1 \) to \( na \).
Estimate the Error (Ej) of the classification results obtained by classifying the features of $S_j$.

5. using the feature subsets of the best k ants, update the pheromone trail intensity:

- For $j = 1$ to $k$, /* update the pheromone trails */

$$\Delta \tau_i = \begin{cases} 
\max_{g=1:k\setminus j} (\tau_i - E_g) & \text{if } f_i \in S_j \\
0 & \text{otherwise}
\end{cases}$$

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

6. If the number of iterations is less than the maximum number of iterations, or the desired E has not been achieved, initialize the subsets for next iteration and go to step 3

- For $j = 1$ to $n_a$,
  - From the features of the best $k$ ants, randomly produce $m - p$ feature subset for ant $j$, to be used in the next iteration, and store it in $Sj$.

5. IMPLEMENTATION DETAIL:

MATLAB: MATLAB [13] stand for matrix laboratory is a numerical computing environment and fourth-generation programming language. Developed by Math Works, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and Fortran. I have used Matlab 7.8 .0.347(R2009A) version for implementation of algorithm.

DATASET: Abalone dataset [14] used for result analysis

Abalone dataset detail: The dataset is comes from an original (non-machine-learning) study: Warwick J Nash, Tracy L Sellers “The Population Biology of Abalone in Tasmania” (1994). In our experiment we have used seven continuous attributes of this dataset.

RESULTS: After running different methods on different threshold values we found that our ant colony optimization algorithm outperforms than other methods on same threshold values that is given below in table.

<table>
<thead>
<tr>
<th>THRESHOLD VALUE</th>
<th>METH</th>
<th>KNN By (MFS)</th>
<th>KNN By (MDF)</th>
<th>FC-MMNC</th>
<th>KNN with ACO</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td></td>
<td>77.268</td>
<td>77.658</td>
<td>66.850</td>
<td>88.631</td>
</tr>
<tr>
<td>0.40</td>
<td></td>
<td>70.612</td>
<td>80.110</td>
<td>74.912</td>
<td>95.778</td>
</tr>
<tr>
<td>0.55</td>
<td></td>
<td>77.658</td>
<td>83.180</td>
<td>97.371</td>
<td>97.371</td>
</tr>
<tr>
<td>0.70</td>
<td></td>
<td>78.636</td>
<td>82.88</td>
<td>95.08</td>
<td>96.19</td>
</tr>
<tr>
<td>0.92</td>
<td></td>
<td>85.12</td>
<td>86.26</td>
<td>89.27</td>
<td>96.19</td>
</tr>
</tbody>
</table>

Table 5.1 Average accuracy table based on different threshold values

Graph generates result for different used methods

6. CONCLUSION AND FUTUREWORK

In this paper, we presented a feature selection search procedure based on the Ant Colony Optimization meta-heuristic function. The proposed algorithm utilizes both local importance of features and overall performance of subsets to search through the feature space for optimal solutions. When used to select features in presented datasets, the proposed algorithm outperformed other feature selection methods (MFS, MDF, and FC-MMNC). Results show this algorithm selects a good subset of features that are useful to common machine learning algorithms by improving
their accuracy and making their results easier to understand especially in data sets with irrelevant or redundant features. It is obvious that in feature selection, improvement in correct classification rate depends on the correlation between features, and hence depends on data set. Therefore in data sets with uncorrelated features and without irrelevant features, feature selection may be result decreasing of correct classification rate. Other advantage of algorithm is that it scales only with the number of features. Therefore, does not require extra computation cost if the number of the data points in a dataset increases. Now the problem with algorithm is more computation time so we can enhance the given algorithm by using different methods With the help of new method we can improve feature subset selection problem.

References: