Improved UTA Feature Selection using Ant Colony Optimization

Mohammad Shokouhifar*1, Fardad Farokhi2, Amir Hossein Abdollahi3, Gholam Hasan Sajedy Abkenar4  
1,3,4Member of Scientific Association of Electrical & Electronic Engineering  
2Assistant Professor of Electrical Engineering Department  
Islamic Azad University of Central Tehran Branch, Tehran, Iran  
Corresponding Author Email: m_shokouhifar@ieee.org

Abstract—Feature Selection is the problem of choosing a minimal subset from the set of all features which are sufficient and necessary for classifier. UTA [1] is a simple algorithm performed in the trained artificial neural network. UTA evaluates the features according to their accuracy by removing them one by one. This algorithm classifies features into three categories: relevant, irrelevant and redundant features. UTA can guarantees that all of the relevant features are useful, but the disadvantage of UTA is that all correlated features will be determined as irrelevant/redundant features; because they are evaluated one by one. But in fact some of them may be relevant features. Ant colony optimization (ACO) is widely used for feature selection, and has very good performance; but it needs to a reasonable running time. In this paper at the beginning a UTA algorithm is performed, and then an ACO is used for finding those useful features which UTA could not find them. Proposed algorithm (called UTAACO) efficiently improved the performance of UTA, and reduced the computational time of ACO. Obtained results indicate the robustness of UTAACO.

Keywords—feature selection; UTA algorithm; ant colony optimization (ACO); relevant features;

I. INTRODUCTION

Feature Selection (FS) is the problem of selecting a minimal subset from the set of all original features that achieves the best performance in terms of accuracy and the needed running time [2]. It is achieved by removing the irrelevant, redundant and noisy features [3]. FS spreads throughout many fields, include signal processing [4], face recognition problem [5], text categorization [6]. In recent years, many FS approaches have been proposed. There are two key subjects in constructing a FS algorithm: search strategies and evaluation measures.

With respect to search strategies, exhaustive [7], heuristic [8], and random [9] search strategies were proposed. Given a feature set with size \( N \), the task of FS is a search for an optimal feature subset through the challenging \( 2^N \) candidate subsets. The designation of what an optimal subset will be achieved, is depending on the given problem to be solved.

In exhaustive search, all possible subsets are evaluated and the best one is selected as optimal feature subset. Although an exhaustive search may be used for this purpose, it is quite impractical for most data sets. Usually FS algorithms use heuristic or random search strategies in an attempt to avoid this exorbitant complexity. However, the degree of the optimality of the final selected feature subset by heuristic or random search is often reduced [10].

And with respect to evaluation measures, the FS methods can be divided into two main categories: classifiers-specific (so-called wrapper approaches) [11]-[15], and classifier-independent (so-called filter approaches) [16]-[19]. The former applies a learning algorithm to evaluate the significance of selected feature subsets based on their classification accuracies or contribution to the classification boundary. While the latter constructs a classifier independent measure to evaluate the goodness of the generated feature subsets, such as inter-class distance [16], dependence [17], consistency [18], and mutual information [19].

Wrapper methods have good solution quality in terms of classification accuracy and length of selected subset, but they are too time-consuming. On the other hand filter approach reduces the computational time, while wrapper approach improves the solution quality. The common disadvantage of filter approaches is that they ignore the interaction with the learning algorithm. Wrapper methods are computationally more complex, but take the dependency of the classifier on the feature subset into account [20].

Evolutionary algorithms belong to the random search category. Among too many methods which proposed for FS, population-based evolutionary algorithms such as genetic algorithm [21], and ant colony optimization (ACO) [22]-[24] have attracted a lot of attention. These approaches attempt to achieve better solutions by knowledge from the previous iterations. In [22], an ACO-based methodology was applied to select the effective features for speech classification problem. In [23], a new rough set method based on ACO was proposed, which adopts mutual information-based feature significance as heuristic information. In [24], in the each iteration of ACO algorithm, the classification accuracy and the number of selected features is used for evaluation of the feature subsets represented by ants.

A different approach for FS is based on the examination of the parameters of an artificial neural network which is trained by using all original features. Setiono and Lio [25] have performed FS in a trained feed-forward neural network. In the each iteration, all of the weights of connections associated with a specific feature were set to zero, and the amount of reduction of accuracy was measured. Then the worst feature was removed and the neural network was retrained in following iteration.

Utans [1], also has proposed a FS approach named UTA using a trained neural network. In UTA approach, after training the network, in the each iteration the average of one feature in all instances is calculated, and then the value of the selected feature in all instances has been replaced by the
calculated mean value. Then the trained network is tested on the new dataset with the new replaced feature. If the accuracy is decreased, it means that the replaced feature was relevant, but if the accuracy did not change or even improves, that feature is considered as ineffective and should be removed from the feature vector.

In this paper a hybrid approach for effective FS base on UTA algorithm and ACO (called UTAACO) is proposed. At the beginning a UTA algorithm is performed, and then an ACO algorithm is used for finding those useful features which UTA could not find them.

II. UTA FEATURE SELECTION ALGORITHM

As noted above, UTA is a simple FS algorithm which is based on trained artificial neural network and proposed by Utans [1]. In this algorithm, after training the network using all original features, the effect of substituting an input in all instances by its mean value on result were calculated. On the other hand, the average of one feature in all instances is calculated. Then the selected feature in all input vectors has been replaced by the calculated mean value. Then the trained network was tested in the new data set with the new replaced feature. The UTA error factor (UEF) for each feature is defined as:

\[ \text{UEF} = (\text{FP}(\text{new}) + \text{FN}(\text{new})) - (\text{FP}(\text{old}) + \text{FN}(\text{old})) \]  

Where FP(old) and FN(old) are respectively the false positive and false negative using the whole features. FP(new) and FN(new) are respectively the false positive and false negative when the \( i \)-th feature was replaced by the mean value. After calculating the UEFs for all features, the features will be classified into three categories:

1- If UEF is positive, it means that the \( i \)-th feature is a relevant feature.
2- If UEF is equal to zero, the \( i \)-th feature is an irrelevant feature with a probability.
3- If UEF is negative, the \( i \)-th feature is a redundant feature with a probability.

The multilayer perceptron (MLP) neural network [20] is used as classifier before performing the UTA algorithm. It consists of an input layer, an output layer, and one or more hidden layers. Each layer consists of multiple neurons. An artificial neuron is the smallest unit that constitutes the artificial neural network. A typical three-layered MLP structure which is used as classifier can be seen in Fig. 1.

![Figure 1](image)

Figure 1. The Structure of a typical three-layered MLP neural network used as classifier, for a data set with \( D \) features, \( K \) neuron in first hidden layer, \( L \) neuron in second layer, and \( M \) output neurons.

The UTA algorithm guarantees the relevant features, but the disadvantage of UTA is that it cannot guarantees that all of the considered irrelevant/redundant features are useless. For example, if there are some correlated features, the UTA discard all of them, because it evaluates the features one by one. While some of them may be relevant features. In this paper, we perform an ACO algorithm in order to find these relevant features among the unselected features by UTA.

III. ANT COLONY OPTIMIZATION ALGORITHM

Ant Colony Optimization (ACO) is a metaheuristic proposed by Dorigo [26], in order to solve combinational optimization problems. Ants can find the shortest paths between their nest and food sources by marking the paths with a chemical substance called pheromone that they deposit on the ground when traveling their way. When ants arrive at a decision point, they have a probabilistic choice based on the intensity of pheromone that they smell. Finally when they want to come back to the nest, the probability of choosing the same path is higher (due to increase the pheromone amount on that path).

The deposition of pheromone is the main factor in enabling real ants to find the shortest paths over a period of time. Each ant probabilistically prefers to follow a direction rich in the pheromone. The pheromone decays over time, resulting in much less pheromone on less popular paths [27]. Given that over time the shortest path will have the higher rate of ant traversal. So this path will be reinforced and the others diminished until all ants follow the same, shortest path. Based on this idea, artificial ants can be deployed to solve complex optimization problems via the use of artificial pheromone deposition. Additionally, if a sudden change to the environment occurs (e.g. a large obstacle appears on the shortest path), the ACO system can respond to this and will eventually converge to a new solution.

As shown in Fig. 2, two paths are given that connect two nodes. At the first, the amount of pheromone on these two paths are equalized (a). So there is an equal chance to choose between these two paths. Therefore, the same number of ants will move through them. However, it takes longer for the ants that choose the longest path, causing more evaporation of the pheromone which they deposited. Typically, the pheromone deposit rule is considered as to penalize the positive reinforcement in longer paths. In both cases, the pheromone level of the longer path decreases in relation to the pheromone level of the shorter one (b). So after some iteration and over time, the shorter path predominates (c).

![Figure 2](image)

Figure 2. The pheromone evaporation and deposition on a simple problem with a decision point, and two different paths: Over time, the pheromone amount on the shortest path is greater, and almost all ants traverse on it.
The basic idea of ACO algorithm is to have a population of artificial ants that cyclically construct the solutions for the combinatorial optimization problem. In which the ants move every branch from one node to another node and so construct paths representing solutions. Starting in an initial node, every ant chooses the next node in its path according to transition rule. At each step, every ant computes a set of feasible expansions to its current partial solution and selects one of these depending upon two factors: local heuristics and previous knowledge. ACO is particularly attractive for FS as there seems to be no heuristic that can guide search process to find the optimal minimal subset every time. Additionally, it can be the case that ants discover the best feature subsets as they proceed throughout the search space.

IV. METHODOLOGY

The basic procedure of UTAACO is as follow: At the first, UTA algorithm is performed for all original features. So the features are classified into three categories according to the results of the UTA: relevant, irrelevant, and redundant. As noted in section 2, UTA guarantees the selected relevant features, but it cannot ensure that all irrelevant/redundant features are useless. So we apply an ACO algorithm in order to find those useful features that UTA could not find them. Given a colony of artificial ants to search through the search space consist of unselected features by UTA. These ants perform a number of iterations. During every iteration, each ant constructs its solution. The pheromone is updated iteratively, and the algorithm would be stopped when a termination condition will be satisfied.

A. Graph representation for FS problem

The FS problem must be reformulated into an ACO-suitable problem. The idea of ACO to solve FS problem is to model it as the search for a minimum cost path in a graph. As seen in Fig. 3, the nodes represent those unselected features by UTA algorithm, with the edges between them denoting the choice of the next feature by the ant. Based on this reformulation of the graph representation, the transition rules and pheromone update rules of standard ACO algorithms can be applied. In this paper, the pheromone and heuristic value are not associated with the links between the features. Instead, each feature in the graph (the unselected features by UTA algorithm) has its own pheromone value and heuristic value, aims to reduce the complexity of search space.

![Graph representation for FS problem](image)

**Figure 3.** An illustrative process of constructing a solution, where the condition feature set through those unselected features by UTA algorithm, is \( \{a_1,a_2,a_3,a_4\} \), and the constructed solution by the ant is \( \{a_1,a_2,a_4,a_6\} \).

B. Construct feasible solutions by ants

Each ant selects some features among the unselected features by UTA algorithm. These features are added to the preselected relevant features by UTA and then the ant constructs a solution. The probability of ant \( \pi \) choosing feature \( i \) in its subset at time \( t \), is:

\[
P_i^\pi(t) = \frac{\tau_i^{\alpha} \cdot \eta_i^{\beta}(t)}{\sum \tau_i^{\alpha} \cdot \eta_i^{\beta}(t)}
\]

Where, \( t \) and \( k \) denote the number of iterations and ants respectively. Note that \( i \) and \( l \) belong to the set of conditional features that did not selected by UTA. \( \tau_i \) and \( \eta_i \) are the pheromone value and the heuristic information of choosing feature \( i \). In addition \( \alpha \) and \( \beta \) are two constant parameters in the range of 0–1 that determine the relative importance of the pheromone trail and heuristic information.

C. Heuristic information

For most methods which used ACO to solve optimization problems, such as traveling salesman problem, the heuristic information is available before the construction of solutions. However, in UTAACO the heuristic information is dynamically calculated during the construction process of solutions. The significance of selected feature subsets is adopted as heuristic information for UTAACO. The significance of subsets is defined by their predictive accuracy and the number of features within them. After all ants construct their solutions and generate their subsets, they are evaluated by the cost function defined as:

\[
Cost_k = \varphi \times E_k + (1-\varphi) \times E_k \times \frac{L_k}{N}
\]

Where \( N \) is the number of all unselected features by UTA, and \( L_k \) is the number of features selected by ant \( k \). The \( E_k \) is the classification error rate of the subset consist of both the preselected features by UTA, and the selected features by ant \( k \). \( \varphi \) is a constant that adjusts the relative importance between the classification error rate and the number of selected features, and also adjusts the number of needed features that we want to add them with UTA preselected features. There is \( E_k \) both in left and right terms, because of normalize the effect of classification error rate and the number of selected features.

D. Pheromone updating

After all ants have constructed their solutions, they are evaluated by the Eq. 3, and the best ant is considered. After that, the pheromone trails on each node (each feature) should be updated according to the Eq. 4,5:

\[
\tau_i(t+1) = \tau_i(t) + \Delta \tau_i
\]

\[
\Delta \tau_i = -\lambda \tau_i(t) + \begin{cases} \frac{Q}{L_k} \\ 0 \end{cases} \quad \text{for nodes of the best ant}
\]

Where \( Q \) is a user-defined constant that adjusts the amount of deposited pheromone on the associated nodes of
the best ant. The $L_k$ is the length of the selected subset by the best ant. $\lambda$ ($0 < \lambda < 1$) is representing the evaporation of pheromone trails for all nodes: If $\lambda = 0$, the ACO algorithm cannot converge at all, and if a high value was considered for it, the algorithm converge very soon but usually in a non-optimal solution. The optimal value for $\lambda$ should be determined by some experiments. Note that in Eq. 4, a negative reinforcement occurs by the pheromone evaporation on all nodes, and a positive reinforcement occurs by the deposition of pheromone on the nodes of the best solution.

The ACO algorithm would be stopped when suitably high classification accuracy has been achieved with this subset, and the length of this subset must be minimal. Else the pheromone is updated, a new set of ants are created and the process iterates once more. The overall process of FS based on UTASFS can be seen in Fig. 4.

![Figure 4](image)

Figure 4. The overall flow chart of the UTAAACO algorithm.

V. EXPERIMENTAL RESULTS

The proposed FS approach (UTAACO), is implemented in MATLAB R2008a on an Intel PC with 2.53GHZ processor and 4GB memory running on windows vista, to test the performance. The number of ants was set to 30, and the max-iterations determined 100. Determination of the value of pheromone evaporation factor ($\lambda$) is very important and strongly affects the performance of ACO. If $\lambda = 0$, the ACO algorithm cannot converge at all. And if $\lambda$ set at high value, it converges quickly but in a sub-optimal solution. Results show that the best value for $\lambda$ is 0.06. In our experiments, we set $\alpha = 1$, $\beta = 0$, $Q = 0.1$, $\varphi = 0.22$. To evaluate the feature subsets generated by ants, an MLP neural network was used as classifier. In each dataset 40% of entire data set was used for training and the remaining was used for test.

This paper is concluded by presenting experimental results using UTAAACO on various datasets provided by UCI Machine Learning Repository which is regularly used in papers, included Audiology, Breast Cancer, Mushroom, Vote, Wine and Zoo. We also compare the performance of UTAAACO with UTA algorithm [1], and also with an ACO-based wrapper algorithm that given in [24]. Results of FS are given in Tables 1 and 2. In Table 1, the “Original” column denotes the number of all original features. The “UTA”, “ACONN”, and “UTAACO” column respectively denotes the feature numbers after running the three algorithms. Classification accuracy in the data sets consists of all original features, and also consists of selected features by the three algorithms respectively given in Table 2.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Audiology</td>
<td>70</td>
<td>22</td>
<td>19</td>
<td>25</td>
</tr>
<tr>
<td>B. Cancer</td>
<td>9</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Mushroom</td>
<td>23</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Vote</td>
<td>17</td>
<td>10</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>Wine</td>
<td>14</td>
<td>5</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Zoo</td>
<td>17</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

TABLE II. CLASSIFICATION ACCURACY OF THE THREE ALGORITHMS.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Audiology</td>
<td>94.5%</td>
<td>94.0%</td>
<td>97.0%</td>
<td>96.5%</td>
</tr>
<tr>
<td>B. Cancer</td>
<td>95.2%</td>
<td>95.9%</td>
<td>94.9%</td>
<td>95.9%</td>
</tr>
<tr>
<td>Mushroom</td>
<td>98.1%</td>
<td>99.2%</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Vote</td>
<td>95.3%</td>
<td>94.4%</td>
<td>92.5%</td>
<td>95.1%</td>
</tr>
<tr>
<td>Wine</td>
<td>96.4%</td>
<td>96.4%</td>
<td>94.4%</td>
<td>96.8%</td>
</tr>
<tr>
<td>Zoo</td>
<td>94.1%</td>
<td>94.0%</td>
<td>92.9%</td>
<td>93.4%</td>
</tr>
</tbody>
</table>

TABLE III. RUNNING TIME IN THREE ALGORITHMS (IN MINUTES).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Audiology</td>
<td>4.7</td>
<td>345</td>
<td>43</td>
</tr>
<tr>
<td>B. Cancer</td>
<td>3.2</td>
<td>230</td>
<td>32</td>
</tr>
<tr>
<td>Mushroom</td>
<td>21.5</td>
<td>975</td>
<td>187</td>
</tr>
<tr>
<td>Vote</td>
<td>3.1</td>
<td>234</td>
<td>27</td>
</tr>
<tr>
<td>Wine</td>
<td>2.8</td>
<td>185</td>
<td>30</td>
</tr>
<tr>
<td>Zoo</td>
<td>2.5</td>
<td>174</td>
<td>22</td>
</tr>
</tbody>
</table>

UTA is a simple and fast algorithm, but ACO needs to a more reasonable time for effective FS. As seen in Table 3, UTAAACO reduced the needed running time of ACO.
VI. CONCLUSION

This paper discussed the shortcoming of UTA method, which is a simple and very fast FS algorithm. UTA algorithm guarantees that all considered relevant features are useful, but it cannot ensure that considered irrelevant/redundant features are useless. For example, if there are some correlated features, the UTA discard all of them, because it evaluates the features one by one. While some of them may be relevant features. In this paper, we perform an ACO algorithm after using UTA, in order to find these relevant features among the unselected features by UTA. We proposed a hybrid FS technique based on ACO and UTA algorithm. The proposed algorithm (UTAACO) has the following advantage: (a) It has both advantage of UTA algorithm and ACO technique for FS; (b) It improved the performance of UTA method by considering those features that were omitted mistakenly (c) It reduced the running time of ACO algorithm by decreasing of feature space.

REFERENCES


