CHAPTER 4

MULTI OBJECTIVE FUNCTION BASED ANT COLONY OPTIMIZATION (MOFACO) TECHNIQUE

4.1 INTRODUCTION

Opinion classification involves classifying the opinionated review text into two forms as positive or negative sentiment reviews. Machine learning is used for text classification with different classification algorithms such as Decision Tree, Naive Bayes, and SVM and so on. This approach is applied for review text classification of data such as movie, product, medical reviews and news reviews. Sentiment analysis is also done by the other methods such as dictionaries, word lexicons, word senses and so on (Buche et al., 2013).

Multi objective function is better because the feature selection in machine learning is a global optimization problem that reduces feature number, removes irrelevant, noisy and redundant data, resulting in recognition accuracy. It is an important step that affects the system performance of pattern recognition. Typically, feature selection problems are solved through the use of single objective optimization techniques like genetic algorithm. This type of technique optimizes only a single quality measure; for example, recall, precision or F-measure at a time. Sometimes, it is unable to capture a good classifier’s quality reliably by a single measure. A good quality classifier must have recall, precision and F-measure values optimized simultaneously instead of doing parameter values alone. And also swarm intelligence technique is
used widely to address the selection of optimal feature set (Gupta & Joshi 2016).

In customer review documents, reviewers convey positive, negative or both the sentiments about the objects and attributes. Document level and sentence level classification would not express the likes and dislikes of consumers about particular attributes of object. For example, if users comment on a mobile phone, they basically will comment upon Camera result, LCD size, speaker, weight and so on. For example, on camera output, 125 comments express the positive opinions and 25 comments may be negative. If a new customer is interested in camera quality of mobile, he or she can take decisions easily as to purchase the mobile or not. To explore the detailed opinion on a product or any topic or service, a detailed opinion mining study is required, which is called feature based opinion mining.

Classifying the entire documents according to the opinions towards certain objects is called sentiment classification. One part of opinion mining in product reviews is generating feature based summary. To generate a summary on the features, product features are first identified, and aggregation of positive and negative opinions on them is done. Features retrieved are the product attribute components and other aspects of the products. Fitness function can be set to meet the objective of a particular measure.

In all the supervised methods, reasonably high accuracy can be obtained subject only to the requirement that the test data should be similar to training data. Moving a supervised sentiment classifier to another area/domain would require collecting annotated data in the new domain and retraining the classifier. This dependency on annotated training data is the most important shortcoming of all the supervised methods and it can be applied for different objectives in the fitness functions.
4.2 MATERIALS AND METHODS

In the current study, data from online movie reviews are collected, i.e., from IMDB dataset and the reviews from medical dataset then after removing stop words, features are extracted and weighed. Then the different versions of SVM classifier are applied. TF-IDF is a widely used feature selection technique for classifying the documents and semantic based feature selection Information Gain is used. Classification results are evaluated by the parameters such as classification accuracy, precision and recall. The flow chart of the proposed algorithm is given in Figure 4.1.
Stop words are commonly occurring words in a sentence and they are non-significant words that exist in a document. They should not be considered for creating indexes for a document or sentence. Stemming is a process of documenting similar tokens into a single type. Stemming procedure identifies and removes the prefixes, suffixes and unwanted plurals in a word to find the stem or root of a word (Savoy 1999).

Information Gain is a feature ranking method based on decision trees that reveals good classification performance. A threshold value of 75 is used for checking the features; if a feature has a greater IG value than the threshold, the feature is chosen; if not, it is not selected.

4.2.1 Ant Colony Optimization (ACO) for Feature Selection

The problem in feature selection lies in locating an optimal subset which is used to reduce computational overheads and to improve classification accuracy. A feature selection problem may be represented as an ACO-suitable problem by using the node for representing a feature and edge representing the cost function linking to the next feature in a graph. Ant colony algorithms are optimization techniques inspired by the foraging behavior of real ants in nature. When searching for food, ants primarily explore the area surrounding their nest in a arbitrary manner. When an ant finds a food source, it estimates the quantity and quality of the food and brings the food back to the nest. During return, the ant deposits a chemical pheromone trail on the ground. It is easy to understand that the quantity of pheromone deposited, would depend on the quantity and quality of the food, and this pheromone deposited, will guide other ants to the food source. The indirect contact between the ants through pheromone trails enables them to find the shortest paths between the nest and food sources. This special
characteristic of real ant colonies is exploited in artificial ant colonies in order to solve difficult combinatorial optimization problems. In ant colony algorithm, artificial ants probabilistically build solutions by taking into account dynamical artificial pheromone trails. The essential component of ACO algorithm is the pheromone model including the state transition rule and updating rule, which is used to probabilistically sample the search space (Blum & Dorigo 2004).

Each Ant represents a feature subset and denotes the path taken by the ant. An optimal subset of features is obtained while the ant traverses the graph and with visit to minimal number of nodes and at the same time, satisfying the termination criteria as shown in Figure 4.2.

Figure 4.2 Representation of ACO for feature selection.

Foods searching habit and intelligence in food searching activity of real ants have attracted many researchers to simulate the ant’s behavior in optimization problems. Whenever it searches for the food, an ant deposits a chemical substance called pheromone in the ground (Dorigo & Socha 2006). By smelling the pheromone, the other ants are able to choose their path. Each ant prefers a path with high pheromone trail, so that the other paths that
remain unused for a long time would be evaporated. The pseudo code for ACO is given below:

1: Initialize pheromone trail
2: while stopping criteria not met do
3: for all ants do
4: Deposit ant randomly
5: while solution incomplete do
6: Select next element randomly according to the pheromone trail
7: end while
8: end for
9: Update pheromone trail
10: end while

ACO is suitable for multi objective optimization problems and it has been applied for several applications. The combination of the classifier parameters, C and \( \gamma \), based on RBF kernel of the SVM classifier is represented by using ant’s solution. The classification accuracy of the SVM classifier is utilized to direct the updation of solution archives. Based on the solution archive, the transition probability is calculated to choose a solution path for an ant. The overall process to hybridize ACO and SVM is proposed (Alwan & Ku-Mahamud 2012).

In the current research work, the number of ants selected for ACO is equivalent to 15. Using ACO, the parameters of kernel functions are tuned. Termination condition is set to maximize the overall classification accuracy. SVM is implemented like SVM with poly kernel, SVM with RBF kernel (10,0.01), SVM with RBF kernel(100,0.1) and SVM with RBF kernel with ACO, MOFACO is utilized for better feature selection before applying the classification algorithms such as CART, Naïve Bayes and SVM. Finally ACO is used for parameter optimization.
The focus of ACO optimization algorithm is to generate reduced size of salient feature subsets. ACO feature selection uses hybrid search technique combining wrapper/filter approaches. It modifies standard pheromone update and heuristic information measurement rules based on these approaches. It emphasizes the selection of salient features, and attaining a reduced number. It chooses a reduced number’s salient features using a subset size determination scheme. Termination condition maximizes the overall classification accuracy.

An novel multi objective fitness function is proposed in this work and is given by the Equation (4.1)

\[
f(c) = \frac{e^{\delta-1}}{e^\delta - 1} \frac{1}{N_L}
\]

where
- \(N_L\) is the number of leftout features
- \(\delta\) is given by \(\text{wrongly classified instances} \div \text{Total number of training instances}\)
- \(e\) is 2.713

(4.1)

Exponential fitness function was selected for faster convergence. The basis for any ACO algorithm is a constructive heuristic for probabilistic solution creations. A constructive heuristic assembles solutions as elements sequences from a finite solution components set.

The basic thing for any ACO algorithm is a constructive heuristic for probabilistic solution creation. The function of a constructive heuristic is assembling solutions as element sequences from a finite solution component set. A solution construction begins with an empty partial solution, and at each production step, a current partial solution is prolonged by adding a feasible
solution component from the solution component set (Dorigo & Blum 2005). The probabilistic transition is represented by Equation (4.2).

\[
P_i^k(t) = \begin{cases} 
\frac{[\tau_i(t)]^\alpha [\eta_i]^\beta}{\sum_{i \in \mathcal{J}^k} [\tau_i(t)]^\alpha [\eta_i]^\beta} & \text{if } i \in \mathcal{J}^k, \\
0 & \text{otherwise}, 
\end{cases} 
\]  

(4.2)

where \( \mathcal{J}^k \) represents a set of feasible features that are added to a partial solution; \( \tau_i \) and \( \eta_i \) are pheromone value and heuristic desirability related to feature \( i \) respectively. \( \alpha \) and \( \beta \) are two arguments determining pheromone value’s and heuristic information’s relative importance.

Transition probability used by ACO is a balance between pheromone intensity, \( \tau_i \) and heuristic information (expressing move desirability), \( \eta_i \). This balances an exploitation–exploration trade-off. The \( \alpha \) and \( \beta \) are used to achieve the best balance between exploitation/exploration. If \( \alpha = 0 \), then no pheromone information is used, i.e., earlier search experience is not considered. The search degrades to a stochastic greedy search. If \( \beta = 0 \), attractiveness (potential benefit) of moves is neglected.

Pheromone evaporation on nodes is triggered after ants reach their solutions, and according to Equation (4.3), each ant \( k \) deposits a quantity of pheromone, \( \Delta \tau_i^k(t) \), on a node that it has used,

\[
\Delta \tau_i^k(t) = \begin{cases} 
\phi \cdot \mathcal{J}(S^k(t)) + \frac{\varphi_i(n-|S^k(t)|)}{n} & \text{if } i \in S^k(t), \\
0 & \text{otherwise}, 
\end{cases} 
\]  

(4.3)
where $S^k(t)$ is a feature subset derived by ant $k$ at iteration $t$, and $|S^k(t)|$ is its length. It is assumed that the performance of the classifier is equally important compared to subset length (Nemati et al., 2009), and thus the values $\phi = 0.5$, $\phi = 0.5$ are chosen. Pheromone is added using Equation (4.4).

$$
\tau_i(t+1) = (1 - \rho)\tau_i(t) + \sum_{k=1}^{m} \Delta\tau_i(t) + \Delta\tau^f_i(t)
$$

(4.4)

where $m$ is the number of ants at each iteration and $\rho \in (0,1)$ is a pheromone trail decay coefficient. In this, $g$ indicates the best ant at each iteration.

### 4.2.2 Proposed Multi-Objective Function ACO (MOFACO) Method

In many real-life optimization problems there are several objectives to optimize. For such Multi-Objective Problems (MOP), there is not usually a single best solution but a set of solutions that are superior to others when considering all objectives. This set is called the Pareto set or non-dominated solutions. This multiplicity of solutions is explained by the fact that objectives are generally conflicting ones (Deb et al., 2014).

More formally, a multi-objective optimization problem is defined by a quadruplet $(X, D, C, F)$ such that $X$ is a vector of $n$ decision variables, i.e., $X = (x_1, \ldots, x_n)$; $D$ is a vector of $n$ value sets defining the domains of the decision variables, i.e., $D = (d_1, \ldots, d_n)$; $C$ is a set of constraints on $X$, i.e., a set of relations restricting the values that may be simultaneously assigned to the decision variables; and $F$ is a vector of $m \geq 2$ objective functions $F(X) = (f_1(X), f_2(X), \ldots, f_m(X))$; without loss of generality, it assume that these different objective functions have to be minimized (the functions having to be maximized may be multiplied by $-1$).
The generic algorithm, called MOF-ACO, is implicitly parameterized by the MOP (X, D, C, F) to be solved. It shall consider that ants build solutions within a construction graph \( G = (V, E) \) the definition of which depends on the problem to be solved, and that pheromone trails are associated with vertices and/or edges of this graph. It shall also assume that a heuristic information \( \eta^i \) is defined for every objective function \( f_i \in F \).

Algorithm MOF-ACO(#Col, #\tau)

1. Initialize all pheromone trails to \( \tau_{\text{max}} \)
2. repeat
   1. for each colony \( c \) in 1..#Col
      1. for each ant \( k \) in 1..nbAnts
         1. construct a solution
      2. for \( i \) in 1..#\tau
         2. update the \( i \)th pheromone structure trails
         3. if a trails is lower than \( \tau_{\text{min}} \) then set it to \( \tau_{\text{min}} \)
         4. if a trails is greater than \( \tau_{\text{max}} \) then set it to \( \tau_{\text{max}} \)
   3. until maximal number of cycles reached

MOF-ACO is also parameterized by the number of ant colonies \#Col and the number of considered pheromone structures \#\tau. Above algorithm describes the generic framework of MOF-ACO (#Col, #\tau). Basically, the algorithm follows the MAX-MIN ant system scheme. First, pheromone trails are initialized to a given upper bound \( \tau_{\text{max}} \). Then, at each cycle every ant constructs a solution, and pheromone trails are updated. To prevent premature convergence, pheromone trails are bounded within two given bounds \( \tau_{\text{min}} \) and \( \tau_{\text{max}} \) such that \( 0 < \tau_{\text{min}} < \tau_{\text{max}} \). The algorithm stops iterating when a maximum number of cycles has been performed (Leguizamón & Coello 2011).
Construction of a solution $S$:

$S \leftarrow \phi$

$Cand \leftarrow V$

while $Cand \neq \phi$ do

choose $v_i \in Cand$ with probability $p_{v_i}$

add $v_i$ at the end of $S$

remove from $Cand$ vertices that violate constraints

end while

Solution construction: Above algorithm describes the algorithm used by ants to construct solutions in a construction graph $G = (V, E)$ the definition of which depends on the problem $(X, D, C, F)$ to solve. At each iteration, a vertex of $G$ is chosen within a set of candidate vertices $Cand$; it is added to the solution $S$ and the set of candidate vertices is updated by removing vertices that violate constraints of $C$. The vertex $v_i$ to be added to the solution $S$ by ants of the colony $c$ is randomly chosen with the probability $p_{v_i}^c$ defined as (4.5):

$$p_{v_i}^c = \frac{[\tau_{v_i}^c]^{\alpha} [\eta_{v_i}^{c}]^{\beta}}{\sum_{v_j \in Cand} [\tau_{v_j}^c]^{\alpha} [\eta_{v_j}^{c}]^{\beta}}$$ (4.5)

Where $\tau_{v_i}^c$ and $\eta_{v_i}^{c}$ respectively are the pheromone and the heuristic factors of the candidate vertex $v_i$, and $\alpha$ and $\beta$ are two parameters that determine their relative importance (Alaya et al., 2007).

Pheromone update: Once all ants have constructed their solutions, pheromone trails are updated as usually in ACO algorithms: first, pheromone trails are reduced by a constant factor to simulate evaporation; then, some pheromone is laid on components of the best solution. More precisely, the quantity $\tau^i(c)$ of the $i$th pheromone structure laying on a component $c$ is updated as (4.6):
\[
\tau' (c) \leftarrow (1-\rho) \times \tau' (c) + \Delta \tau' (c) \quad (4.6)
\]

Where \( \rho \) is the evaporation factor, such that \( 0 \leq \rho \leq 1 \), and \( \Delta \tau' (c) \) is the amount of pheromone laid on the component \( c \).

### 4.3 EXPERIMENTATION AND RESULTS

Experiments are carried out using the IMDB movie dataset and medical opinions collected from various websites. The movie dataset obtained in the public domain consists of experts’ opinions which is treated as the class label. For the medical blogs collected, three experts have rated their opinions and the best answer is selected by voting. The features are selected using IG. The top ranked features are selected and classified using MOFACO-CART, MOFACO-Naïve Bayes, MOFACO-SVM-Poly, MOFACO-SVM RBF (10,0.01), MOFACO-SVM RBF (100,0.1) and MOFACO-SVM-ACO methods are used. Tables 4.1 & 4.2 and Figures 4.3 to 4.8 show the classification accuracy, recall and precision of IMDB dataset and medical blog dataset respectively.

#### 4.3.1 IMDB Dataset

Table 4.1 Results for IMDB Dataset using multi objective function ACO

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Classification Accuracy</th>
<th>Recall for positive</th>
<th>Recall for negative</th>
<th>Precision for positive</th>
<th>Precision for negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOFACO-CART</td>
<td>80.89</td>
<td>0.8267</td>
<td>0.7911</td>
<td>0.7983</td>
<td>0.8203</td>
</tr>
<tr>
<td>MOFACO-Naïve Bayes</td>
<td>81.89</td>
<td>0.8422</td>
<td>0.7956</td>
<td>0.8047</td>
<td>0.8345</td>
</tr>
<tr>
<td>MOFACO-SVM-Poly</td>
<td>82.33</td>
<td>0.84</td>
<td>0.8067</td>
<td>0.8129</td>
<td>0.8345</td>
</tr>
<tr>
<td>MOFACO-SVM RBF (10,0.01)</td>
<td>83.89</td>
<td>0.8578</td>
<td>0.82</td>
<td>0.8266</td>
<td>0.8522</td>
</tr>
<tr>
<td>MOFACO-SVM RBF (100,0.1)</td>
<td>88.67</td>
<td>0.8711</td>
<td>0.9022</td>
<td>0.8991</td>
<td>0.875</td>
</tr>
<tr>
<td>MOFACO-SVM-ACO</td>
<td>90.89</td>
<td>0.8978</td>
<td>0.92</td>
<td>0.9182</td>
<td>0.9</td>
</tr>
</tbody>
</table>
Figure 4.3 Classification Accuracy for IMDB Dataset using multi objective function ACO

Figure 4.3 reveals that the classification accuracy of MOFACO-SVM-ACO is better by 11.64% than MOFACO-CART, by 10.42% than MOFACO-Naïve Bayes, by 9.88% than MOFACO SVM-Poly, by 8.01% than MOFACO-SVM RBF (10,0.01), by 2.47% than MOFACO-SVM RBF (100,0.1) for IMDB dataset.

Figure 4.4 Recall for IMDB Dataset using multi objective function ACO
Figure 4.4 shows that the recall of MOFACO-SVM-ACO performs better by 8.24% & 15.06% than MOFACO-CART, by 6.39% & 14.5% than MOFACO-Naïve bayes, by 6.65% & 13.12% than MOFACO SVM-Poly, by 4.55% & 11.49% than MOFACO-SVM RBF (10,0.01) and by 3.01% & 1.95% than MOFACO-SVM RBF (100,0.1) when compared with the recall for positive and negative for IMDB dataset.

Figure 4.5 Precision for IMDB Dataset using multi objective function ACO

It can be observed from Figure 4.5 that the precision of MOFACO-SVM-ACO performs better by 13.97% & 9.26% than MOFACO-CART, by 13.17% & 7.55% than MOFACO-Naïve bayes, by 12.16% & 7.55% than MOFACO SVM-Poly, by 10.49% & 5.45% than MOFACO-SVM RBF (10,0.01) and by 2.1% & 2.81% than MOFACO-SVM RBF (100,0.1) the precision for positive and negative by using IMDB dataset.
4.3.2 Medical Opinion Dataset

Table 4.2 Results for Medical Opinion Dataset using multi objective function ACO

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Classification Accuracy</th>
<th>Recall for positive</th>
<th>Recall for negative</th>
<th>Precision for positive</th>
<th>Precision for negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOFACO-CART</td>
<td>81.22</td>
<td>0.8289</td>
<td>0.7956</td>
<td>0.8022</td>
<td>0.823</td>
</tr>
<tr>
<td>MOFACO-Naïve Bayes</td>
<td>81.44</td>
<td>0.8267</td>
<td>0.8022</td>
<td>0.8069</td>
<td>0.8223</td>
</tr>
<tr>
<td>MOFACO-SVM-Poly</td>
<td>82.89</td>
<td>0.8467</td>
<td>0.8111</td>
<td>0.8176</td>
<td>0.841</td>
</tr>
<tr>
<td>MOFACO-SVM RBF (10,0.01)</td>
<td>84.56</td>
<td>0.8644</td>
<td>0.8267</td>
<td>0.833</td>
<td>0.8591</td>
</tr>
<tr>
<td>MOFACO-SVM RBF (100,0.1)</td>
<td>89.44</td>
<td>0.88</td>
<td>0.9089</td>
<td>0.9062</td>
<td>0.8834</td>
</tr>
<tr>
<td>MOFACO-SVM-ACO</td>
<td>91.56</td>
<td>0.9022</td>
<td>0.9289</td>
<td>0.9269</td>
<td>0.9048</td>
</tr>
</tbody>
</table>

Figure 4.6 Classification Accuracy for Medical Opinion Dataset using multi objective function ACO
Figure 4.6 reveals that the classification accuracy of MOFACO-SVM-ACO is better by 11.97% than MOFACO-CART, by 11.69% than MOFACO-Naïve Bayes, by 9.94% than MOFACO-SVM-Poly, by 7.95% than MOFACO-SVM RBF (10,0.01) and by 2.34% than MOFACO-SVM RBF (100,0.1) for medical opinion dataset.

![Recall for Medical Opinion Dataset using multi objective function ACO](image)

**Figure 4.7  Recall for Medical Opinion Dataset using multi objective function ACO**

Figure 4.7 represents that the recall of MOFACO-SVM-ACO works better by 8.46% & 15.45% than MOFACO-CART, by 8.73% & 14.63% than MOFACO-naïve bayes, by 6.34% & 13.54% than MOFACO SVM-Poly, by 4.27% & 11.64% than MOFACO-SVM RBF (10,0.01) and by 2.49% & 2.17% than MOFACO-SVM RBF (100,0.1) when compared with the recall for positive and negative for medical opinion dataset.
Figure 4.8 Precision for Medical Opinion Dataset using multi objective function ACO

Figure 4.8 reveals that the precision of MOFACO-SVM-ACO performs better by 14.42% & 9.46% than MOFACO-CART, by 13.84% & 9.55% than MOFACO-naïve bayes, by 12.53% & 7.3% than MOFACO SVM-Poly, by 10.67% & 5.18% than MOFACO-SVM RBF (10,0.01) and by 2.25% & 2.39% than MOFACO-SVM RBF (100,0.1) when compared with the precision for positive and negative for medical opinion dataset.

4.4 SUMMARY

In this chapter, an opinion mining is performed for movies available in online database and medical opinion dataset by classifying the review comments from the users into positive or negative. SVM classifier is chosen because of its high accuracy and automatic learning. The semantic based feature extraction is proposed by using multi objective functions with SVM and ACO optimized RBF kernel. Results show that the classification accuracy of MOFACO-SVM-ACO performs better by 11.64% than
MOFACO-CART, by 10.42% than MOFACO-Naïve Bayes, by 9.88% than MOFACO SVM-Poly, by 8.01% than MOFACO-SVM RBF (10,0.01) and by 2.47% than MOFACO-SVM RBF (100,0.1) by using IMDB dataset. Similarly, the classification accuracy of MOFACO-SVM-ACO proves to be good by 11.97% than MOFACO-CART, by 11.69% than MOFACO-Naïve Bayes, by 9.94% than MOFACO SVM-Poly, by 7.95% than MOFACO-SVM RBF (10,0.01) and by 2.34% than MOFACO-SVM RBF (100,0.1) for medical opinion dataset. Further investigation is made to improve the speed in feature selection and produce more specific feature subset and better parameter optimization.

Table 4.3 Comparison of Proposed Method with Other Techniques Found in the Literature

<table>
<thead>
<tr>
<th>Author</th>
<th>Technique used</th>
<th>Classification Accuracy Obtained (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basari et al. (2012)</td>
<td>SVM with PSO</td>
<td>77</td>
</tr>
<tr>
<td>Singh and Husain(2014)</td>
<td>SVM with Ngram feature selection</td>
<td>81.15</td>
</tr>
<tr>
<td>Tripathi and Naganna(2015)</td>
<td>SVM -TF-IDF with Ngrams</td>
<td>84.75</td>
</tr>
<tr>
<td>MOFACO-SVM-ACO</td>
<td>MOFACO-SVM-ACO in parameter optimization</td>
<td>90.89</td>
</tr>
</tbody>
</table>