An Effective Optimized Recommender System based on Feature Selection Technique

Major project Submitted in partial fulfilment of the requirements

For the award of degree of

Master of Technology
In
Computer Science & Engineering

Submitted By

HITESH HASIJA

(Roll No. 2K13/ISY/08)

Under the guidance of

Mr. RAHUL KATARYA

(Assistant Professor)

Department of Computer Science & Engineering

Department of Computer Science & Engineering

Delhi Technological University

Delhi

Session 2013-2015

CERTIFICATE

This is to certify that **Mr. Hitesh Hasija** (2k13/ISY/08) has carried out the major project titled "An Effective Optimized Recommender System based on Feature Selection Technique" as a partial requirement for the award of Master of Technology degree in Computer Science & Engineering with Information Systems as specialization from Delhi Technological University.

The major project is a bonafide piece of work carried out and completed under my supervision and guidance during the academic session 2013-2015. The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)
Mr. Rahul Katarya
(Assistant Professor)
Department of Computer Science & Engineering
Delhi Technological University
Bawana Road, Delhi-110042

Acknowledgement

I take this opportunity to express my sincere gratitude towards **Mr. Rahul Katarya, Assistant Professor** (Computer Science & Engineering) for his constant support and encouragement. His excellent guidance has been instrumental in making this project work a success.

I would like to thank **Dr. O.P. Verma**, H.O.D of Department of Computer Science & Engineering for his useful insights and guidance towards the project. His suggestions and advice proved very valuable throughout.

I would like to thank members of the Department of Computer Science & Engineering at Delhi Technological University for their valuable suggestions and helpful discussions.

I would also like to thank my family and friends, who have been a source of encouragement and inspiration throughout the duration of the project. I would like to thank the entire DTU family for making my stay at DTU a memorable one.

Hitesh Hasija
Roll No. 2k13/ISY/08
M.Tech (Information Systems)
E-mail: hitoo.hasija@gmail.com

Abstract

Task of providing recommendations is achievable through recommender systems (RS). There are various classifications of RS i.e. content, collaborative and hybrid based. Apart from these three, there is one context aware RS. Because, through research it has been proved that, ratings provided by user for a movie or TV program also depends on demographic information, day type, time, mood, and other factors of environment. Considering all these contextual features and providing recommendation increases the computational time complexity of our program to a very high value. Apart from that, RS also suffer from cold start problem, scalability problem and data sparsity problem. Hence, artificial neural networks are used to provide recommendations and make this task achievable. Now, training of Artificial Neural Networks takes a very large time, due to high dimensional features. Because, the number of contextual feature attributes could vary from 24 to 35. In order to reduce time complexity, a perfect subset of these attributes should be considered. But, again reducing such a high dimensional contextual attributes is a kind of combinatorial optimization problem. This problem could be solved by using Ant Colony Optimization (ACO). ACO with heuristic information as either covariance or fuzzy values are used as heuristic function. At the end, back propagation algorithm is used, for training the neural network, only on those feature subset obtained via ACO with covariance or fuzzy c means measures. While testing the RS, around 80% of the data set is classified as training data set and rest 20% of the data set is classified as testing data set. Accuracy of recommender system is determined, on testing data set. Finally, mean absolute error has also been calculated and results are analysed by comparing the accuracy of recommender system with previous approaches.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction to Recommender system

Data mining methodologies are used to provide recommendations about what book to read, what movies to be watched, etc. and the system with the help of which it is achieved is known as Recommender Systems (RS). Recommendation algorithms are used behind these RS. RS are very essential from user as well as from product developer point of view. Because, if a user is not able to take proper decisions, that which movie to be watched or which product to be purchased, then RS plays a very vital role in taking these decisions. From product developer point of view, if a particular product is in demand and visited or purchased by users very frequently, then it means that users like it. But, if it is not there, then it means that product is decreasing the market capture of that company, to which the product belongs. Recommender Systems are categorized into various types [1, 2] and they are:-

- Content based RS [3]:-

In content based RS, the past ratings provided by a user are used to predict the new ratings. For example, in case of a movie lens data set, if a person has given high ratings to all the movies containing a specific actor, then usually it is found that he would like the movies of same actor in future too. But, the problem with this approach is you cannot predict the other habits of user. You cannot predict the shopping pattern of a user on an ecommerce website based on his movie likings in the past. Pandora radio is a music RS and it works on content based filtering approach.

- Collaborative filtering based RS [4]:-

In collaborative RS, similar liking pattern amongst different users are determined to provide recommendations. For example, user A and user B has rated different movies with almost near about same ratings, then it means that likings of A and B are near about similar, so based on this if user A follows a particular shopping pattern on an e commerce website, then user B would also be following the same pattern. Now, these similar liking users are determined by generating different clusters. Clustering is done by using k nearest neighbour classifier or other methods. Last.fm is a music recommender system based on collaborative filtering approach. But, this approach mainly suffers from sparse data problem. Majority of the RS are build on collaborative filtering technique [5-8], which has been successfully developed in the past few years.

- Hybrid RS :-

In hybrid based RS, content and collaborative both techniques are used to overcome the shortcomings of both and their advantages could also be utilised simultaneously. Netflix is a movie recommender system which is based on this approach. It is basically provided to overcome the disadvantages of one by using other approach and vice a versa.

1.2 Introduction to Context aware recommender systems

RS also depends upon context into which user's provide recommendations. Context is defined as any information which can be used to characterize a person, place or object that is considered as relevant to the interaction between a user and its application. Context aware RS are based on the theme of environment around the user while he/she was rating any movie or liking any TV program. In collaborative based RS, the recommendations are provided based on the similar liking of users. In collaborative first of all, the friend zone of

a particular user is determined, basically it consists of all those users who have similar likings as that of the person, for which we want to provide recommendations. That methodology of making groups of similar minded users or similar liking of users is known as clustering. Now, clustering is also categorized in 2 ways:-

a. Hard clustering algorithms,

In hard clustering algorithms, a person or say an item either belongs to a cluster with full membership value as 1 or it do not belongs to that cluster at all. Hence, membership is binary in nature, which could be defined as either 0 or 1.

b. Soft clustering algorithms.

But, in real time scenario, the membership of a user in a particular group or the membership of an item in a particular group is a fuzzy value. That fuzzy value is determined with the help of fuzzy c means clustering algorithm. Fuzzy c means is different from k means or other clustering approaches in a way that, it provides a fuzzy membership value of a particular item to be belonging in a particular cluster or not.

1.3 Organization of the Thesis

The remaining thesis is described as follows: Chapter 2 presents the literature survey on context aware recommendation system. Chapter 3, Basic Fundamentals, consists of an introduction to the ACO, neural networks with back propagation technique, along with covariance parameter and fuzzy c means clustering algorithm. Chapter 4, Methodology used to solve the problem, provides two approaches to solve recommender system problems. Chapter 5 covers Experiments and Results, and its comparison with other approaches. Chapter 6, Conclusion and Future Work, summarize the thesis with some possible enhancements as future work.

Chapter 2

Literature Survey

2.1 Problems existing in Recommender Systems

Recommender system suffers from 3 major problems [9]:-

a. Cold start problem:-systems require a large amount of existing data on a user in order to provide accurate recommendations.
b. Scalability problem:-a very large computation power is required by the systems, because millions of users and products are available.
c. Data Sparse problem:-the users do not provide proper ratings to all the available items.

Artificial Neural networks [10] are also used to provide recommendations, but as a coin has two aspects, in the similar way, there are some pros and cons of using artificial neural networks. In the past work, neural network had already been used, to provide ratings in a movie recommender system. But, that was in a binary format, that whether a particular movie is liked by the user or not. Hence, from analysis it could be concluded that, it would be having only one neuron in the output layer, whose value could be either 0 or 1. The value 0 signifies that it has not been liked by the user, but value 1 signifies that it has been liked by the user. This movie recommender system was designed in such a way that, it uses the past ratings provided by the user to predict its future ratings. But, as we know that, this case of predicting values fails in case of cold start problem, that is what to recommend ?, when the system has not enough data to be trained over to it. Hence, the solution of that problem was proposed in, by “Know and Hong”. It deals with cold start problem into it.

First constraint [11] was how to use these neural networks for recommendations, as there were different parameters to be defined first before using a neural network. There could be many number of hidden layers included in our neural network, but how many hidden layers would be efficient, is again a question to be answered. It has already been proved in that, a single hidden layer feed forward neural network provide the recommendation with same accuracy as that of usual back propagation algorithm. Therefore, there is not any problem in using back propagation algorithm to train our network.

Second problem [11, 12] while using neural networks is, the activation function used to convert neuron's weighted input into its corresponding output values. These activation functions could be any linear or non linear activation functions. But, if linear activation functions are used then it usually takes quite a long time for convergence or to be trained. Because the weights are also updated accordingly, thus it also affects the accuracy of our RS. Another drawback with linear activation functions is that, they are not able to cope up with the random input and output pattern. As the input usually do not varies linearly with the output values.

Third issue [12] while using neural networks is that, the values of weights and biases are generally assigned randomly, but it takes a lot of time to converge, hence it further leads to the scalability problem with the neural networks. If the activation functions we are using at hidden layer and output layer are infinitely differentiable, then we can use random values of weights and biases. After that, by using back propagation algorithm we can transform our neural network towards the desired output values.

Fourth point is that [13], which functions are infinitely differentiable functions and whether by using them the RS with that much accuracy could be constructed or not, as we were desirous to get. Some of the examples of infinitely differentiable functions are sigmoid

function, radial basis function, hyperbolic tangent function, sine, cosine and many more non regular functions. Hence, the thesis uses sigmoid function for the activation of hidden layer neurons and hyperbolic tangent function for the activation of output layer neurons.

Fifth note [14, 15], while providing recommendations with the help of neural networks is that, how many numbers of neurons at the input layer, hidden layer and output layer could be used. If the numbers of neurons used at input layer are very large, then it increases the computational complexity of our program. Thus, the scalability problem would not be able to solve up to that extent. As far as dealing with context aware RS is concerned, then as we know that there are very context associated with dataset like in case of a movie lens data set, the contextual attributes are genre, actor, year, companion, weather conditions, etc. Therefore, all the contextual attributes should be used to provide accurate and efficient recommendations. But, as we all are aware that, using 24 dimensional feature vector attributes of context aware RS, as the number of input layer neurons, makes training of neural network a very cumbersome task. [16] Proposed an approach based on neural networks in order to reduce the dimensions of our feature vector. Principal component analysis basically deals with the Eigen value components of a covariance matrix. The Eigen vectors are determined in such a way that, all Eigen vectors are not correlated at all. They are also less in numbers as compared to the number of attributes provided in the given context.

2.2 Feature Subset selection problem

Feature subset selection is considered as one of the discrete optimization problem [17], because if the actual set consists of 'n' different attributes, then there are 2^n possible subsets for that set. Considering all the feature subsets and determining the best one is not possible feasibly. Therefore, in order to solve this problem, optimization and other learning

techniques are frequently used. [18] Suggests that, feature subset selection is considered as one of the challenging problems of machine learning. Basically, it also deals with dimensionality reduction. As soon as the features vectors of a given set gets reduced, then correspondingly the subset also gets reduced, and it would be helpful in reducing the dimensions of given problem statement. Feature subset selection problem has a very wide number of applications [19], like in data mining, text mining, machine learning, artificial intelligence, etc. But, the point to be considered over here is that, methodology used for feature subset selection should be accurate and efficient, otherwise it may also leads to high computational complexity of the system to be designed.

The efficient way to solve a feature subset selection problem is, using evolutionary algorithms which work on the principle of population of data sets and have already shown a greater performance with it. One of the optimization algorithms has been proposed by M.Dorigo [20] in 1990, known as ACO, could be efficiently used to solve this problem. ACO basically works on the principle of pheromone laying behaviour of ants [21]. There are two different methodologies to be followed in order to solve feature subset selection problem, they are as follows:-

i. Filter method [22],

Filter method; do not use the concept of any learning algorithm. It works on the criteria of separability like k nearest neighbour classifier. But, on the other hand, wrapper method uses the concept of learning methodology, like as that of ACO or any other algorithm in which the result of future iterations are dependent upon past iterations.

ii. Wrapper method [23].

As wrapper method uses the concept of optimization and all, thus no doubt that, they produce much better results [24]. But, the drawback of using wrapper method is that, they are much difficult to run on large datasets, because these iterations take a lot of time to run.

The feature selection method is further classified into 5 subcategories [25]:-

1. Forward Selection :- It begins with an empty set and the features are added to it greedily one at a time
2. Backward Elimination:- It begins with a feature set containing all the features and features are removed greedily one at a time
3. Forward Stepwise Selection: - It begins with an empty set and features are either added to it or removed from it greedily one at a time.
4. Backward Stepwise Elimination: - It begins with a feature set containing all the features and features are further added to it or removed from it greedily one at a time.
5. Random Mutation: - It starts with a set of randomly selected features and features are further added to it or removed from it randomly, with number of iterations as the stopping criteria.

A hybrid approach based on ACO and mutual information has already been used in the forecaster [26]. In the given thesis, we are going to use a form of hybrid approach only, which would be based on ACO. But, as ACO consists of a heuristic function. The heuristic function defined in the thesis, has been used in such a way that, it will consists of either the covariance of different attributes with the ratings provided or fuzzy c means clustering measures. For example, if we are using a movie lens data set, then different feature vectors would be there like genre, actor, year, companion, etc., the value of covariance of features or fuzzy values are applied in the form of heuristic function, while applying ACO to it.

Chapter 3

Fundamentals

3.1 Ant Colony Optimization

3.1.1 Introduction to Ant Colony Optimization

In travelling salesman problem [20], the actual problem is to cover all the cities from source city to destination city in such a way that the distance covered or the value of weights assigned on edges should be as less as possible. It works on the pheromone laying behaviour of ants. If there exists two paths 'A' and 'B' from source node to destination node, then first of all the ants are initialized to random trails or random paths, but as soon as the ant reaches to the destination node, then it puts the pheromone or chemicals to attract other ants while traversing down the path. So that, other ants should also follow that path while coming from source to destination node. But, if the ants reaches to path 'A' more earlier and with minimum value of edges or distance covered as compared to path 'B', then it is required to increase the value of pheromone on path 'A' and also to decrease the pheromone level on path 'B'. This is achieved with the help of two constants,

1. ρ is the pheromone evaporation coefficient
2. $\Delta \tau_{xy}^{k}$ is the amount of pheromone deposited by the k_{th} ant in its tour.

While implementing the ACO for thesis work, two major issues are kept in mind:-

1. The rule for updating pheromones
2. Calculation of the value of heuristic function

In case of travelling salesman problem, the pheromone updating rule is same as that of all other approaches i.e. if the path has been covered by the ant previously then it would be considered more for future iterations and if not, then value of pheromones would be decreased on it. The inverse of distance between the nodes is taken as the heuristic function.

3.1.2 Modification in Ant Colony Optimization

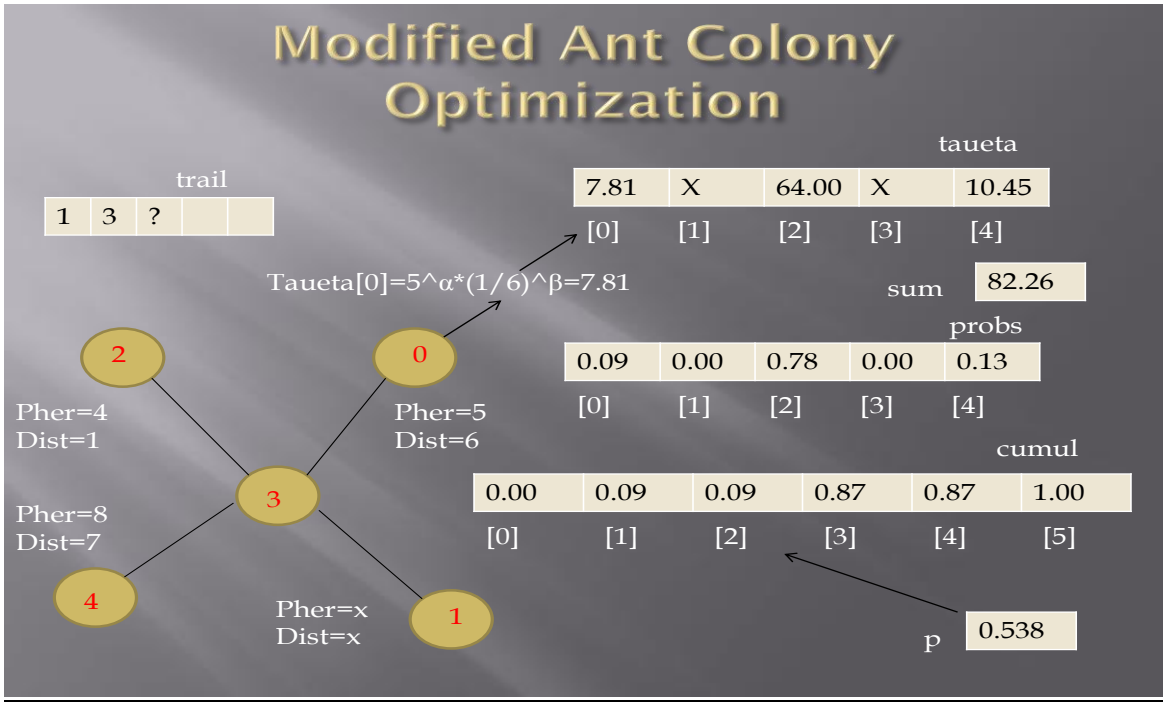

Figure1. Modified Ant Colony Optimization

While implementing ACO for this thesis as described in Figure1, Fisher Yates shuffle algorithm has been used initially to randomize the order of nodes selected during its first iteration. After that, it uses roulette wheel selection algorithm, i.e. first of all the probabilities of all the nodes to be selected as next node is determined and stored inside an auxiliary array. Secondly, we construct another helping array, but with size one greater than the previous one and cell [0] is filled with vale 0.0. Thirdly, all the values of that array are nothing but the cumulative sum of the probabilities up to that node. Fourthly, a random number is generated and if that value falls between the values of node 2 and node 3, then

node 2 would be considered as our next node. In this example, after traversing nodes 1 and 3, next node is determined by calculating their probability to be traversed, based on the pheromones of those connecting edges. After that a sum of all the probabilities is taken in an arbitrary variable like sum. Then all the values are divided by that arbitrary variable, to get the desired probability. Lastly, roulette wheel selection algorithm is applied.

3.1.3 Mathematical approach towards Ant Colony Optimization

As described in [49, 50],

$$p_{xy}^{k} = \frac{\left(\eta_{xy}^{\beta}\right)\left(\tau_{xy}^{\alpha}\right)}{\sum y \epsilon \text{allowed } y \left(\eta_{xy}^{\beta}\right)\left(\tau_{xy}^{\alpha}\right)} \quad (1)$$

τ_{xy} is the amount of pheromone which is deposited for transition from feature subset *x* to *y*,

α is a parameter used to control the influence of τ_{xy},

η_{xy} is the heuristic function for transition of state '*xy*'

(dxy), where *d* is the distance between feature subsets '*x* and *y*' and

$\beta \geq 1$ is a parameter for controlling the influence of η_{xy}.

When all the ants have completed a solution, the trails are updated by

$$\tau_{xy} \leftarrow (1-\rho)\tau_{xy} + \sum_{k} \left(\Delta\tau_{xy}^{k}\right) \quad (2)$$

Where, τ_{xy} is the amount of pheromone deposited for a state transition *xy*,

ρ is the pheromone evaporation coefficient and

$(\Delta \tau_{xy}^{k})$ is the amount of pheromone deposited by k_{th} ant, and it is calculated by the formula:-

$$(\Delta \tau_{xy}^{k}) = \begin{cases} Q/L_K \text{ if ant k uses curve } x\ y \text{ in its tour} \\ 0 \text{ otherwise} \end{cases} \quad (3)$$

Where, L_k is the cost of the k_{th} ant's tour and Q is a constant.

As mentioned in Equation 2 and 3, if that ant covers a particular path let us say 'AB' in its tour of traversing from A to D, then right hand side factor of the summation sign would get increased and it will cancel out the decreasing effect produced by left hand side factor. But, in case that path has not been covered in the tour, then factor towards the left of summation sign will dominate to that of right hand side. Thus, it decreases the effect of that path to be considered again, while traversing the tour.

3.2 Fuzzy C Means Clustering Algorithm

3.2.1 Introduction to Fuzzy C Means Clustering Algorithm

A good clustering algorithm should be designed in such a way that it should maximize the intra cluster similarity and should minimize the inter cluster similarity, as specified in. Clustering algorithms are classified into 2 parts [28]:-

I. Partition based clustering algorithms

II. Hierarchal based clustering algorithms

In partition based clustering algorithms, the given set of n items is divided into different defined partitions, and based on that clustering is formed. In this case, we are well aware of the number of cluster we have to form as the numbers of partitions are already defined to us. But, on the other hand, in hierarchal clustering algorithms, they are further categorized into 2 parts, agglomerative and divisive. In hierarchal clustering algorithms, a tree of clusters gets formed known as dendrogram. While implementing clustering there are two important issues to be considered. Firstly, number of clusters to be formed. Secondly, measure to calculate the distance between them.

3.2.2 Mathematical approach for Fuzzy C Means Clustering

Fuzzy c means clustering algorithm is developed by "Dunn" and further improved by "Bezdek". In fuzzy c means clustering algorithm, the value of membership function is initially, assigned a random value from 0 to 1. This value of membership function is modified in further iterations based on the probability distribution of different items in different clusters. In different iterations, the main focus of algorithm is to minimize, the value of objective function [27], which is defined as follows with Equation 4:-

$$J_m(P) = \sum_{k=1}^{n} \sum_{i=1}^{c} [A_i(x_k)]^m \parallel x_k - v_i \parallel^2 \quad (4)$$

In this equation, outer loop runs n number of times, where n is the number of data points taken. Inner loop runs c number of times, where c is the number of clusters defined initially. In this case, c is equal to the number of contextual attributes taken for our data set. M is defined as fuzziness coefficient; it decides the measure of overlapping to be provided between different clusters. If the value of m is very large, then it means that a higher overlapping could be provided between different clusters. But, on the other hand if the value is very small, then overlapping is not possible at all, and there would be no reason of using fuzzy c means clustering algorithm over there. The value of m could range from 1 to

infinity. X_k is the value of data point to be considered, by subtracting its value from $V_{k,}$ where $V_{k\ is}$ the value of centre point of that cluster, which is considered in the inner loop. Lastly, A_i is the value of membership function, to be calculated or modified again and again in every loop, so that its fuzziness of data point with respect to a cluster could be determined. Its value is modified for all iterations, according to the formula:-

$$A_i^{(t+1)}(x_k) = [\sum_{j=1}^{c} (\frac{\| x_k - v_i^{(t)} \|^2}{\| x_k - v_j^{(t)} \|^2})^{\frac{1}{m-1}}]^{-1} \quad (5)$$

The value of membership function is modified as per the rule given in Equation 5, and as the value of membership function is modified, accordingly, the value of centre for different clusters is also modified in all iterations, as data point's increases or losses the membership values with respect to different clusters. The formula used for centre calculation is as follows, defined in Equation 6:-

$$v_i = \frac{\sum_{k=1}^{n} [A_i(x_k)]^m x_k}{\sum_{k=1}^{n} [A_i(x_k)]^m} \quad (6)$$

Finally, in the last step, we have to determine the stopping condition. It is defined, in terms of some threshold value, which could be taken as 0.15 and its equation is defined as follows;-

$$| P^{(t+1)} - P^{(t)} | = \max | A_i^{(t+1)}(x_k) - A_i^{(t)}(x_k) | \quad (7)$$

Hence, if | P (t+1) - P (t) |≤ε, then stop as described in Equation 7, otherwise repeat all the steps of membership function calculation and centre calculation as mentioned above.

3.3 Neural Networks

3.3.1 Introduction to Neural Networks

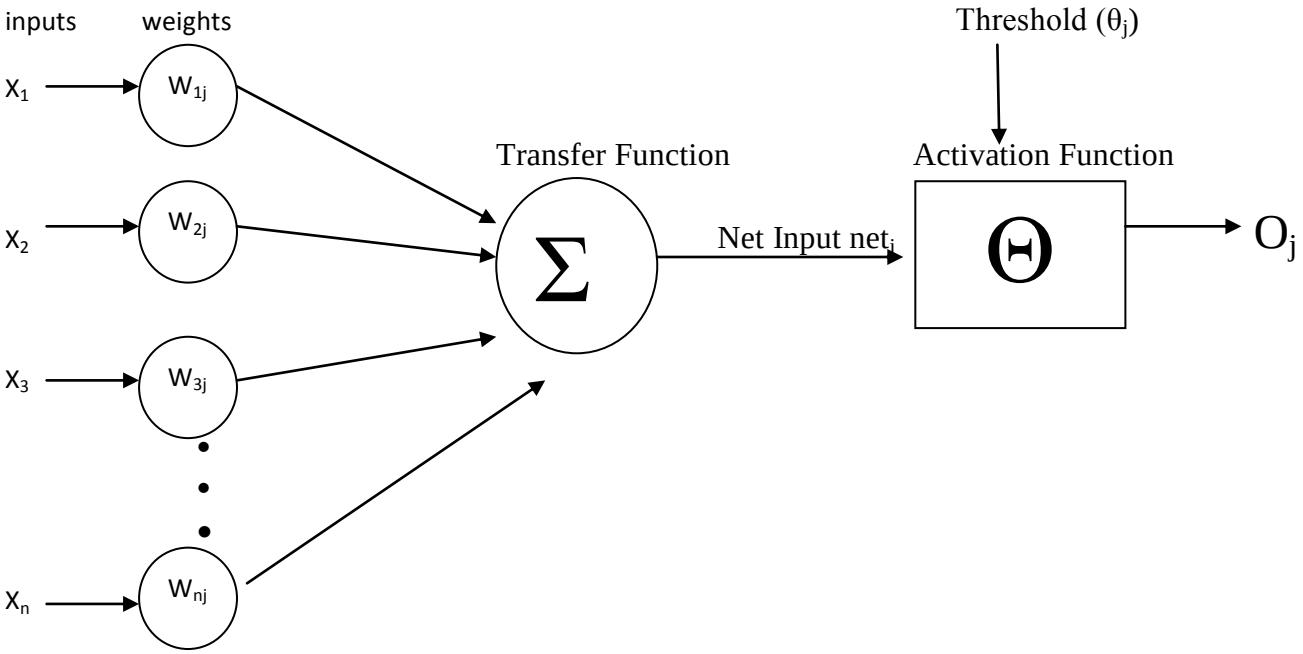

Figure2. Schematic diagram of Artificial Neural Networks [29]

An artificial neural network (ANN), as presented in Figure 2 and as described in [29], is a set of interconnected neurons where each circular node acts as an artificial neuron and an arrow indicates a connection from the output of a neuron to the input of other neuron. Each arrow consists of some weights value assigned to that connection. An ANN consists of different layers of neurons; the given example set consists of n neurons in input layer and 1 neuron in output layer. The above given example network is a kind of feed forward neural network, because it is a directed acyclic graph.

Artificial neural network is again meta heuristic approach [29] to solve problems. It basically consists of different layer of neurons and the first layer is known as input layer of neurons and the last layer is known as output layer of neurons. In between these two layers, there are 'n' numbers of layers and all are known as hidden layer of neurons. Now, the neurons in input layer are connected to the neurons of output layer with the help of edges

and weights are assigned over those edges. Apart from that, biases are also connected to the hidden layer neurons and also the output layer neurons. Initially, the values of weights and biases are assigned randomly with the help of a random function. Then, output values are calculated for the corresponding input values. First of all, an input signal is applied to the input layer of neurons, then based on the values of weights and biases, as well as based on some mathematical function, the activation has been provided. Another function is used to compute the output values of those neurons, again with the help of weights and biases assigned. Hence, artificial neural networks basically work on 3 different parameters and they are as follows:-

- The activation function used to convert input values of neuron into output values,
- Methodology used to train the neural networks, like back propagation is used.
- The interconnection defined between different layers of neurons, like number of nodes in input, hidden and output layer, as well as how they are interconnected.

As the value of weights and biases are assigned randomly, thus the output of the neural network would not be equivalent to the desired output. This difference between the desired output and actual output is known as the error. Hence, training is required to set the values of weights in such a way that, the neural network will produce output corresponding to the given input set of values. For training, back propagation algorithm is used, which is basically a supervised learning mechanism. Because, in supervised learning, we are aware of the input values and corresponding output values both. So, this error is propagated in the backward direction, to train the neural network. In other iteration, again the output values are computed and compared with the corresponding input values, till the sum of absolute difference of error for all the neurons become less than 0.01 or the number of iterations reaches to a value of 1000.

3.3.2 Back Propagation learning Algorithm with formulas

Back Propagation [29] is a method of backward propagation of errors, used for the training of artificial neural networks. It is a supervised learning method. Basically, there is a contribution of neuron of each layer in determining the results. Hence if any change is required in order to get the results as per our convenience, there is a need to update the weights of interconnection by some methodology, and that algorithm is known as Back Propagation learning methodology. The weights are assigned between -1 to +1 using a random function in visual studio. For implementing back propagation algorithm three layers of neurons are used. Input layer is denoted by i, hidden layer by j and output layer by k. Steps to be followed are [29]:-

i. To compute output of hidden layer:-

$$V_j = \sum_{i=1}^{N_i} w_{ji} y_i + \theta_j, \quad (8)$$

Where, W_{ij} is the value of weights of interconnections from i th layer to jth layer,

Y_i is the values of input signals applied from ith layer to jth layer,

θ_j is the values of biases applied to the hidden layer neurons,

ii. To compute error for each output layer neuron:-

$$e_k = d_k - y_k, \quad (9)$$

Where, d_k is the desired output value; y_k is the obtained output value,

e_k is the absolute difference value between obtained and desired output,

iii. To compute local gradient for each output layer neuron:-

$$\delta_k = (d_k - y_k) * y_k * (1 - y_k), \quad (10)$$

Where, δ_k is the change required in the weights of interconnections,

iv. To adjust weights for output neuron:-

$$W_{kj}=W_{kj} + \eta * \delta_k * y_{j,} \quad (11)$$

Where, η is the value of learning rate to be applied,

v. After every iteration calculate average error as:-

$$E=\sum_{k=0}^{n}(dk - yk)^2, \quad \text{Iterate till } E >=0.01 \quad (12)$$

3.4 Covariance parameter for similarity matching

Covariance is a measure of how much one variable is dependent upon another variable. If the values of covariance obtained between any two attributes like companion and ratings provided by user in movie lens data set is '1', it signifies that as the value of one variable increases then correspondingly the value of another variable also increases. But, if the value obtained is '-1', then it signifies negative correlation, it means that as the value of one variable increases then correspondingly the value of another variable decreases. And, if the value obtained is '0', it signifies that, as the value of one variable increases or decreases then it does not affect at all, the value of other variable.

$$\mathrm{cov}(x, y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{(n-1)} \quad (13)$$

Where, n is equal to the number of samples considered for determining the value of covariance, x is independent and y is dependent variable, X_i is the value of that independent variable at i^{th} instance and Y_i is the value of dependent variable at i^{th} instance, $\bar{x}$ is the average value of that independent variable and $\bar{y}$ is the average value of dependent variable.

Chapter 4

Methodology

4.1 Explanation for Ant colony Optimization, Neural Networks, and Covariance

In order to solve problem of feature subset selection using ACO along with heuristic function, a suitable heuristic function should be defined. As Ant colony optimization is also used for solving standard travelling salesman problem, so from there we get the idea of our heuristic function, it uses the distance values assigned between the nodes as its heuristic function. But, the main motive of travelling salesman problem is to reduce the value of distance covered; hence the inverse of this distance value is taken as its heuristic function. Therefore, the actual problem of covering minimum distance in travelling salesman problem has been converted to the minimization of distance values while traversing the nodes from source city to the destination city. In the similar way, the feature subset problem has been converted to graph form and then a suitable heuristic function is defined in the form of covariance values of different attributes with respect to the ratings of movies provided in the data set.

After that, we have to determine the nodes in a given graph. For this particular problem, the nodes would be the possible subsets of given features. If the given feature set consists of 'n' possible contextual attributes, then number of nodes would be 2^n possible subsets of given set. Secondly, all the nodes are connected with weights or edges between them. These weights are nothing but the covariance values of that particular subset with respect to the ratings provided. Hence, heuristic function as used in travelling salesman problem is the inverse of these weight values, because in that case our aim was to minimize the distance

covered. But, here in this problem our aim is to maximize the value of covariance of a particular subset. Hence, more would be the value more affects that particular subset will produce in determining the ratings of users. In the meanwhile, while implementing ACO, values of four constants used are as follows:-

a. Alpha (pheromone influence factor) =3
b. Beta (local node influence factor) =2
c. Rho (pheromone evaporation coefficient) =0.01
d. Q (pheromone deposit factor) =2.00

Now, after the execution of ACO, the result would be sequence of nodes to be traversed, just like that of a travelling salesman problem. But, out of that sequence of nodes we have to consider only the first subset as our answer, that subset could be defined in any manner as, it could be consisting of 5 nodes or 7 nodes or even more than that. Finally, after determination of feature subset, that which all features to be considered and which not, there is a need to move ahead with training of neural networks. Now, the training of neural networks is done only with the help of those features obtained after applying covariance values as heuristic function to our data set. Initially, the weights and other interconnections have been assigned random values. Because, if the value of activation functions used at hidden and output layer are infinitely differentiable, then random values would also get converge as per output. Numbers of neurons in input layer are equal to the, number of features present in the feature subset obtained. As, our aim is to provide recommendations for different movies, so we have to first find out the ratings provided by different users to those movies with the help of our neural network based RS. At the end, the neural network is trained on the features subset obtained, and using data present in our data set. But, before training the neural network, data set is divided into 2 parts, 80% of our data set is considered as training data set, which is only used for training the neural network, and rest

20% is used as testing data set, which is used for testing the RS. In order to train the neural network based optimized recommender system, back propagation algorithm has been used.

Now, with the help of neural network obtained, we could also determine the values of ratings of those movies which are not yet provided in our dataset, as we could solve now the actual cold start problem. Because, now the recommender system designed could provide recommendations for every user and for any kind of movie. After solving cold start problem, there is a need to solve scalability and sparsity problem also. As far as the scalability problem is concerned, then as neural network has been trained only on that feature subset obtained. Therefore, the computational time complexity of our recommender system is much less as compared to the computational time complexity of other RS, as they used to consider all the contextual attributes, while providing training to the neural network. Accuracy for training data is computed by using the formula:-

Accuracy= (No. of samples correctly classified / Total no. of samples)*100, ---------- (16)

As given in equation (16), accuracy is determined by determining the number of misclassified samples, out of total number of samples obtained. In the similar way, mean absolute error is calculated just as the reverse of accuracy obtained, because if any sample is misclassified or if any sample fails to provide proper rating while testing, then it would be counted as the error. After that, precision and recall values are also determined. In order to compare it with other previous approaches, as in the previous work, the numbers of clusters are defined already and mean absolute error has been calculated based on those different cluster values. Similarly, here also mean absolute error is determined and accuracy is calculated with different values of obtained clusters i.e. with 4 clusters, 8 clusters, and 16 clusters and so on. Now, as in clusters also, sometimes the results are calculated by defining the number of neighbours before starting the algorithms.

4.2 Flowchart for Ant colony Optimization, Neural Networks, and Covariance

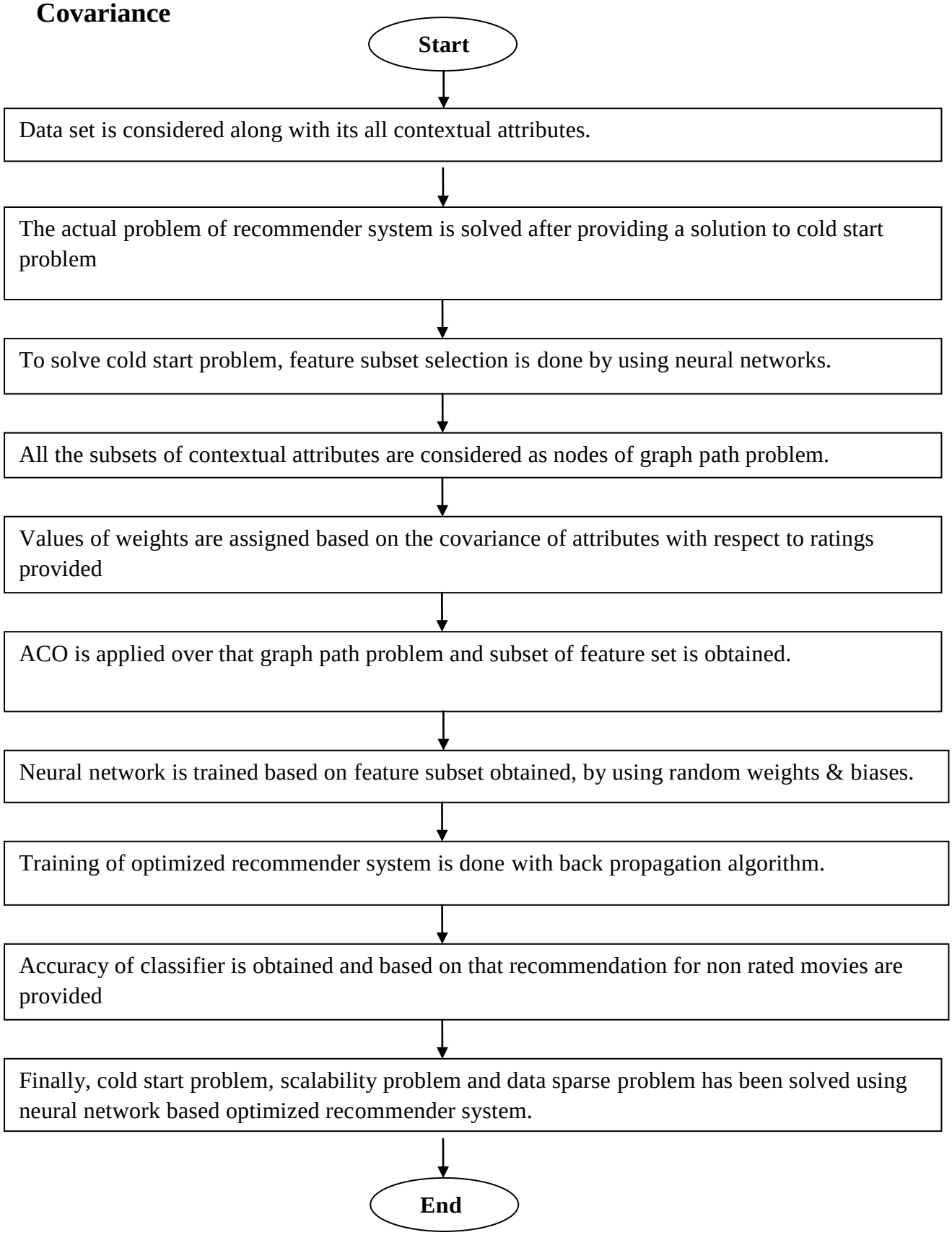

Figure3. Flowchart for methodology 1 used to solve the problem.

4.3 Explanation for Ant colony Optimization, Neural Networks, and Fuzzy C Means Clustering

With fuzzy c means clustering algorithm, as the given approach is used to form clusters or it is used to divide the given data sat of n items into different partitions or clusters which are already defined in number. Thus the number of clusters would be equivalent to the different contextual attributes. As initially, in the given table we are usually provided with the movies as the items with first column and genre or contextual attributes as first row. So, we have to start with a tabular matrix, drawn between movies and its contextual attributes. Corresponding to each and every movie we are provided with the values of its contextual attributes that whether that movie consists of comedy, romance, adventure, action or not. Hence, these values of different contextual attributes act as data point in an n dimensional plane where, n is defined as the number of contextual attributes. After applying fuzzy c means clustering algorithm over to it, the clusters get formed as per the contextual attributes defined. Now, these clusters, consists of different movies into them with some membership function defined for each item or movie. As fuzzy c means clustering algorithm, says that an item could also belongs to more than one cluster but with different values of membership function defined over to it. Hence, in this case also, one movie could also belong to more than one cluster, with different membership function values defined over to it. Finally, we get the result in the form of tabular matrix again, but between different movies or items and clusters obtained. From here, we could obtain the values of different contextual attributes with respect to different movies, in the form of membership function defined. Finally, these values of membership function are added to obtain the contribution of different contextual attributes in different movies, which is further used as the heuristic function in ACO, and rest all approach is same as that of previously defined one.

4.4 Flowchart for Ant colony Optimization, Neural Networks, and Fuzzy C Means clustering algorithm

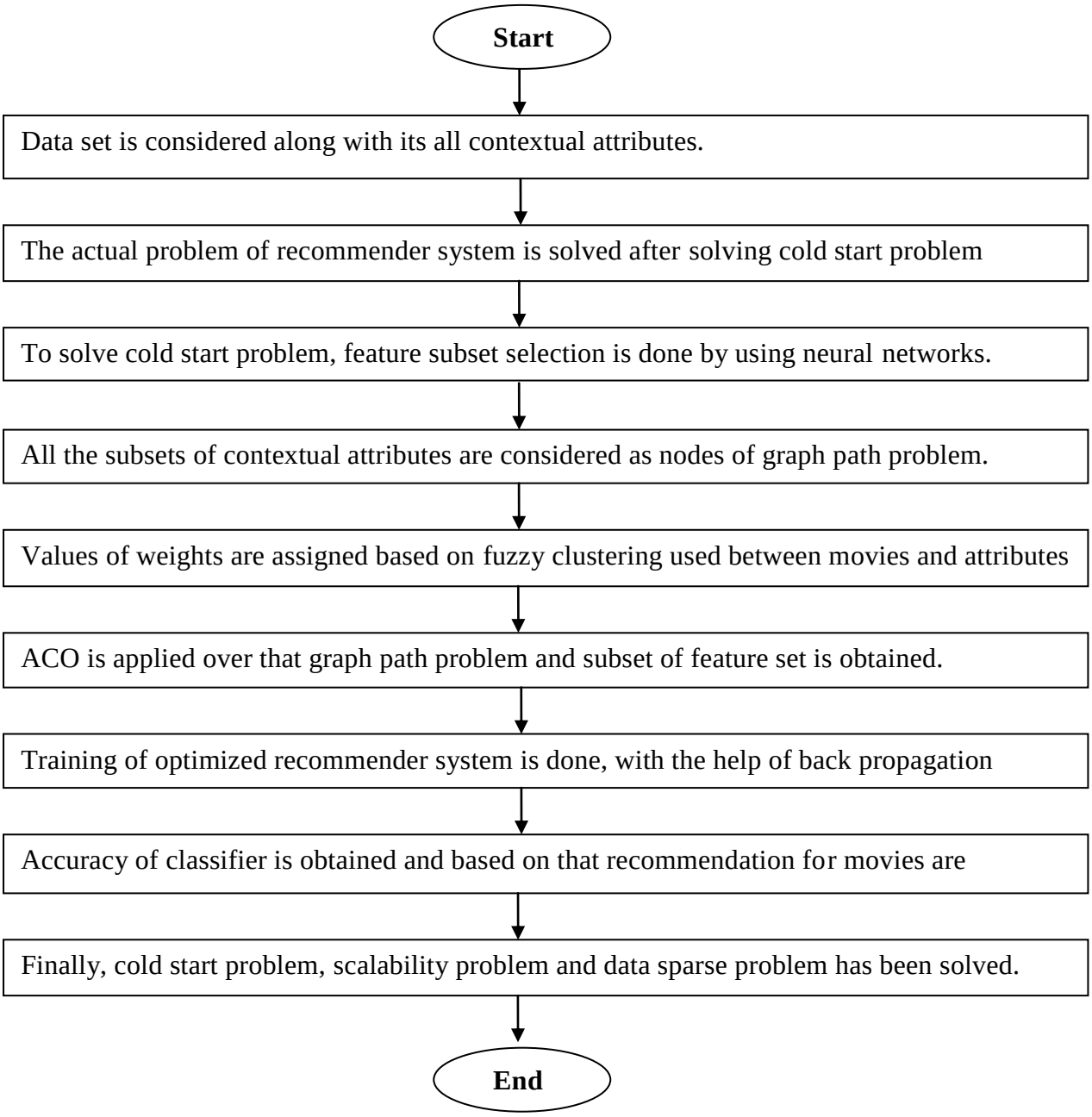

Figure4. Flowchart for methodology 2 used to solve the problem

Chapter 5

Experiment and results

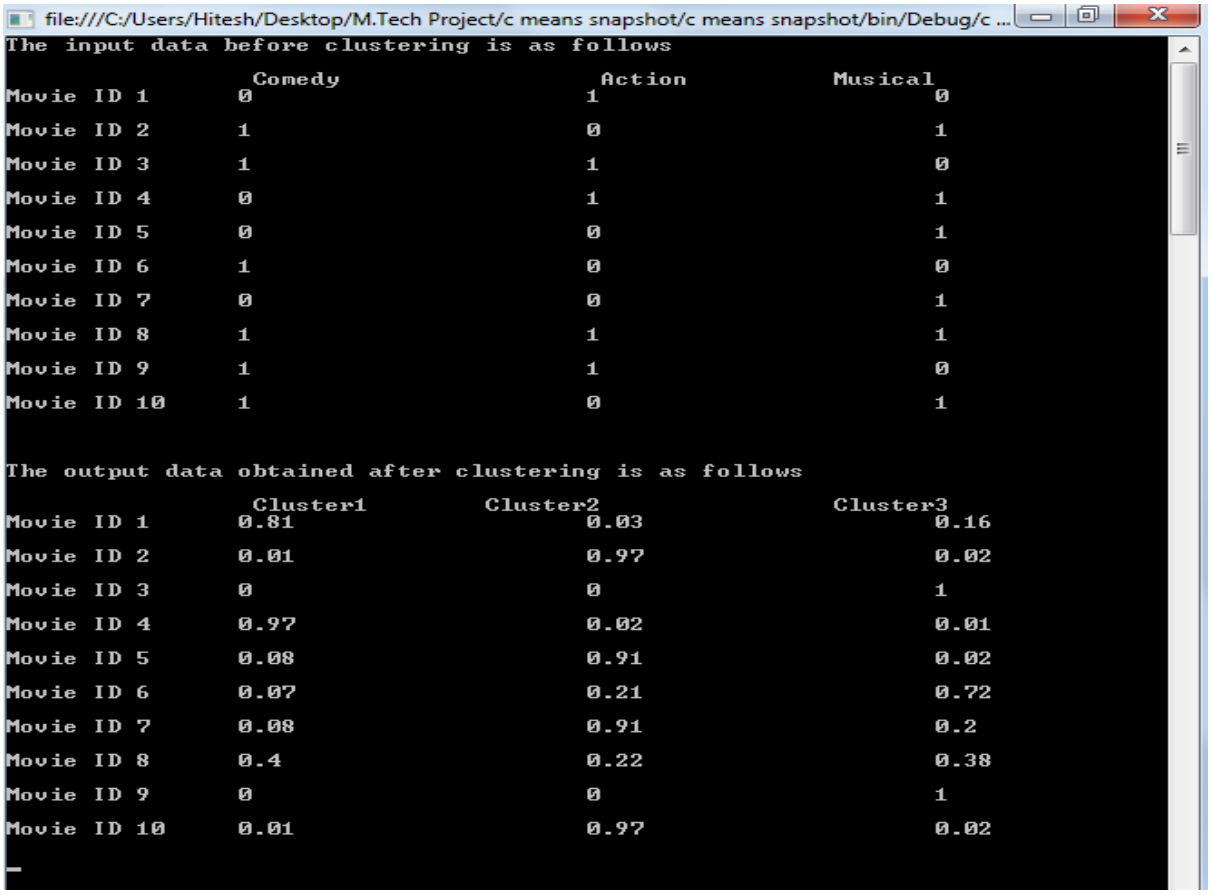

Figure5. Snapshot as implementation of basic fuzzy c means clustering algorithm.

The given snapshot mentioned above in Figure5, is a simple example of fuzzy c means clustering algorithm applied over a dummy data set, in which 3 contextual attributes are taken i.e. comedy, action and musical. Every row corresponds to one movie and represented by its Movie ID. Every movie has certain characteristics i.e. every movie is related with some genre which are associated with it. In movie lens data set, there are about 19 genres given which represents different characteristics of a movie. In order to represent these movies with some features of musical, some features of action and some features of comedy, we have to use fuzzy c means clustering algorithm.

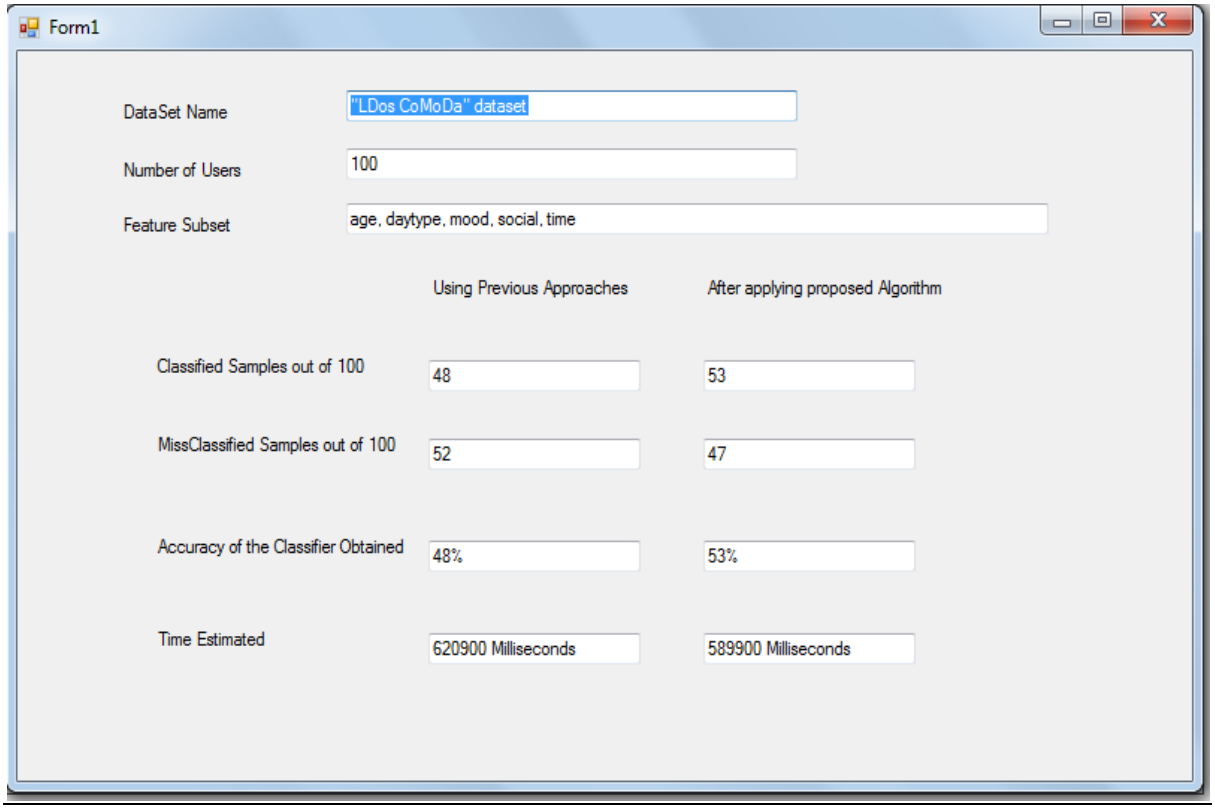

Figure6. Snapshot as implementation of proposed algorithm on “LDOS COMODA” set.

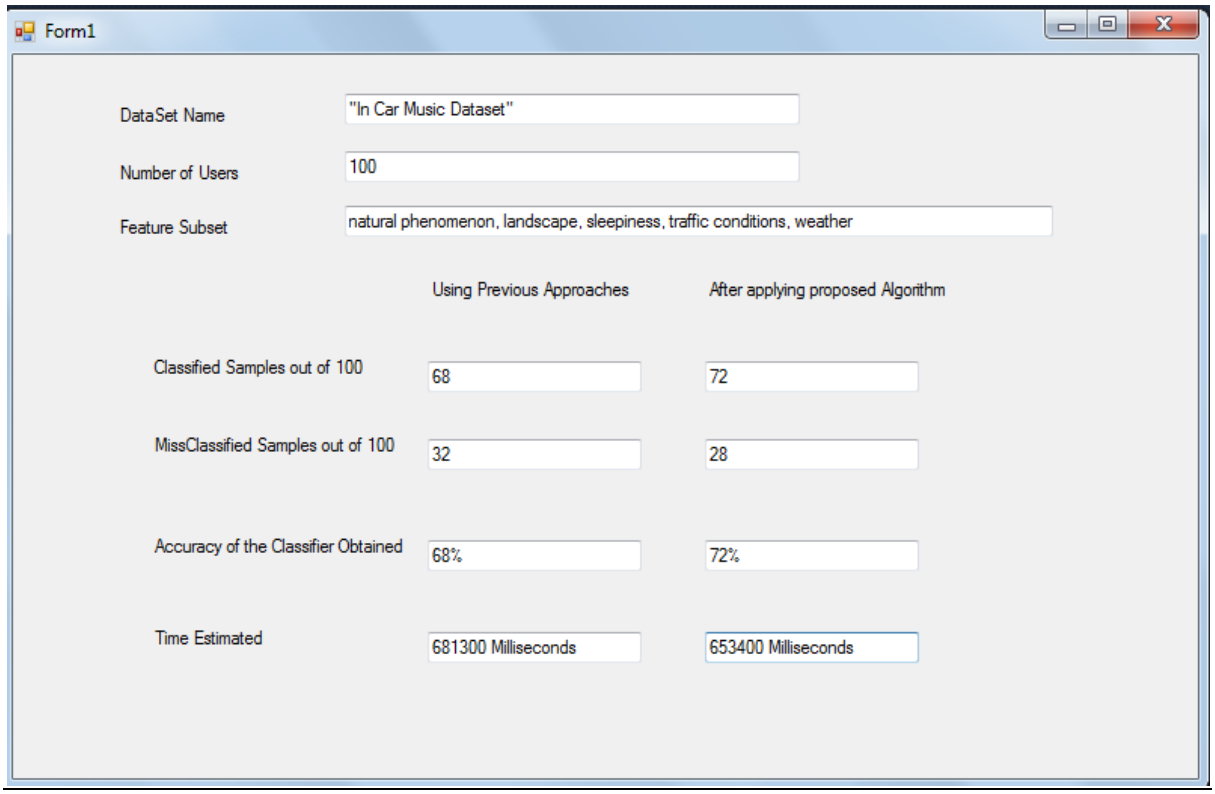

Figure7. Snapshot as implementation of proposed algorithm on “In Car Music data” set.

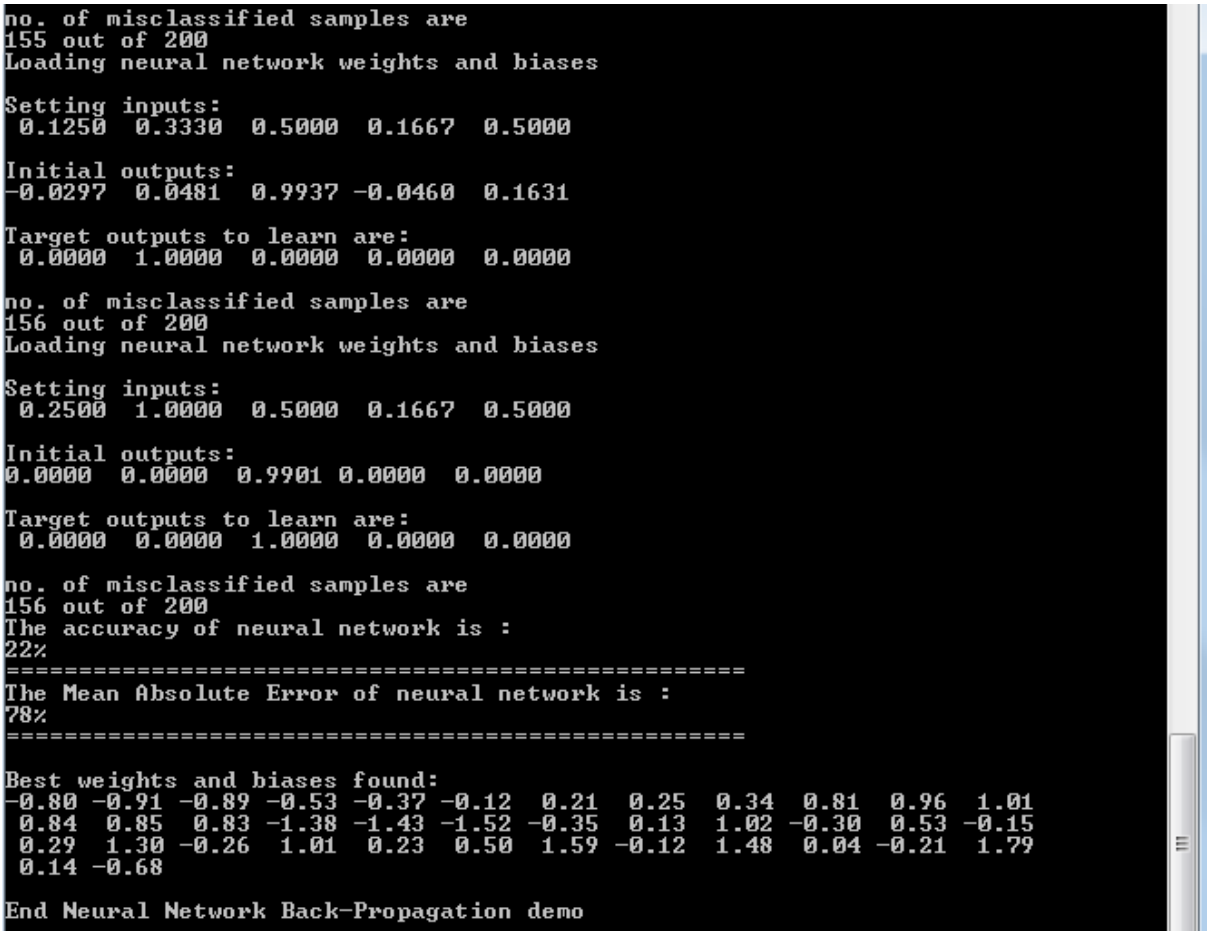

Figure8. Calculation of Mean absolute error with 4 clusters

As mentioned in the given snapshot of Figure8, implementation for the calculation of mean absolute error with 4 clusters has been done. The numbers of clusters are equivalent to the number of possible subsets of contextual features obtained. Hence, the numbers of clusters are already defined to be 4, and accordingly the membership values of different movies are obtained. Finally, the determination of perfect feature subset is done by using ACO. Results are obtained by training the neural network on that perfect feature subset and then observing its behaviour by testing it on rest of the 20% of our data set. Total number of samples taken over here are 1000 out of which, 800 are used for training the neural network and rest 200 are for testing. Misclassified samples are obtained by comparing the obtained ratings from actual ratings.

Further, a graph has been drawn after obtaining these values of mean absolute error with different number of clusters and comparison with all the previous approaches [30] has been provided. The graph is designed not only, for number of clusters, but also for, the number of neighbours to be taken inside a cluster and based on that clusters are formed. Graph shown below, proves that efficiency of defined approach in the thesis is better as compared to other previous approaches.

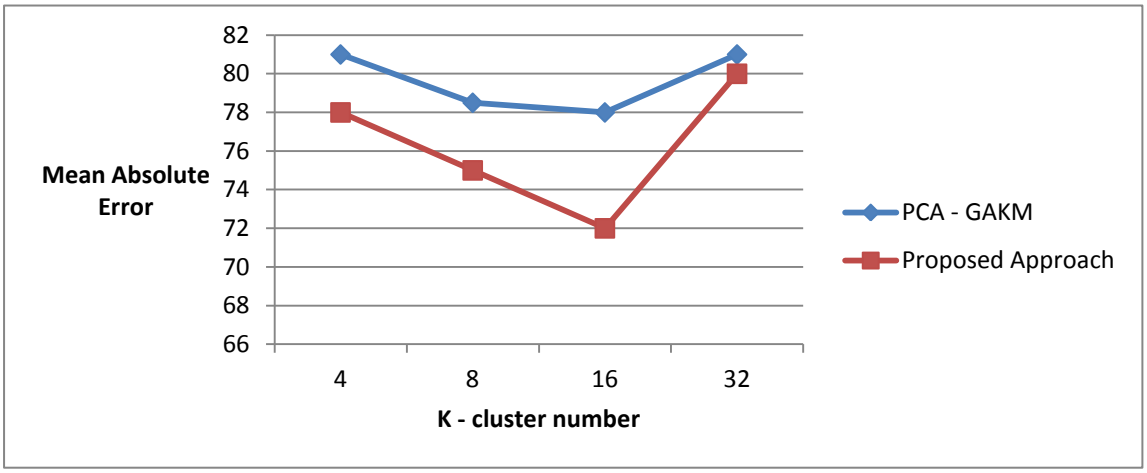

Figure9. Precision error on different cluster numbers

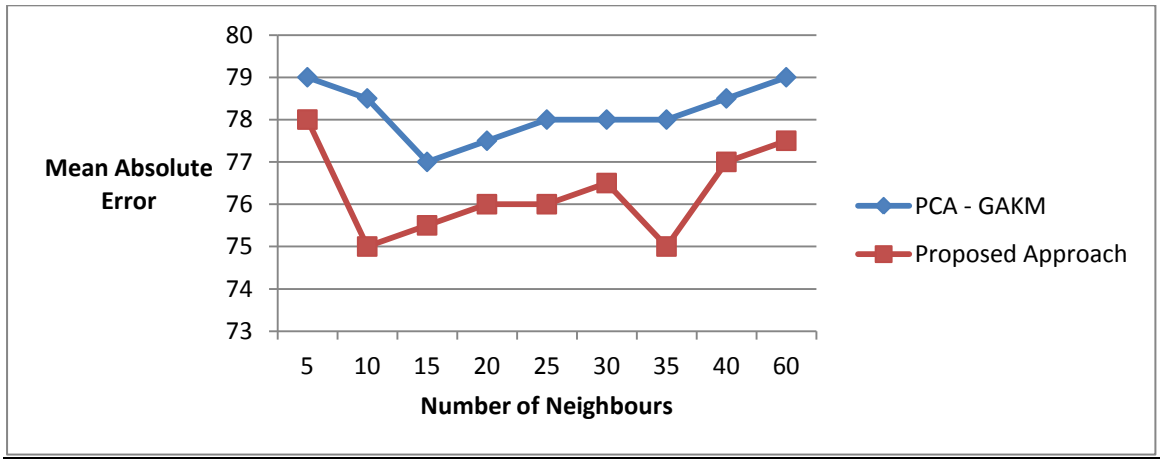

Figure10. Comparing accuracy with clustering based Collaborative Filtering approaches

Chapter 6

Conclusion & Future Work

ACO with heuristic information is a very efficient evolutionary algorithm especially for optimization problems like travelling salesman problem, discrete combinatorial optimization problem, etc. Covariance and fuzzy c means clustering algorithms as heuristic function provided better results. The results obtained through ACO with fuzzy c means clustering algorithm, i.e. subset of feature vectors, have a great influence to the ratings provided by user as compared to other subset of features. Our methodology used to solve the problem of RS is more efficient as compared to already existing methods. Similarly, when it has been used with fuzzy c means clustering algorithm, then it has been observed that value of mean absolute error has decreased, justifying our approach as much efficient one.

As far as, the future work is concerned then ACO with covariance or fuzzy values to be used as heuristic function, could also be applied to content based and collaborative based recommendation systems, along with back propagation. Instead of covariance, another parameter to measure the correlation i.e. correlation coefficient can also be determined between different feature subsets and ranking provided by user to different movies or TV programs. There are various similarity measures like cosine similarity function, sine similarity function, etc. or other non linear variations that could also be applied along with ACO. But, the condition is that it should be able to produce better results. Other than, ACO there are various optimization algorithms, which could also be applied and as neural networks uses sigmoid function and hyperbolic tangent function for the activation of hidden and output layer neurons, then other non linear functions could be applied in order to get better results.

References

1. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of state-of-art and possible extensions”, IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June 2005.
2. Gediminas Adomavicious et al, “Incorporating Contexual information in recommender systems using a multidimensional approach”, ACM transactions on Information Systems, vol. 23, Issue1, 2005
3. Alan Eckhardt, “Similarity of user’s preference models for Collaborative filtering in few ratings scenario”, Journal of Expert Systems with Applications 39(12), pp. 11511-11516, 2012.
4. Alan Said, “Identifying and utilizing Contextual Data in Hybrid Recommender Systems”, In Proceedings of RecSys’10, pp. 365-368,ACM 2010.
5. Burke, R., “Hybrid web recommender systems”, In the Adaptive Web, pp. 377–408. Springer Berlin / Heidelberg (2007).
6. G.Linden, B.Smith, J.York, “Amazon.com recommendations: item to item collaborative filtering”, IEEE Internet Comput. 7 (1) (2003) 76–80.
7. B.M.Sarwar, G.Karypis, J.Konstan, J.Riedl, “Recommender systems for large scale e-commerce: scalable neighbourhood formation using clustering”, in: Proceedings of International Conference on Computer and Information Technology, Dhaka, Bangladesh, 2002.
8. B.M.Sarwar, G.Karypis, J.Konstan, J.Riedl, “Item - based collaborative filtering recommendation algorithm”, in: Proceedings of the 10th International WWW Conference, Hong Kong, 2001, pp.285–295.

9. Sanghack Lee and Jihoon Yang and Sung-Yong Park, "Discovery of Hidden Similarity on Collaborative Filtering to Overcome Sparsity Problem", Discovery Science, 2007.
10. H. J. Kwon, and K. S. Hong, "Personalized smart TV program recommender based on collaborative filtering and a novel similarity method", IEEE Trans. Consumer Electron., vol. 57, no. 3, pp. 1416- 1423, August 2011.
11. M. Krstić and M. Bjelica, "Personalized TV program guide based on neural network", Proc. 11th Intl. Symp. Neural Network Applications in Electrical Engineering (NEUREL 2012), pp. 227-230, 2012.
12. G.-B Huang, Q.-Y Zhu, C.-K Siew, "Extreme learning machine: theory and applications", Neurocomputing, vol. 70, no. 1-3, pp. 489-501, December 2006.
13. G.-B. Huang, H. A. Babri, "Upper bounds on the number of hidden neurons in feed forward networks with arbitrary bounded nonlinear activation functions", IEEE Trans. Neural Networks, vol. 9, no. 1, pp. 224-229, January 1998.
14. F. S. da Silva, L. G. P. Alves, and G. Bressan, Personal TV ware: "An infrastructure to support the context-aware recommendation for personalized digital TV", Intl. J. Computer Theory and Engineering, vol. 4, no. 2, pp.131-136, April 2012.
15. S. H. Hsu et al., AIMED – "A personalized TV recommendation system in P. Cesar et al. (eds.), Interactive TV: a shared experience", Lect. Notes Computer Science, vol. 4471, pp. 166-174, 2007.
16. D. Mladenic, "Feature selection for dimensionality reduction", Berlin Springer Verlag, 2006, ch-5
17. Marko Krstic and Milan Bjelica, "Context aware personalized program guide based on Neural Network", IEEE, 2012.

18. H.R.Kanan, K Faez, and M Hosseinzadeh, “Face Recognition system using ant colony optimization based selected features”, IEEE Symposium on Computational Intelligence in Security and Defense Applications, pp. 57-62, USA, 2007.
19. W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu and Z. Wang, “A novel feature selection algorithm for text categorization”, Expert systems with applications, vol. 33, no. 1, pp. 1-5, July 2007
20. M. Dorigo and G.D.Caro, “Ant colony optimization: A new Meta heuristic”, congress on Evolutionary Computing, 1999.
21. E, Bonabeau, M. Dorigo and G. Theraulaz, “Swarm Intelligence: from natural to artificial systems Oxford University Press”, New York, 1999.
22. R.O.Duda and P.E.Hart , “Pattern classification and scene analysis”, John wiley and sons., Chichester, 1973.
23. G.Forman, “An extensive empirical study of feature selection metrics for text classification”, Journal of machine learning research 3, pp. 1289-1305, 2003.
24. R. Jensen, “Combining rough and fuzzy sets for feature selection”, Ph.D. dissertation, School of information, Edinburgh University, 2005.
25. D. Mladenic, “Feature selection for dimensionality reduction”, Berlin Springer Verlag, 2006, ch-5
26. C.K.Zhang and H.Hu, “Feature selection using the hybrid of Ant Colony Optimization and mutual information for the forecaster” , in Proc. 4th International Conference Machine Learning and Cybernetics, vol. 3, pp. 1728-1732, Aug 2005
27. Lin Zhu, Fu-Lai Chung, Shitong Wang. “Generalized Fuzzy C-Means Clustering Algorithm With Improved Fuzzy Partitions” [J]. IEEE Transactions on Systerms,2009:39-3.

28. A. K. Jain and R. C. Dubes, “Data clustering: A review,” ACM Computing Surveys, vol. 31, 1999.
29. C. Christakou and A. Stafylopatis, “A hybrid movie recommender system based on neural networks”, Proc. 5th Intl. Conf. Intelligent Systems Design and Applications (ISDA’05), pp. 500-505, 2005.
30. Zan Wang, Xue Yu, Nan Feng, Zhenhua Wang, “An improved collaborative movie recommendation system using computational intelligence”, Journal of visual languages and computing 25 (2014) 667-675.