CHAPTER 5

CONTEXTUAL DEEP WEB SEARCH USING SHUFFLED FROG LEAPING ALGORITHM

This chapter utilizes the potential of nature inspired algorithms to augment the performance of deep web search systems. Prevalent techniques to develop the deep web search systems are content based filtering and social context based filtering. These techniques suffer from the cold-start, sparsity and scalability problem. In content based deep web search systems, a Nature Inspired Algorithm (NIA)-Case Based Reasoning (CBR) hybrid approach is developed to tackle the cold-start problem. In social context based deep web search systems, the scalability problem is dealt by employing NIA such as memetic algorithm (MA) and shuffled frog leaping algorithm (SFLA). Subsequently, ‘Web of trust’ of user’s is incorporated to solve the cold-start and sparsity problems. To further enhance the performance, personal demographic context of users is integrated with the social context in deep web search systems. Empirical results over MovieLens and Epinions dataset substantiate the strength of presented approaches in augmenting the efficacy of deep web search systems.

5.1 Introduction

The rapid growth of internet technologies has brought tremendous changes in the web development trends; from single tier web architecture to multi-tier web architecture, simple html based informative websites to database driven websites. Introduction of server side scripting and database technology to the internet has added a new dimension to the web; deep web. Deep web refers to websites which maintain their contents in the databases located on the web server. Such content can be accessed only when a user submits a query to the dedicated search engines provided by these dynamic web applications. Digital libraries, e-commerce websites, e-reservations websites, blogs, forums etc are all examples of deep web search systems. The rising popularity of deep web applications in the recent years has lead to an explosive growth of information which exceeds the ability of human beings to process them; the information overload problem. To alleviate this information overload problem, intelligent techniques are being used to automatically identify the needs of active
users and recommend products, information and services to potentially interested users. Examples of such web applications include web personalization, web mining, semantic web and intelligent web etc. These applications take into consideration heterogeneity of sources and user interests to intelligently utilize and serve vast, rich and shared web resources and services available on the web e.g. Amazon, Netflix etc.

The widespread approaches used to develop deep web search systems include content based filtering (Balabanovic and Shoham, 1997) and collaborative filtering (Goldberg et al 1992)/ Social context based filtering (Tamine-Lechani et al, 2010). These approaches primarily identify the traits of the active user and then retrieve relevant content based on either the user’s personal past preferences i.e. user’s personal interests (content based filtering technique) or automatically suggest relevant content on the basis of user’s social community preferences (collaborative filtering technique). This chapter primarily presents approaches to overcome the inherent limitations of content and social context based filtering systems. It then explores the effect of user’s personal ‘web of trust’ and demographic context on the performance of social context based deep web search systems.

5.2 Content based Deep Web Search Systems

Content based deep web search systems are developed using content based filtering techniques. These techniques attempt to understand the context of user’s current information need based on the past preferences made by the user itself. The most successful artificial intelligence technique applied (Aamodt and Plaza, 1994) to perform content based filtering is Case Based Reasoning (CBR). CBR (Kolodner, 1993) is a problem solving paradigm that emphasizes reasoning and planning from prior experiences. Each case in CBR systems represents a plan or sub-plan to solve a particular problem. CBR retains every new experience in the memory, each time a problem is solved, making it immediately available for future problems. However, CBR approach proves inefficient in a situation for which it has no past experience. For example systems developed by Niknafs et al (2003), Camacho et al (2006) and Yang and Cheng-Shu (2008) fail to provide any suggestions to the user if the initial cases are not populated manually. This problem is popularly known as Cold-Start
problem. It is more prevalent in the systems which depend upon past solutions to solve future queries.

In the recent studies, integration of nature inspired algorithm like Genetic Algorithm (GA) with CBR has been observed as successful innovation over traditional CBR technique (Jarmulak et al, 2000). Genetic algorithms (GAs) are population-based meta-heuristics invented by John Holland in the 1960s (discussed in Chapter 3, Section 3.3.1). Ahn et al (2004) developed a hybrid (CBR-GA) model named SOCBR (Simultaneous Optimization of CBR) that outperformed the other models namely COCBR (Conventional CBR), FWCBR (CBR with Feature Weighting) and ISCBR (CBR with Instance Selection) in judging the status of corporates being bankrupt or non-bankrupt. Other applications such as travel and foodmart (Yang and Cheng-Shu, 2008), tablet formulation problem (Jarmulak et al, 2000), Corporate Bond Rating (Shin and Han, 2008) and timetable (Grech and Main, 2005) solutions etc also established the strength of CBR-GA hybrid approach. Although existing CBR-GA hybrid techniques performed better than conventional CBR, yet the cold-start problem persisted. This work solve the prevalent Cold-Start problem by utilizing the potential of nature inspired algorithm (NIA) in Case Based Reasoning systems.

5.2.1 NIA-CBR Hybrid Approach for Content based Deep Web Search

The architecture of the prototype NIA-CBR hybrid system (Mehta et al, 2010) developed to solve the cold-start problem is shown in the Figure 5.1. The system employs agents (Wooldridge, 2003) to carry out the various activities in the system. The team of agents constitutes following agents - User Agent (UA), Content Retrieval Agent (CoRA), Case Based Reasoning Agent (CBRA), Nature Inspired Algorithm Agent (NIAA), Repository Agent (RA) and Verification Agent (VA). The working of various agents in the system is as follows:

**Step 1** The User Agent (UA) accepts the query submitted by the user and communicates it to the Content Retrieval Agent (CoRA) for further processing. CoRA is assisted by two subordinate agents for query processing–Case Based Reasoning Agent (CBRA) and Nature Inspired Algorithm Agent (NIAA). CoRA invokes CBRA (to fetch the similar cases based on preferences provided by the user) and NIAA (for generating the new cases). CoRA builds an abstract representation of the query received from UA, and processes it to generate the results.
Step 2 CBRA refers the Case Base library to check if it has any prior cases stored with respect to the user specified query. CBRA then tries to revise the existing cases according to the instant inputs received from the user. Finally, the cases are adapted according to the desires of the user and communicated to CoRA for further processing.

Step 3 CoRA checks if the number of cases received from CBRA are sufficient (that is if they are above threshold) to satisfy the user. In such a situation if there are no resultant cases retrieved by CBRA (e.g. if user submits a new query for which CBRA has no past experience), prevalent CBR based systems fail to provide any results to the user. However in the presented system CoRA invokes NIAA to generate novel cases. Simultaneously CoRA also invokes Verification Agent (VA) to verify the validity of cases generated by CBRA, before communicating them to the UA. It helps in providing only the updated and valid information to the user.
Step 4 NIAA initially invokes Repository Agent (RA) to gather the information from the repository. NIAA employs nature inspired algorithm to optimize the information received from RA and evolve cases according to the user specified requirements.

Step 5 Thereafter NIAA stores new cases and their sub-cases in the CBR library for solving future queries.

Step 6 CoRA communicates integrated cases (from VA and NIAA) to the User Agent (UA). The top N cases with minimum time within the specified budget are presented to the user by the User agent.

The presented NIA-CBR hybrid technique eliminates the need to populate the initial cases manually as NIAA generates novel cases to tackle this situation. In addition, this agent also aids in populating additional cases if the number of cases generated by the CBR agent are not sufficient.

5.2.2 Case Study of NIA-CBR hybrid approach for Content based Deep Web Search: E-reservation systems

To assess the competence of presented NIA-CBR approach with respect to traditional CBR technique, a case study on E-reservation systems was conducted. In the prevalent studies nature inspired algorithm namely genetic algorithm (GA) was used to enhance the performance of CBR systems. Thus in this study GA was employed to solve the cold-start problem. The various parameters of GA adapted for the case study on E-reservation systems are as follows:

Size and Structure of Chromosome: The size and structure of chromosome depends on the user preference of the route. Route with one hop and two hops have chromosome of size 3 and 4 respectively.

Structure of chromosome for route with one hop:

<table>
<thead>
<tr>
<th>Source</th>
<th>Budget</th>
<th>Hop</th>
<th>Budget</th>
<th>Destination</th>
</tr>
</thead>
</table>

Structure of chromosome for route with two hops:

<table>
<thead>
<tr>
<th>Source</th>
<th>Budget</th>
<th>Hop</th>
<th>Budget</th>
<th>Hop</th>
<th>Budget</th>
<th>Destination</th>
</tr>
</thead>
</table>
Size of population determines the number of plans generated by the system. An initial population of 200 chromosomes was generated. Any change in population size leads to corresponding change in quantitative number of plans generated by the algorithm.

**Objective Function:** To compute the fitness of a particular chromosome, the fitness function is as follows:

\[ F(Chr) = \sum(n) \]  

(5.1)

Where \( Chr \) an individual chromosome, \( n \) is the total number of ways one can reach the destination from the source.

**Selection, Crossover and Mutation Operators:** Elitism (Mitchell, 1999) mechanism was used for selecting the chromosomes for next generation. The algorithm performed multipoint crossover with crossover probability of 0.7 and a mutation probability of 0.03.

**Evolve operator:** An additional operator namely evolve operator was developed. The functionality of evolve operator involved duplication of best fit genes with only a slight modification to the parameters. If the new genes are better, old genes are replaced with the new genes. As a result genes evolve into better solutions, thereby improving the fitness of the chromosome. The operator was used along with the conventional crossover and mutation operators for optimizing the cases within specified number of generations.

**5.2.3 Performance Evaluation**

Rigorous experiments were performed to assess the competence of presented NIA-CBR approach with respect to traditional CBR technique. For E-reservation systems, there were no benchmark datasets available. Therefore, to generate the datasets, specialized programs were developed to retrieve the real data of Indian flight\(^1\) and train\(^2\) modes of transport. The number of stops considered in each case (i.e. flight and train) includes 162 cities and 20 cities respectively for train and flight modes of transport.

1. http://www.cleartrip.com
2. http://www.indianrail.gov.in
On the whole dataset constituted 90000 tuples of information about various trains and 900 tuples for diverse flights, between the two stations with all possible information. The various attributes of trains and flights mode of transport include source, destination, fares of different classes, arrival time and departure time. Figure 5.2 illustrates the graphical user interface created for e-reservation system. The system takes source, destination, departure date, modes of transport- train, flight or both and budget as input from the user and generates complete plans for the journey based on the selected modes of transport.

![Figure 5.2 GUI for E-reservation System](image)

The various studies performed to evaluate the NIA-CBR approach are as follows:

- To validate the ability of NIA-CBR hybrid system for serving diverse information requirements of the users. For a query like reaching Calcutta from Bangalore on a budget of ₹9000 using both modes (train and flight), the algorithm generates varied kinds of plans like 1-hop plans, plans with 2-hops and direct plans as shown in Figure 5.3. All these plans are synchronized across train and flight modes.
of transport with different permutations and combinations (Figure 5.3). Plans with 2-hops may be Flight-Flight-Train or Flight-Train-Train etc. The optimized plans within the user specified budget are depicted in the decreasing order of time for complete journey.

To examine the capability of NIA-CBR hybrid approach in solving cold-start problem and generate novel plans. System was tested on 50 random queries with different source and destinations for both (train and flight) modes of transport and travel budget of around ₹4000. Figure 5.4 shows the number of plans with 1-hop and Figure 5.5 exhibits the number of plans with 2-hops between source and destination, generated by CBR alone and NIA-CBR hybrid. These figures precisely illustrate that NIA-CBR is able to generate plans even for the preliminary queries whereas CBR system starts generating plans only after getting some experience of the related queries. Thus, NIA-CBR approach is competent in providing results irrespective of any past experience. Figure 5.6 compares the total number of plans with zero hops, one hop and two hops between the source and destination, retrieved by traditional CBR system and hybrid NIA-CBR. These results establish the performance of
presented approach to generate total number of indirect plans (1-hop and 2-hop) as compared to the number of available direct plans which are very less as indicated in Figure 5.6. Together the direct and indirect plans can lead to an enhanced user experience.

Figure 5.4 Travel Plans with 1-hop

Figure 5.5 Travel Plans with 2-hops
5.3 **Social Context based Deep Web Search Systems**

In deep web search systems, social context pertains to the social community of the user with similar kinds of interests and preferences. Deep web search systems developed using social context based filtering techniques (SF) (Herlocker et al, 1999) primarily identify the community of the active users by comparing the profile of active user with other users and then retrieve items based on the community preferences. Generally, in social context filtering systems, profiles are represented by the explicit/implicit ratings made by the users on the various items of interest. Explicit rating/voting refers to the preferences expressed by the user for the item, usually on a discrete numerical scale. For example the GroupLens (Konstan et al, 1997) system uses a scale of one to five for users to rate the Netnews articles. Users explicitly rate each article after reading it. Implicit rating refers to the interpretation of the user behaviour or selections to input a vote or preference based on web browsing data or purchase history etc. SF technique does not exploit any information about the features of the products, thus it has been successfully applied to extensive range of applications (Schafer et al, 2007).
Research in SF turned vigorous with the development of Tapestry system (Goldberg et al, 1992) that was based on pull-based collaborative filtering approach. In such systems, users willing to have suggestions had to proactively pull them out from the databases. The immense success of this system led to the development of push based SF systems (Maltz and Ehrlich, 1995). In contrast to pull based systems, users of these systems had the privilege to push the items of interests to their friends and peers. With the wide acceptance of semi-automatic systems, fully automatic SF systems came into being e.g. GroupLens system (Resnick et al, 1994) developed for Usenet newsgroup articles, Bellcore’s Video Recommender (Hill et al, 1995) for movies and Ringo system (Shardanand and Maes, 1995) for music etc. Other commercial SF systems include Amazon.com for books, Jester system (Goldberg et al, 2001) for jokes and PHOAKS system (Terveen et al, 1997) to help users find pertinent information on the web.

In general, two main approaches (Breese et al, 1998) are used to develop the SF systems; Model-based SF and Memory-based SF. Memory-based SF technique utilize the whole user-item rating dataset to compute the relevant items for the user. Grouplens and Ringo are both memory based SF systems. These systems employ statistical techniques to identify a set of users known as neighbours whose past behaviour has been similar to the target user. Neighbours’ preferences are then combined to compute relevant suggestions for the target user e.g. K-nearest neighbour is the most prevalent technique employed to identify the set of neighbours in order to recommend relevant items to the user. However, performance of this technique starts declining with the increase in number of users.

On the contrary, Model-based SF systems primarily prepare a learning model of the community’s historical preferences using machine learning algorithms. Thereafter the trained model is used to compute relevant items for the active user. The widespread techniques used to develop the learning model include Bayesian network (Jin et al. 2004) and clustering algorithms (Kohrs et al, 1999). Among these, clustering techniques are used to handle the scalability problem as they reduce the size of data to be processed online to 1/k times where k is number of clusters. Thus, clustering algorithms are considered more preferable techniques to develop the learning model in SF systems.

K-means clustering is the most common technique used for clustering as it takes less time; nevertheless it has good chance of getting trapped in local optima. To overcome this problem, global search techniques like nature inspired algorithms such as Genetic algorithm; memetic algorithm, particle swarm optimization etc have become more popular. Ujjin and Bentley (2003) used particle swarm optimization algorithm for similarity computation to improve the accuracy of social context based filtering systems. The efficiency of genetic clustering algorithm (Zhang and Chang, 2006) has been proved to be better than k-means and memory based social filtering in tackling scalability problem. Bedi et al (2009) pointed that social filtering systems based on the behaviour of Ants perform better than k-means over decision support metrics. The above approaches emphasized that nature inspired algorithms based SF systems perform better than their conventional counterparts. However following drawbacks still persist.

- **New user problem (Cold-Start)** - Since preferences of the new user are not available, it’s hard to identify the community of the new user.
- **New item problem** - System does not recommend new items until sufficient number of users have rated them.
- **Sparsity problem** - Users are usually reluctant in providing ratings, it becomes difficult to make accurate predictions.
- **Scalability problem** - SF systems need to process large amount of data from large number of users for making predictions. Therefore, computational resources become a critical issue to find users with similar tastes.

In this chapter, studies have been performed to handle the Scalability, Cold-Start and Sparsity issues in model based SF systems. The nature inspired algorithms namely memetic and shuffled frog leaping algorithm are employed to deal with the scalability problem. To tackle the cold-start and sparsity problem, user’s trusted social network is utilized for computing relevant items. Thereafter, a context aware approach is developed by integrating the user’s personal demographic context in nature inspired algorithm based SF technique to further enhance the accuracy of the system.
5.3.1. Social Context Filtering (SF) using Nature Inspired Algorithms (NIA)

The presented approach to develop the model based SF system using nature inspired algorithm is depicted in Figure 5.7. In model based SF systems, the process of computing relevant items for the target user begins with the specification of the user’s feedback in the form of initial set of ratings generated explicitly or implicitly. Subsequently, the system tries to estimate the rating function \( R \) (Equation 5.2) for the (user, item) pairs that have not yet been rated by the users.

\[
\text{Rating Function}(R): \text{User} \times \text{Item} \rightarrow \text{Rating} \quad \text{(5.2)}
\]

![Figure 5.7 Methodology to Incorporate Nature Inspired Algorithm (NIA) in SF](image)

After estimating the rating function \( R \) for the whole \( User \times Item \) space (Equation 5.2), SF system retrieves the highest-rated items (or N highest-rated items) for each user.

The working of presented model based SF system using nature inspired algorithms is divided into two phases. In the Phase-1 that is preprocessing phase, background data maintained in the form of user-item rating matrix is divided into Source/training \( s(i,j) \)
dataset and test/target \( (t(i,j)) \) dataset. The training dataset is utilized to create the learning model using nature inspired algorithm based clustering approach as shown in Figure 5.7. After forming the predetermined number of clusters and computing their centroids, information is stored in the database for future predictions. In the next phase, clustered data of trained model is utilized to compute relevant items for the users. The detailed working of the presented model is as follows:

**Phase-I: Developing Social Context Model using NIA based Clustering**

The nature inspired algorithms considered to develop the clustering model in social context filtering are Memetic algorithm (MA) (Banati and Mehta, 2010c) and Shuffled frog leaping algorithm (SFLA) (Mehta and Banati, 2012a). The various parameters of MA and SFLA (as discussed in Chapter 3, Section 3.3.2 and Section 3.3.4 respectively) adapted to develop the clustering model are as follows:

**Structure of the Potential Solution:** It consists of N memes as shown in Figure 5.8. The N memes represent the centroids of N user based clusters respectively.

| Centroid 1 | Centroid 2 | ...... | Centroid N-1 | Centroid N |

**Figure 5.8 Structure of Frog/Chromosome**

For all other users, similarity weight \( (sw\,(i,j)) \) of every user is computed with every meme/centroid using the adjusted cosine similarity function as given in Equation 5.3. Users are assigned to the meme with maximum similarity weight to generate the cluster.

\[
sw(i,j) = \frac{\sum_{n=1}^{n_{items}} (U_{in} - \bar{o}(n))(U_{jn} - \bar{o}(n))}{\sqrt{\sum_{n=1}^{n_{items}} (U_{in} - \bar{o}(n))^2} \sqrt{\sum_{n=1}^{n_{items}} (U_{jn} - \bar{o}(n))^2}} \tag{5.3}
\]

In Equation 5.3, \( Ui \) and \( Uj \) are the vectors containing ratings given by user \( i \) and \( j \) for item \( n \). \( \bar{o}(n) \) refers to the average of the ratings given by all users on the \( n \)th item. On the basis of similarity weight, users belonging to the respective clusters for a solution of size \( N = 20 \) are shown in Figure 5.9. In the Figure, first, second and
third column indicate id number of the cluster, centroid of the cluster and id’s of the various users belonging to the corresponding clusters respectively.

<table>
<thead>
<tr>
<th>id</th>
<th>Centroid</th>
<th>users</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>177</td>
<td>3,32,42,50,64,75,77,90,95,98,102,109,171,177,199,277,220,246,265,285,306,316,321,324,325,334,350,351,</td>
</tr>
<tr>
<td>4</td>
<td>295</td>
<td>295,256,363,339,359,368,308,305,141,530,73,556,5,252,185,239,192,515,240,343,593,108,194,393,5,</td>
</tr>
<tr>
<td>5</td>
<td>493</td>
<td>2,15,10,116,125,153,197,234,261,272,276,364,365,457,492,493,552,</td>
</tr>
<tr>
<td>8</td>
<td>181</td>
<td>255,111,25,26,43,49,90,126,139,149,156,162,169,181,190,196,227,248,280,37,375,377,381,400,403,474,4,</td>
</tr>
<tr>
<td>9</td>
<td>465</td>
<td>31,51,55,11,14,153,118,224,286,315,346,348,394,411,471,432,445,461,465,510,519,555,560,</td>
</tr>
<tr>
<td>10</td>
<td>21</td>
<td>13,21,66,68,72,74,106,121,124,151,242,279,276,288,303,307,329,426,442,475,478,458,</td>
</tr>
<tr>
<td>11</td>
<td>389</td>
<td>75,463,298,430,231,441,37,37,384,111,232,409,33,124,57,178,146,410,67,221,438,213,112,389,</td>
</tr>
<tr>
<td>12</td>
<td>541</td>
<td>41,44,62,105,154,164,267,336,338,361,390,410,429,664,476,539,527,541,</td>
</tr>
<tr>
<td>13</td>
<td>245</td>
<td>28,48,245,261,316,319,327,326,613,413,344,556,</td>
</tr>
<tr>
<td>15</td>
<td>129</td>
<td>129,186,198,201,206,279,520,63,</td>
</tr>
<tr>
<td>17</td>
<td>148</td>
<td>26,39,42,44,49,78,86,100,132,147,149,166,220,229,299,293,294,954,444,466,495,500,517,538,542,005,</td>
</tr>
<tr>
<td>18</td>
<td>14</td>
<td>10,14,19,26,24,103,123,129,130,206,211,233,241,294,292,308,412,427,495,496,</td>
</tr>
<tr>
<td>19</td>
<td>513</td>
<td>6,9,16,27,34,52,60,61,70,81,82,104,115,121,157,150,173,174,103,102,205,219,251,275,278,283,297,297,200,</td>
</tr>
<tr>
<td>20</td>
<td>292</td>
<td>1,5,45,96,130,142,193,179,212,214,153,235,235,257,292,299,258,299,977,449,523,594,</td>
</tr>
</tbody>
</table>

Figure 5.9 Centroids and Users for solution of size = 20

Objective Function: Equation 5.4 depicts the objective function used by both MA and SFLA to compute the fitness of a particular solution (Sol) as follows:

$$\text{Fitness}(\text{Sol}) = \frac{\sum_{i=1}^{N} \text{Fitness}(C_i)}{N} \tag{5.4}$$

where \(C_i\) is the cluster and \(N\) is the number of clusters in the solution. The fitness of a particular cluster is determined by computing the mean similarity of all users in the cluster with its centroid/meme as given in Equation 5.5.

$$\text{Fitness}(C) = \frac{\sum_{i=\text{Cent}}^{n} \text{sw}(i,Cent)}{n} \tag{5.5}$$

where \(Cent\) is the centroid of the respective cluster and \(n\) is the number of users in the cluster and \(\text{sw}(i,Cent)\) is the similarity weight of every user with centroid of the cluster. Figure 5.10 depicts the fitness value of every cluster in a chromosome of size \(N = 20\).
Figure 5.10 Snapshot of Fitness value of the Cluster in a Chromosome

Convergence criteria: It is set to fixed number of generations for both the algorithms. Thus algorithms are stopped after defined number of generations.

Crossover Methodology in Memetic Algorithm (MA): MA adapted the crossover operator (Merz and Zell, 2002) to create the population for next generation. The various steps followed to generate an offspring of the two chromosomes A and B are:

- For all memes/genes $b_i$ in parent $B$, determine the nearest meme $a_i$ in parent $A$, where $1 \leq i \leq N$, $N$ is number of memes/genes in each solution.
- Assign $b_i$ to the memes of $a_i$ with maximum similarity. For each memes/genes $a_i$ in chromosome $A$, a list of assigned genes is maintained. e.g. If $a_i$ is nearest to $b_i$, $b_i$ is inserted into the list.
- Finally the offspring chromosome contains memes/genes of $A$ which are closer to more than $n$ memes/genes of $B$ where $n$ lies between $0$ to $K$. Value of $K$ may be decided according to the problem. To retain size of chromosome, rest of the memes were selected randomly from the list of memes in parent $B$ which were closer to $A$.
- To maintain the size of initial population, second offspring is produced by applying same recombination strategy in the reverse order.
Local Search: Local search optimization is the key quality of memetic algorithms. The strategy used to perform the local search in the presented system is as follows:

- From an individual chromosome with $N$ clusters, cluster with maximum fitness is selected.
- Then the $n$ worst members (members with minimum similarity with the centroid) from the chosen cluster are extracted and redistributed in the other $N - 1$ clusters.
- Thereafter from the left out $N - 1$ clusters, $t$ worst clusters are identified.
- All the members from these $t$ worst clusters are reassigned among themselves based on their similarity with the centroids of the $t$ clusters.
- This process of local search optimization is followed for defined number of iterations for every individual in the population.
- In next step an additional K-means operator is used to update the centroids of whole population after every generation.

Phase II: Compute Relevant Items using NIA based Model developed in Phase I

The main task of social context filtering (SF) systems is to predict the relevant items for an active (online) user on the basis of past preferences given by the social community of the user. In SF, the prediction task has been broadly categorized as rating prediction and choice prediction. Choice prediction involves the task of predicting which user would rate what items and is often based on implicit user feedbacks for example ‘who rated what type of task’ in KDDCUP 2007 (Bennett et al, 2007).

In contrast rating prediction refers to the task of estimating the rating score that a user may assign to an item. This is evaluated by holding out certain amount of user ratings in the form of test dataset and then comparing the predicted ratings with the true ratings. Rating scores measure the extent to which a user likes an item. Thus the performance of rating prediction reflects an algorithm’s ability to capture user’s preferences over items. The present study is performed on rating prediction task. After creating the models (in
previous phase) using Shuffled frog leaping and Memetic clustering algorithms, test dataset is used to evaluate the predictive accuracy of the respective model for social context filtering. Ratings for the unrated items are predicted using two methods as follows:

**Rating Prediction using a single cluster:** In this method, the cluster having highest similarity with the active user was used to predict the ratings. The prediction algorithm (Shardanand and Maes, 1995) modified to predict how user $i$ in the test dataset $t(i,j)$ would rate item $j$ is as follows:

- Compute the similarity weight ($sw(i, Cent)$) of active user $i$ with the centroids (Cent) of all clusters using the similarity function given in Equation 5.3.
- For prediction, select the cluster with maximum similarity weight for the active user. The users of the selected cluster comprise the neighbourhood ($Ni$) community of the active user $i$.
- Thereafter compute the similarity weight $w(i,k)$ of each user $k$ in $Ni$ using Equation 5.6 as follows:

$$w(i, k) = \frac{constant(t) - sw(i,k)}{constant(t)}$$

where value of $constant(t) \leq$ Number of users in the chosen cluster.

- Finally compute the predicted ($p(i,j)$) value of item $j$ for active user $i$ from test dataset using Equation 5.7 given below:

$$p(i, j) = \frac{\sum_{k=1}^{Ni} w(i,k) \cdot rating(k,j)}{\sum_{k=1}^{Ni} w(i,k)}$$

where $rating(k,,j)$ refers to the rating of user $k$ over item $j$.

**Rating Prediction using multiple clusters:** This methodology (Bedi et al, 2009) utilizes more than one cluster to predict the ratings of the unrated items. It involves selection of top $nc$ clusters having maximum similarity with the active user. For each cluster, compute the rating quality ($Q_c$) of each item for the target user $i$ using members of the selected cluster. The function used to compute the ratings is as follows:
\[ \text{Quality}(Q_c) = \frac{(UB + \text{avgrating})}{2 \cdot UB \cdot \sqrt{\text{var}}} \]  \hspace{1cm} (5.8)

where \( UB \) is upper bound of the ratings, \( \text{avgrating} \) represents average rating of the item in the chosen cluster and \( \text{var} \) is the variance of ratings for the item, given by users in the chosen cluster. After computing the quality \( Q_c \) of each unrated item in the chosen cluster, the clusters in which \( Q_c \) for each item lies in the interval \((\text{maximum} - .005) \leq Q_c \leq \text{maximum} + .005\) are selected for predicting the ratings using Equation 5.9 as follows:

\[ \text{Rating} = \frac{\sum_{c=1}^{nc} (Q_c \cdot \text{avgrating})}{\sum_{c=1}^{nc} Q_c} \]  \hspace{1cm} (5.9)

where \( Q_c \) is the quality of item in selected cluster and \( nc \) is the number of clusters selected.

### 5.3.2 Evaluation of Nature Inspired Algorithm based Social Context Filtering Approach

To demonstrate the effectiveness of the presented approaches, experiments were performed on widely used Movie-Lens social filtering dataset collected by the GroupLens research project at the University of Minnesota. Dataset contained 100,000 ratings ranging from 1-5 for 1682 movies rated by 943 users. Evaluation of social context filtering systems depends upon the task for which they are used. The two main end user tasks in social filtering systems are Annotation in context and Find Good Items (Herlocker, 2004). Annotation in context refers to the task of annotating the items with prediction information that may be utilized by the users to decide which information to read or follow foremost. The second task provides users with ranked list of the items along with predictions for how much a particular user would like them. Performance of presented approaches is measured for both the end user tasks using the two most popular evaluation metrics:

**Decision support metrics** - These metrics measure the rate with which a social context filtering system makes correct or incorrect decisions about whether an item is good. Decision support metrics are suitable for the task of finding good items when users have true binary preferences. These parameters are used to determine the capabilities of the systems in predicting high quality items (that is the items that would be rated highly by the active user) as given in Equation (5.10), Equation (5.11) and Equation (5.12) respectively.
**Precision** = \((N \ast 100)/P\) \hfill (5.10)

**Recall** = \((N \ast 100)/R\) \hfill (5.11)

\[F1 = \frac{2 \ast \text{Precision} \ast \text{Recall}}{\text{Precision} + \text{Recall}}\] \hfill (5.12)

where \(P\) denotes the number of predicted ratings, \(N\) is the subset of \(P\) with number of predicted ratings greater than the average ratings and \(R\) is the number of ratings given by active user which are greater than the average ratings. The \(F1\) measure combines the precision and recall ideas of information retrieval for cluster evaluation (Kowalski, 1997). \(F1\) measure values lie in the interval \([0, 100]\). Larger values of \(F1\) measure indicate the better clustering quality. However these metrics do not directly measure the capability of an algorithm to accurately predict the ratings. Hence, statistical metrics are also used for evaluations.

**Statistical Accuracy metrics:** This metric is particularly important for evaluating the task in which predicted ratings are displayed to the user. For example MovieLens social context filtering system predicts the number of stars that a user may give to each movie and displays that prediction to the user. This metric establishes the statistical strength of the algorithm by comparing the estimated (predicted) stars against the actual stars given by the user to each item. Thus a social context filtering system may fail if the predicted ratings displayed to the user are incorrect even if it was able to correctly rank the items. The statistical accuracy metric namely mean absolute error (MAE) was used to compute the predictive accuracy of the presented approaches. Let \(\{a1, a2 \ldots an\}\) be the actual user ratings, \(\{p1, p2 \ldots pn\}\) be the predicted values of same ratings then the mean absolute error (MAE) is computed as given in Equation 5.13

\[|\bar{E}| = \sum_{i=1}^{N} \frac{|p_i - a_i|}{N}\] \hfill (5.13)

Using these evaluation parameters, all experiments were performed for five independent runs with following initial parameter settings- Training dataset: 80\%, Test dataset: 20\%, Population Size: 50, Number of Generations: 100, Number of memeplexes: 5 and Number of Memetic iterations: 16. The experiments were performed to analyze the effect of sparsity and variation in cluster size over the quality of items generated by the presented social filtering approaches as follows:
Evaluate the performance of MA based social (2D) filtering approach with respect to

- K-means
- Genetic algorithm

Evaluate the performance of SFLA based social (2D) filtering approach with respect to

- Memetic Algorithm
- Genetic algorithm

Evaluate the performance of MA based social (2D) filtering approach

Comparison of Memetic algorithm based 2D Social context filtering (MASF) approach with respect to K-means algorithm approach. This experiment (Banati and Mehta, 2010c), compares the decision support capability of MASF technique with respect to K-means approach. The values of precision, recall and f1 measure metrics are computed by varying the number of clusters(C) like C=10, 20 and 30 as shown in Figure 5.11, Figure 5.12 and Figure 5.13 respectively. In the figures K is used to represent results for K-means algorithm and M for Memetic algorithm based approach. Figure 5.11-5.13 depict that precision drops smoothly with the increase in number of relevant results or number of recommendation (N).

Figure 5.11 Comparison of MASF and KSF over Precision, Recall and F1 for C=10
This is expected as the average quality of the relevant content decreases with the increase in number of results to be displayed. On the contrary, recall increases with the increase in number of results from N=5 to N=30 as desired in the social filtering systems. Figure 5.11 and Figure 5.12 also depict that MASF performs better as compared to KSF for precision, recall and f1 measure. Thus MASF has more probability of providing high quality
items to the user as compared to KSF. However for \( C=30 \) both MASF and KSF show almost similar performance as shown in Figure 5.13. To verify this behaviour mean absolute error (MAE) was computed for \( C=10, 20 \) and 30. As shown in Figure 5.14, MAE for MASF is less than KSF for variations in the size of clusters. This shows that MASF has better predictive accuracy as compared to KSF for both the decision support metrics and statistical metrics. Error bars in Figure 5.14 illustrate that results are significant with 95% confidence interval.

![Figure 5.14 Mean Absolute Error of MASF and KSF over Number of Clusters](image)

- **Comparison of Memetic algorithm based 2D Social context filtering (MASF) approach with respect to Genetic Algorithm based approach (GASF).** In all the experimental evaluations M and G are used to denote memetic and genetic algorithm respectively. The following perspectives were identified to evaluate the ranking and predictive accuracy of both the genetic and the memetic algorithm ([Banati and Mehta, 2010d](#)):  
  - Variation in the number of clusters  
  - Effect of varying the sparsity levels in the dataset  
  - Impact of using more than one cluster for making predictions.
Variation in the number of clusters: This study evaluated the impact of varying number of clusters K=10, 20 and 30 on predictive accuracy. It was performed by developing the clustering model of users whose rated movies were greater than or equal to 30 in the MovieLens dataset. The results as shown in Figure 5.15, Figure 5.16 and Figure 5.17 depict the efficiency of precision, recall and f1 parameters over top $N$ number of predictions.

Figure 5.15 Comparison of MASF and GASF over Number of Clusters=10

Figure 5.16 Comparison of MASF and GASF over Number of Clusters=20
Figure 5.17 Comparison of MASF and GASF over Number of Clusters=30

Figure 5.15 and Figure 5.16 illustrate that MA based social filtering approach has slightly better precision, recall and f1 measure values as compared to GASF. On the contrary, for N = 30 as shown in Figure 5.17, GASF performs slightly better. To validate these results, mean absolute error was computed (as shown in Figure 5.18). These results depict that MAE of MASF ranges from 0.59 to 0.8 whereas for GASF, MAE is more than 0.83 for all values of N. Thus, for all values of N, MASF significantly outperforms GASF.

Figure 5.18 MAE of MASF and GASF over Number of Clusters
Evaluating the Effect of Varying the Sparsity Levels in the Dataset: This experiment measured the effect of sparsity on precision, recall and f1 metrics. Sparse datasets were generated by varying the users according to the count of their rated movies. Consequently only the users with rated movies count greater than threshold were considered for developing the model. The threshold values of 200, 100, 50 and 30 rated movies reduced the number of users (U) to 148, 361, 563 and 730 respectively as shown in Table 5.1. Sparsity of dataset is computed as follows:

$$\text{Sparsity}(\%) = \left( 1 - \left( \frac{\text{non-zero-ratings}}{\text{no-of-users} \cdot \text{no-of-movies}} \right) \right) \times 100 \quad (5.14)$$

<table>
<thead>
<tr>
<th>Sparsity Level</th>
<th>Number of movies rated</th>
<th>Number of Non-zero entries</th>
<th>Number of users</th>
<th>Sparsity in Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;200</td>
<td></td>
<td>43922</td>
<td>148</td>
<td>81.74</td>
</tr>
<tr>
<td>&gt;100</td>
<td></td>
<td>67584</td>
<td>361</td>
<td>87.78</td>
</tr>
<tr>
<td>&gt;50</td>
<td></td>
<td>88221</td>
<td>563</td>
<td>90.68</td>
</tr>
<tr>
<td>&gt;30</td>
<td></td>
<td>94849</td>
<td>730</td>
<td>92.28</td>
</tr>
</tbody>
</table>

Figure 5.19 Comparison of MASF and GASF over Data with Sparsity = 81.74%
Figure 5.20 Comparison of MASF and GASF over Data with Sparsity = 87.78%

Figure 5.21 Comparison of MASF and GASF over Data with Sparsity = 90.68%
Results (Figure 5.19 to Figure 5.22) for the datasets with varied sparsity levels illustrate that there is only trivial difference in the decision support accuracy of both nature inspired algorithms namely memetic algorithm and genetic algorithm. To substantiate these results statistical accuracy of the algorithms is evaluated using Mean absolute error (MAE). The MAE results obtained for sparsity = 81.74%, 87.78%, 90.68% and 92.28% as shown in Figure 5.23, precisely demonstrate that for all values of N, MAE is less for memetic algorithm. This confirms that MASF approach has more likelihood of providing good quality items to the user as compared to GASF.
Evaluating the Impact of using more than one Cluster for making Predictions: In this study, predictions for the active user were computed by selecting neighbours from the single cluster and from multiple clusters. Both the memetic and genetic algorithm were evaluated based on decision support metrics for making small and large number of predictions. Results as indicated in Figure 5.24 and Figure 5.25 illustrate that values of recall, precision and f1 metrics improve considerably for predictions made using more than one cluster.

![Figure 5.24 Comparison of MASF and GASF for N=15](image1)

![Figure 5.25 Comparison of MASF and GASF for N=30](image2)
However results also revealed that both algorithms exhibited comparable performance for all decision support metrics. To validate these results, mean absolute error (MAE) was computed as shown in Figure 5.26. Results demonstrate that in both the cases (predictions made using one cluster and three clusters) predictive accuracy of memetic algorithm is significantly better than the genetic algorithm. Results also indicate that with the increase in number of clusters used for predictions, error rate also increases. Thus, there is trade-off between the statistical accuracy and the decision support accuracy of the systems. Depending upon the priority or the requirement, decision may be taken to use one or more clusters for making predictions.

**Compare the Performance of Shuffled Frog Leaping Algorithm based 2D Social Context Filtering (SFLASF)**

This study evaluated the predictive accuracy of shuffled frog leaping based 2D social context filtering approach with respect to Memetic algorithm (MA) and Genetic Algorithm (MA) based social (2D) filtering system. To depict the results, abbreviations SFLASF, GASF and MASF have been used to represent SFLA, GA and MA based social context filtering approach. Previous study established that various nature inspired algorithms may show similar behaviour over precision, recall and f1 metrics. Hence, only MAE is used as evaluation metric henceforth. This study compares the performance of SFLASF, GASF and MASF over
- Sparse datasets
- Varying cluster size

**Comparison of SFLASF with respect to GASF and MASF over Sparse datasets:** This study analyzed the performance of SFLASF approach over sparse datasets. The results as observed in Figure 5.27 and Table 5.2 substantiate that statistically shuffled frog leaping algorithm provides more accurate results as compared to genetic and memetic algorithm based social filtering approach. Table 5.3 establishes that SFLASF is able to reduce the MAE upto 26% as compared to GASF and MASF. Thus SFLASF approach performs significantly better in augmenting the quality of retrieved relevant items as compared to memetic and genetic algorithm based approach.

![Figure 5.27 Predictive accuracy of SFLASF, GASF and MASF over Sparse Datasets](image)

**Table 5.2 Mean Absolute error of SFLASF, GASF and MASF over Sparse dataset**

<table>
<thead>
<tr>
<th>Sparsity (%) in dataset</th>
<th>GASF</th>
<th>MASF</th>
<th>SFLASF</th>
</tr>
</thead>
<tbody>
<tr>
<td>81.74</td>
<td>0.795404</td>
<td>0.78966</td>
<td><strong>0.534632</strong></td>
</tr>
<tr>
<td>87.78</td>
<td>0.789245</td>
<td>0.76157</td>
<td><strong>0.682486</strong></td>
</tr>
<tr>
<td>90.68</td>
<td>0.781023</td>
<td>0.718704</td>
<td><strong>0.53055</strong></td>
</tr>
<tr>
<td>92.28</td>
<td>0.750628</td>
<td>0.721085</td>
<td><strong>0.603247</strong></td>
</tr>
</tbody>
</table>
Table 5.3 Percentage Improvement in MAE using SFLA with respect to GA and MA

<table>
<thead>
<tr>
<th>Sparsity (%) in dataset</th>
<th>Performance Enhancement (%)</th>
<th>Performance Enhancement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SFLASF-GASF</td>
<td>SFLASF-MASF</td>
</tr>
<tr>
<td>81.74</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>87.78</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>90.68</td>
<td>25</td>
<td>19</td>
</tr>
<tr>
<td>92.28</td>
<td>15</td>
<td>12</td>
</tr>
</tbody>
</table>

Evaluation of SFLASF with respect to GASF and MASF over varied number of clusters: To further substantiate the effectiveness of SFLA based 2D social context filtering approach with respect to genetic and memetic algorithm, experiments were performed by varying number of clusters as shown in Figure 5.28.

Figure 5.28 Mean Absolute Error of GASF vs. MASF vs. SFLASF over Varied Number of Clusters
Table 5.4 Mean absolute Error values over Varied Number of Clusters

<table>
<thead>
<tr>
<th>No of clusters</th>
<th>GASF</th>
<th>MASF</th>
<th>SFLASF</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.790414</td>
<td>0.757426</td>
<td>0.663465</td>
</tr>
<tr>
<td>15</td>
<td>0.760224</td>
<td>0.709685</td>
<td>0.53055</td>
</tr>
<tr>
<td>20</td>
<td>0.75055</td>
<td>0.725107</td>
<td>0.532888</td>
</tr>
<tr>
<td>25</td>
<td>0.770613</td>
<td>0.678177</td>
<td>0.638702</td>
</tr>
<tr>
<td>30</td>
<td>0.792323</td>
<td>0.656624</td>
<td>0.580966</td>
</tr>
</tbody>
</table>

Table 5.5 Percentage Enhancement in SFLASF over GASF and MASF

<table>
<thead>
<tr>
<th>No of clusters</th>
<th>Performance Enhancement (%) SFLASF-GASF</th>
<th>Performance Enhancement (%) SFLASF-MASF</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>23</td>
<td>18</td>
</tr>
<tr>
<td>20</td>
<td>22</td>
<td>19</td>
</tr>
<tr>
<td>25</td>
<td>13</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>21</td>
<td>8</td>
</tr>
</tbody>
</table>

As illustrated in Figure 5.28 and Table 5.4, for the MovieLens dataset, best mean absolute error value obtained for GASF is 0.75 for n=20, for MASF best MAE is 0.65 for n=30 and for SFLASF, best MAE is 0.53 obtained for both n=15 and n=20. Overall, for all clusters(n) =10, 15, 20, 25, 30 as shown in Table 5.5, MAE of SFLASF is significantly less than Memetic and Genetic algorithm based social filtering. SFLASF model is able to reduce the MAE by approx 23% with respect to GASF and upto 19% with respect to MASF.

Thus, presented model based social context filtering technique using SFLA aids in enhancing the scalability of the system. However the issues of sparsity and cold start still remains. Cold Start problem relates to the situation where active users did not expressed ratings in the past. In practice, user’s rate very few items from the variety of items available and this results in sparse datasets. Increase in sparsity leads to decline in prediction accuracy of the approach. SF approach cannot compute relevant items under such cases and thereby compromises with the quality of recommended items. It has been identified that under such conditions, active user’s personal ‘Web of Trust’ (group of users whom active user has expressed his trust) can be used to compute the relevant items as discussed in the next section.
5.4 Trust Aware Social Context Modelling

Shneiderman (2000) defined trust as “the positive expectation a person has for another person or an organization based on past performance and truthful guarantees”. Thus, trust in other people is established on the basis of history of promises kept. Reliability of a person is judged based on the past experiences of that person with others or personally. People come across large number of persons with similar tastes and preferences but very few are trustworthy and become friends. Reliability of friends plays an important role to consider their preferences while taking decisions. In practice also, although the automated social context filtering (SF) technique resembles the manual filtering approach adopted by the users, the two approaches differ in their actual implementation as shown in Figure 5.29.

![Figure 5.29 Manual vs. Automated Social Filtering](image)

Automated SF approach considers the preferences of all users of the community whose profiles match with the active user profile, to compute relevant items. On the contrary, in manual SF approach only the preferences of the credible peers / true friends with similar interests and preferences are considered to make the buying decision. Thus the element of credibility popularly known as trust is ignored in conventional social filtering approach. Inferring relevant items considering all types of users whether friends or not, thus raises a question on the accuracy of conventional social filtering systems.
In recent years, trust has been explored as either implicit trust or explicit trust as shown in Figure 5.30. Yahalom et al (1993) distinguished trust as direct trust and indirect trust in distributed systems. Direct trust refers to the explicit expression of an entity to indicate trust over another entity about a particular subject.

Indirect trust is studied as propagated trust that is derived from the trust relationships between multiple entities connected in the network of ‘Web of Trust’. Buskens (2002) discussed the possible reasons for emergence of trust in social networks. It was pointed that explicit labelling of trust or distrust by a user over other users facilitates in augmenting the credibility of the system.

![Figure 5.30 Trust Modelling in Social Filtering Systems](image)

Avesani et al (2004) presented “Moleskiing”, a trust aware recommender system for the semantic web. The model allowed the users to explicitly express their views about travel trips as well as trust in other users. Based on this data, system provided reliable information to the users. Massa and Bhattacharjee (2004) developed a trust model based on user’s direct web of trust integrated with propagated trust to recommend the books, movie, music, software etc to on-line users. Study established that trust metric and similarity metric increased the coverage of recommender systems while maintaining the recommendation accuracy.
Hwang and Chen (2004) pointed that trustworthiness of a user may be computed implicitly based on the past rating behaviour of individuals. System inferred the trust score from the user’s rating data and established that incorporation of trust into social filtering process indeed improves the prediction accuracy of the system. Donovan and Smyth (2005) computed the degree of trust utilizing user-item rating matrix. Accordingly, trust weight was computed based on the number of true predictions made by the users. This approach did not require any explicit value of trust to be specified by the users. Papagelis et al (2005) highlighted that trust aware filtering approaches are seen as potential solutions to solve the cold-start and sparsity problems in social filtering systems.

These studies highlighted that implicit approaches do not require explicit value of trust; however reliability of implied approaches has always been debateable. Although it is difficult to formulate the explicit ‘Web of Trust’ as it requires extra efforts from the users, explicit trust is the best indicator of user’s true friend community. Thus in this work user’s explicit ‘Web of Trust’ is utilized to predict the relevant items to be recommended to the user.

### 5.4.1 Trust Aware, SFLA based Social Context Filtering

The social context filtering model developed in Section 5.3.1 (Figure 5.7) has been adapted to incorporate ‘Web of Trust’ as shown in Figure 5.31. In the presented trust aware social context filtering model (Mehta and Banati, 2012b), working of the system is divided into two phases. In the first/pre-processing phase a social context model of user’s historical preferences is developed offline i.e. social communities of the users are formulated on the basis of their rating behaviour. To develop the model, clustering of users is achieved by applying Shuffled Frog Leaping Algorithm. The subsequent online phase identifies the social community of the active user as depicted in Figure 5.31.
Figure 5.31 Trust Aware, SFLA based Social Context Filtering model

Figure 5.32 Community of active user
Relevant items are predicted based on the preferences of the trusted users (as indicated by the ‘Web of Trust’ of active user) in the selected community as shown in Figure 5.32. The figure depicts that though large number of users constitute the community of active user (as indicated by the connected lines), active user explicitly expressed trust over few members of the community (as indicated by arrow lines). Therefore in the next phase only trustworthy members participate in the process of computing relevant items for the active user. Relevant recommendations are generated by estimating the rating score that a target user may assign to an item. To predict \( P \) the rating scores \( P(i, n) \), that an active user \( i \) would assign to an item \( n \), a modified version of Resnick’s (Resnick et al, 1994) formula as shown in Equation 5.15 was employed. This formula allows only the trusted user’s to participate in the process of computing scores for the relevant items as follows:

\[
P(i, n) = \bar{p} + \frac{\sum_{j \in N_T(i)} r(n(j) - j) \cdot sw(i, j)}{\sum_{n \in N_T(i)} sw(i, j)}
\]

where \( \bar{p} \) is the average rating of the current user \( i \) over item \( n \), \( n(j) \) represents the rating of trusted user \( j \) over item \( n \) and \( N_T \) is a set of trusted users, \( j \) represents the average rating of user \( j \) and \( sw(i, j) \) is the similarity weight as depicted in Equation 5.3.

5.4.2 Evaluating the Effect of Incorporating ‘Web of Trust’ Information in SFLA based Social Context Filtering model.

To evaluate the performance of trust aware, SFLA based social context filtering system, Epinions dataset was used for the experiment. Dataset was obtained from Epinions website that provides an option for the users to express their personal ‘Web of Trust’ (i.e. the people whose reviews/ratings have consistently proved to be useful) and their web of distrust (i.e. the people whose reviews/ratings do match their preferences). Dataset contained approximately 50,000 users who rated a total of almost 140,000 different items at least once. The total numbers of items are around 660,000 and the total numbers of issued trust statements are about 490,000.
To evaluate the efficiency of presented approach, study involved only the users who rated more than 100 movies. Preferences of all the users were used for developing the clustering model using SFLA while only trusted users participated in the process of generating the relevant items. Thus, the approach recommends good quality items even if the active users have not rated/ rated few items in the past. This approach minimizes the cold-start and sparsity problems. Figure 5.33 depicts the MAE obtained by varying the number of profiles utilized for computing the recommendations. The results substantiate that trust aware technique provides more accurate results as compared to the conventional SFLASF approach.

![Figure 5.33 Relative accuracy of Trust Aware SFLASF with respect to conventional SFLASF](image)

**Table 5.6 Percentage Gains in Accuracy of Trust Aware SFLASF over Conventional SFLASF**

<table>
<thead>
<tr>
<th>Number of users</th>
<th>% Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>5</td>
</tr>
<tr>
<td>100</td>
<td>19</td>
</tr>
<tr>
<td>140</td>
<td>29</td>
</tr>
<tr>
<td>180</td>
<td>32</td>
</tr>
<tr>
<td>220</td>
<td>45</td>
</tr>
</tbody>
</table>
Table 5.6 illustrates the percentage improvement obtained by considering only the trusted peer network. It can be precisely observed from the Table 5.6 that presented trust aware SFLA based approach further enhances the accuracy of social context based deep web search systems. Since, the relevant items are predicted by considering the preferences of only trustworthy members of the community, approach depicts good performance even if the user has not expressed any interests in the past; solving the cold-start and sparsity problem. Results also establish that with the increase in number of users, gain is also enhanced. This trend depicts the potential for scalability of the presented approach.

The inherent limitations of social context filtering systems such as scalability, cold-start and sparsity are thus handled. These approaches used two types of entities, users and items, to develop the social context model and identify the most relevant items for the target users. However, the approaches did not take into account any additional contextual information, such as time, location, weather, demographic background or the company of other people etc to compute the relevance. In real world, interests and preferences of the users are never static, they change according to the variations in the demographic profile of the users e.g. age, gender and type of occupation they are involved.

During school life, people are more energetic and full of enthusiasm, thus may be more interested in adventurous or thrilling movies for example which may change to comedy movies as they become professional. Interests also vary according to the gender of the user. Nevertheless, most of the existing social context filtering algorithms assume that the users’ and the items’ characteristics are stagnant and ignore the demographic dimension of the user. While such assumptions are acceptable for relatively short period of time, for long periods, interests of the users change according to their lifecycle stages such as child, teenager, adult and old age. Thus, demographic context of the user plays an important factor to cope with the evolutionary nature of the user’s personal interest as discussed in next section.
5.5 Context Aware Social Filtering

Adomavicius and Tuzhilin (2001)(2011) emphasized the need to include user’s contextual information into the process of computing relevant items as explicit additional categories of data in social filtering systems. Accordingly the methodology to compute the rating function (R) is modified to incorporate the additional dimension of context as follows:

\[
\text{Contextual Rating Function}(R): \text{User} \times \text{Item} \times \text{Context} \rightarrow \text{Rating}
\]

where Context specifies the contextual information associated with the application. Thus, the contextual approach estimates the relevancy of items using multiple dimensions (3D) \( \text{User} \times \text{Item} \times \text{Context} \) instead of two dimensions as used in conventional systems (discussed in Section 5.3). Herlocker and Konstan (2001) pointed that in certain applications, quality of results from the social filtering systems may be improved by incorporating the knowledge about the user’s task (i.e. task context) into the search algorithm. Adomavicius and Tuzhilin (2005) presented three algorithmic paradigms to incorporate the contextual information into the filtering process. These include contextual pre-filtering (CPreF), contextual post-filtering (CPoF) and contextual modeling (CM). In all the three approaches, the process started with “Data” on items, ratings and contextual information (“Context”) about the users and results in generating context-specific items for the user. In CPreF, contextual information was used to filter out irrelevant ratings before they were used to compute ratings of relevant items using standard (2D) methods whereas in CPoF contextual information was used after the classical (2D) filtering technique was applied to the standard ratings data. In context modelling, contextual information was embedded within the algorithm used for filtering out irrelevant information/items.

Kenta et al (2009) presented a context-aware approach that considered both users’ current contexts and time series contexts. It was emphasized that users’ action patterns change according to their current contexts. Pedro et al (2010) used temporal context of the users to fetch the context relevant movies. Temporal context was taken into account using variations of the KNN algorithm. Results established that the usage of weekly timing information improved the quality of results and reduced the information overload of the search engine. Wang et al (2010) presented mood-based hybrid social filtering approach and
performed similarity fusion and predicted ratings fusion to improve the accuracy of these systems. Nathan et al (2010) developed time-frequency probabilistic strategies using simple KNN and a matrix factorization approach. It was established that temporal context information serves as a valuable source for the identification or differentiation of users with respect to traditional non-contextual strategies. Gantner et al (2010) used Pair-wise Interaction Tensor Factorization (PITF) technique to model the temporal (week) context for personalized tags in social websites.

Lee et al (2011) extracted user’s preferences from their log data and converted this implied contextual information into implicit feedback graph. Thereafter, random walks technique was applied on the transformed graph. The inclusion of multiple contextual dimensions in the algorithm improved the quality of results. Panniello et al (2012) evaluated the pre-filtering and the post-filtering approaches and established that contextual approaches performed better than non-contextual approaches.

Thus, it was established that contextual approaches yield better results as compared to conventional non-contextual filtering algorithms. However, majority of the contextual approaches as discussed above employ temporal or location context of the users to improve the quality of results. An important context of user’s demographic information also affects the efficacy of social filtering systems as discussed in next section.

5.5.1. Demographic Context Modelling in Social Filtering

Mehta and Banati (2012c) developed a model of demographic context aware social filtering system as depicted in Figure 5.34. The model utilizes agent technology to carry out the various tasks. It employs two types of agents to perform contextual social filtering - the active agents namely User interface (UI) agent and contextual retrieval (CR) agent and the passive agent namely context modelling (CM) agent. The Context Modelling (CM) agent is responsible to develop an offline social context model of community preferences using Shuffled Frog Leaping Algorithm. The work presents two distinct methodologies to incorporate the user’s personal demographic context in social filtering process that may be adopted by the CM agent. Figure 5.35 illustrates the presented SC2D (SFLA based contextual
two dimensional) approach and SC3D (SFLA based contextual three dimensional) approach along with the conventional 2D approach to social filtering.

![Diagram showing the process of contextual retrieval using the shuffled frog leaping algorithm and demographic context awareness in social filtering.]

**Figure 5.34 Demographic Context Aware Social Filtering Approach**

![Diagram showing context modelling approaches.]

**Figure 5.35 Context Modelling Approaches.**
In SC2D (SFLA based contextual 2D) approach, CM (context modelling) agent formulates the 2D social filtering model using SFLA based clustering (as discussed in Section 5.3.1). Subsequently, contextual retrieval (CR) agent utilizes the active user’s demographic context to generate context relevant content/items. Thus the similarity weight (\(sw(i,j)\)) (Section 5.3 Equation 5.3) has been modified to include the demographic context weight (\(cw(i,j)\)) as shown in Equation 5.17.

\[
csw(i,j) = sw(i,j) + cw(i,j)
\]  

(5.17)

CR agent uses the contextual similarity \((csw(i,j))\) weight to identify the contextual community for the active user. To compute the ratings and identify the context relevant results, CR agent follows the steps as discussed in section 5.3.1.

In SC3D (SFLA based contextual 3D) approach (Figure 5.35), CM agent utilizes the user’s demographic context to form the three dimensional (3D) contextual social filtering model using SFLA based clustering algorithm. The clustering model using SFLA is now developed using contextual similarity weight \((csw(i,j))\) (as shown in Equation 5.17) as per Section 5.3.1. Thereafter, contextual retrieval is performed using 3D contextual social filtering model by the CR agent. The User Interface agent (UI) interacts with the CM agent to identify the social community of the active user and subsequently CR agent retrieves the contextual results utilizing the 2D/3D SFLA based social filtering model developed by the CM agent. Both, contextual retrieval (CR) agent and user interface (UI) agent are responsible for performing the various dynamic activities involved in user system interaction.

5.5.2 Evaluating the Efficacy of Demographic Context Aware Social Filtering

To evaluate the effect of incorporating demographic context (SC2D & SC3D), Figure 5.35 in SFLA based social filtering technique; studies were performed using MovieLens dataset which provides demographic information about the users along with their ratings. These studies assess the performance of both the SC2D and SC3D approaches as follows:
Performance Evaluation of SC2D approach

In SC2D approach, SFLA based clustering model was developed using two dimensions only i.e. User * Item. Subsequently, demographic context of the users was utilized along with User * Item dimensions to compute the predictions. The study incorporated demographic context factors such as occupation, gender and integrated occupation and gender context. In the results as shown in Figure 5.36, Figure 5.37 and Figure 5.38, DG, DO and CC refer to SC2D approach with gender, occupation and combined (gender & occupation) context respectively and S2D refers to conventional SFLA based social context filtering technique.

Figure 5.36(a)  
Figure 5.36(b)  
Figure 5.36 Comparison of SC2D approach with ‘Gender’ Context vs. Non Contextual Social Filtering System

Figure 5.37(a)  
Figure 5.37(b)  
Figure 5.37 Comparison of SC2D approach with ‘Occupation’ Context vs. Non Contextual Social Filtering System
Fig 5.38 Comparison of SC2D approach with combined context vs. Non Contextual Social Filtering System

Results of precision recall and f1 as depicted in Figure 5.36(a), 5.37(a) and Figure 5.38(a) indicate that both contextual and non contextual approaches are equally suitable for the task of finding good items and in predicting high quality items that would be rated highly by the active user. However, for the task of predicting the ratings for the items to be displayed to the user, social context filtering with demographic context performs better than non-contextual approach as shown in Figure 5.36(b), 5.37(b) and Figure 5.38(b).

Among the various demographic context factors, integrated (occupation and gender) context contributes slightly more than occupation context in reducing the mean absolute error and significantly better than gender context. MAE establishes the statistical accuracy of the algorithm. Thus an approach may fail if the wrong predicted ratings are displayed to the user even if system was able to correctly rank the items. Therefore contextual approach is better suited as it is statistically more accurate as compared to the non-contextual social filtering technique. Results as depicted in Table 5.7 indicate that among the various demographic

Table 5.7 Percentage Improvement in Accuracy over Varied Demographic Contexts

<table>
<thead>
<tr>
<th>Context</th>
<th>MAE</th>
<th>% Gain w.r.t. conventional approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional</td>
<td>0.857</td>
<td>0.857</td>
</tr>
<tr>
<td>Gender</td>
<td>0.849</td>
<td>0.8</td>
</tr>
<tr>
<td>Occupation</td>
<td>0.809</td>
<td>4.8</td>
</tr>
<tr>
<td>Combined</td>
<td>0.81</td>
<td>4.7</td>
</tr>
</tbody>
</table>
contexts factors (gender, occupation and combined context), although combined context depicts slightly better performance, major contribution is of occupation context. Thus further studies are performed for occupation context only.

Examine the Effect of Varying the Sparsity Levels in the Dataset: It evaluates the performance of presented SC2D approach over sparse data sets. In this study demographic context factor i.e. occupation context that plays a major role in influencing the behaviour of user’s interests is studied over sparse datasets.

**Figure 5.39 Comparison of SC2D and S2D over sparsity = 81.74%**

**Figure 5.40 Comparison of SC2D and S2D over sparsity = 90.68%**
Figure 5.41 Comparison of SC2D and S2D over sparsity = 92.28%

Figure 5.42 Mean absolute error of SC2D vs. S2D over Sparse Datasets

Results of precision, recall and f1 measure as depicted in Figure 5.39, Figure 5.40 and Figure 5.41 indicate that both, SFLA based contextual and non contextual social filtering approaches are equally suitable for the task of ranking the items over varied sparsity levels of the data. However, social context filtering with demographic context outscores the non-contextual approach and lends itself better for predicting the ratings of the items to be displayed to the user. Figure 5.42 indicates improvement in mean absolute error thus establishing the statistical accuracy of the algorithm.
**Performance Evaluation of SC3D Approach**

To evaluate the effectiveness of SC3D (Figure 5.35), a 3-dimensional social context filtering approach, SFLA clustering model was developed using three dimensions i.e. $User \times Item \times Context$. Subsequently the demographic contextual model trained using SFLA was used to generate the Top N items and predict the ratings. The results as shown in Figure 5.43, Figure 5.44 and Figure 5.45, indicate that for all sparse datasets, both contextual and non-contextual social filtering approaches achieve recall upto 90% and precision upto 75%. Good value of F1 measure is attained even for small ...

![Graph 1](image1.png)

**Figure 5.43 Comparison of SC3D and S2D over sparsity = 81.74%**

![Graph 2](image2.png)

**Figure 5.44 Comparison of SC3D and S2D over sparsity = 90.68%**
number of predictions. However, the rating prediction capability of three-dimensional approach is significantly better than non-contextual 2D approach as it significantly reduces the mean absolute error as shown in Figure 5.46. Thus demographic contextual modelling is better suited for intelligent dynamic applications developed using social context based filtering techniques.
Relative Performance of SC2D, SC3D and S2D approach

This study analyses the relative performance of presented SC2D and SC3D approaches with respect to conventional social filtering technique. Results as shown in Figure 5.47 substantiate that presented context aware approaches to SFLA based social filtering are statistically more accurate as compared to non-contextual approach.

![Figure 5.47 Comparison of S2D, SC2D and SC3D approaches.](image)

Table 5.8 Percentage Improvement in Mean Absolute Error of SC2D and SC3D over S2D

<table>
<thead>
<tr>
<th>Sparsity</th>
<th>S2D</th>
<th>SC2D</th>
<th>SC3D</th>
<th>SC2D over S2D</th>
<th>SC3D over S2D</th>
</tr>
</thead>
<tbody>
<tr>
<td>81.74</td>
<td>0.857228</td>
<td>0.810481</td>
<td>0.776232</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>90.68</td>
<td>0.836574</td>
<td>0.814609</td>
<td>0.709268</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>92.28</td>
<td>0.778833</td>
<td>0.754555</td>
<td>0.583001</td>
<td>2</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5.8 depicts that SC2D approach improves accuracy up to 5% and SC3D technique augments significant gain of up to 20% as compared to non-contextual approach. It can also be observed from Table 5.8, that with the increase in sparsity, %gain of SC3D also improves. Thus, these results emphasize that demographic context is an important factor to determine
the interests and preferences of the users. Hence, presented demographic context aware social context based deep web search systems may provide better results to the users.

It was therein established that presented approaches overcome the inherent limitations (cold-start, sparsity and scalability) of content and social context based deep web search systems. In content based deep web search nature-inspired algorithm i.e. genetic algorithm tackles the cold-start problem prevailing in conventional systems. Thus the presented system serves the desires of both the existing as well as naive users. In social context based deep web search systems, it was observed that all nature-inspired algorithms (GA, MA and SFLA) perform equally well for decision support metrics like precision, recall and f1 measure. Nevertheless the mean absolute error of shuffled frog leaping algorithm is significantly (upto 26%) less than both the Genetic Algorithm and Memetic Algorithm based social filtering system. Similar behaviour was also observed between SFLA based contextual and non-contextual social filtering approaches. Inclusion of user’s ‘Web of Trust’ and knowledge of users’ demographic context further reduced the mean absolute error thereby enhancing the predictive accuracy of the system. To further enhance the performance of context aware web search systems, next chapter presents extensions of shuffled frog leaping algorithm namely Improved SFLA and Self-reformed SFLA.