A Dissertation on

RECOMMENDER SYSTEM BASED ON AFFECTIVE FEEDBACK INCORPORATING HYBRID OPTIMIZATION ALGORITHM

Submitted in partial fulfilment of the requirement for the award of degree of

Master of Technology

In

Information Systems

Submitted By:

Harsh Khatri

(2K13/ISY/07)

Under the Guidance of

Mr Rahul Katarya

(Assistant Professor, Department of CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

Bawana Road, Delhi – 110042

(2013-2015)

CERTIFICATE

This is to certify that **Harsh Khatri (2K13/ISY/07)** has carried out the major project titled "**RECOMMENDER SYSTEM BASED ON AFFECTIVE FEEDBACK INCORPORATING HYBRID OPTIMIZATION ALGORITHM**" in partial fulfilment of the requirements for the award of Master of Technology degree in Information Systems by Delhi Technological University.

The major project is bonafide piece of work carried out and completed under my supervision and guidance during the academic session 2013-2015. To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any degree or diploma.

(Project Guide)

Mr Rahul Katarya

Assistant Professor

Department of Computer Science and Engineering

Delhi Technological University

Bawana Road, Delhi-110042

ACKNOWLEDGEMENT

I take the opportunity to express my sincere gratitude to my project mentor Mr Rahul Katarya, Asst. Professor, Department of Computer Science and Engineering, Delhi Technological University, Delhi, for providing valuable guidance and constant encouragement throughout the project. It is my pleasure to record my sincere thanks to him for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Harsh Khatri

Roll No. 2K13/ISY/07

M.Tech (Information Systems)

E-mail: harshkhatri006@gmail.com

ABSTRACT

Websites have become more and more dynamic but they still lack intelligence. Although websites are able to mould themselves according to users' preferences and mouse clicks yet they cannot predict the content user might like, intelligently. The amount of data collected online by web-sites is increasing and therefore the demand for unique content by users is increasing every day. This need for self-organizing and transforming websites, to suit every customer's requirements has become a challenging problem. This work concentrates on solving the problem of creating relevant content for each user. To achieve the required flexibility we propose using mouse movements as a way of capturing the points of interest or the points where user had the most focus by capturing his mouse locations, since mouse pointer usually follows the eye trails for reaching the point of interest on website.

These mouse movements can be used for studying the patterns of user behaviour when exposed to a similar reference web-page by using a pattern analysis technique like Back-propagation neural networks, which enables using soft computing to recursively map users into groups/clusters with similar interests. These clusters can be related to the historical observations as documented in the well-known and referenced MovieLens database, dynamically, as new users provide rating to movies on the system. The correlation between these two systems can be attained by using Teacher Learning Optimization framework. The proposed algorithm therefore produces very effective results even on a cold start and produces linear precision.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

TAXONOMY

TLBO	Teaching learning based optimization
USER	A person or test program accessing the system
BPN	Back propogation network
CBRS	Content based recommender system
CFRS	Collaborative filtering based recommender system
RS	Recommender system
NN	Neural Network

Chapter – 1

INTRODUCTION

Since the creation of retail space manufactures have fought to create content or produce that can be consumed by maximum number of consumers in large quantities. In physical markets this is termed as capturing opportunity or sensing market, since the process of identifying product, following by production and distribution of goods takes a considerable amount of tie, producers have a large time margin to customize and fine tune the product and their production practices. But in the online world, new content is not as important as the right content, and no content is the right content for every user, therefore presentation of content on a website, even though considered the most important factor is presided over by the content that is to be presented.

Content selection [1, 2] is a process that has seen a very large number of paradigm shifts in history from the very basic html menu elements to moderns collection of up to a few hundred algorithms working together for predicting, what a user might like next. For example, Netflix [3].

Such systems are often termed as recommender systems[4], considerably capable of recommending users a content that is supposed to be appealing on the basis of their past experiences, current preferences or similarity derived from knowledge [5] with other users which are supposed to depict similar behaviours in contrast to the selections to be made.

In this research work a similar system is presented that can benefit from various factors like:

- User behaviour,
- Similarity with other users,
- Past experiences, and
- Predictive analysis.

Here, it is proposed that just like various inputs entered by user for choosing a category or genre of movies, his reaction to the visible quo of presented movies play an important role as well. The way a user spend move his mouse is a very important indicator of where the user was trying to focus his gaze and therefore studying the movements of mouse can give us a shallow idea of his affective response to the website content and highlights important parts.

At the same time the past ratings by the same user and other users can provide a very effective mean of grouping users based on their interest in a particular set of genre and movies as a unit. This grouping then can be used for predicting what movies can be liked by the user of same group.

Using the TLBO [6] modelling for weighted normalization of recommendations provide a good probable model with high consistency since the window of escalation is well defined yet user behaviour driven.

The following, chapter 2 documents the previous works done on the topics used as reference in this thesis work including, Content based Recommender systems, collaborative filtering recommender systems, Dynamic recommender systems, and affective reasoning based recommender systems. Chapter 3 lists a brief description of the Fundamental concepts used for designing the presented affective recommender system based on Hybrid optimization algorithms, which include soft computing , Neural networks, TLBO, et al. In chapter 4 I have explained the various components of the proposed system and the proposed algorithm. And last sections, 5 and 6 documents results and conclusion and future work, respectively.

Chapter - 2

LITERATURE REVIEW

2.1 RECOMMENDER SYSTEMS

2.1.1. CONTENT-BASED RECOMMENDER SYSTEMS (CBRS)

A content based recommender[7, 8] system treats every user as a separate entity which is independent of other users, every entity/user is represented by a set of characteristics and features. Profile of every user is created depending upon the the desirability and feedback that were used and presented to the user for recommendations. A set of filtering rules for various components are used, which takes in account the generated user profiles for producing recommendations. Every time a user interacts with the system, his feedbacks are recorded in the form of ratings, comments etc. which enables the system to learn about user's preferences and corresponds to the addition or change in the user profile, which is again used for recommendations at a latter time. Due to the advantages involved, CBRS is a totally independent system which relies on active individual users for building their own feature sets, often termed as profiles, and do not rely on the feature sets of other users present in the system.

CBRS therefore have high degree of transparency, since it clearly lists all the features used for finding recommendations and hence provides explanations with the recommendations. Popular shopping sites like Amazon.com[9] lists a set of items recommended to the user along with a list of items that led to the presented recommendations. Amazon often lists the items present in the shopping cart and along with them lists a set of related articles lso depending upon the past history of the user.

If a user had purchased a camera, most of the recommendations can be from the items related to that camera, including lenses, accessories, photo printers, et al. Interesting feature of a CBRF is that , when a new item in included in the system, features extracted from the data sheets , and therefore the items holds a strong chance to be recommended to the users that have a known interest in related items.

COLLABORATIVE FILTERING RECOMMENDER SYSTEMS (CFRS)

Just as recommendations produced by CBRF[10, 11] are dependent on a single individuals preferences, in a CFRS recommendations are made in a totally different way, i.e. preferences are base on the likings of similar users. This works in a perfect fashion as it can since CBRS cannot handle non-desirable items such as digital media like videos and music which can be effectively mapped with CFRS [12]. Without many difficulties. This feature gives CFRS an advantage over other recommender systems and makes it one of the most popular and most used recommender system. CFRS take decisions based on similarity and a number of methods such as pearson'r correlation or cosine angle are mainly used in grouping item and users. A typical CFRS starts with using simple measures like cosine[13] to calculate the similarity among items that have been purchased by a user.and creates an item-item relationship matrix. A combinations of many algorithms including genetic algorithms, Bayes classifiers, etc are utilized for ascertaining the item-user relationships. Genetic algorithm[14] or a combination of other algorithms is used for forming cluster of users with identical choices and Bayes classifiers are thus used for forming clusters of system users wit h items. The recommendation system therefore takes similarity choices clusters produced earlier, associate them with user profiles using associated rules defined in the system. User-user relationship is then defined with step wise models to form indexed categories. In the indexes, or tree like structures, every node contains a cluster with similar preferences. Recommendations are then created based on the correlation among items, users and user-item clusters. CFRS might be seen as privacy concern. Other problems that can be seen in CFRS implementations include, grey sheep problem[15], update frequency problem, cold start problem[16,17], scalability problem[18], etc.   CFRS records user's behavior and find out patterns.

Even though the sole purpose of the system is to facilitate user's, it is not guaranteed that users will be comfortable with that, since data like browsing history, etc can contain private information a user do not want to share. It is highly likely that users may not utilize the recommendation system much, just as it is possible that the system might not be able to able to suffice the amount of information needed on user's behavior for providing him very accurate predictions.

Cold start problem is frequently faced when the system has a new item or a new user. A new item may not be rated or labeled. Such items are easily abandoned in the commendation process due to the lack of association with other well rated and labeled items. In the case of a new user, the system may present poor accuracy in recommendations due to, once again lack of association.

2.1.2. DYNAMIC RECOMMENDER SYSTEMS

Recommender systems are defined as the system in which "people provide recommendations as inputs, which the system then aggregates and directs to appropriate recipients". A number of other definitions are also given by various other researchers, each of which emphasize on RS as a tool for assisting users in finding relevant information. In this paper, a new category of RS is sketched that are dynamic in nature. Dynamic nature is itself a combination of many factors where every parameter is important for building a standalone system. DRS[19] is a system that is capable of recording changes that happen in the user sphere and the complete system sphere , side by side it records the environmental changes that occurred implicitly or explicitly and the recommendations are accordingly modified. Recommender system can be seen analogous to an advisory circle of friends that helps each other making choices in life. Taking into account user dynamics into RS may require incorporating human psychology that could play a very important part in developing such recommender system that are close to real life advisors. The human mind is extremely complex and difficult to interpret; still there are various personality attributes that could help in revealing human behavior that could then be implemented in recommender system. A novel methodology of making recommendation process more transparent as well as similar to real life recommendations should invariably involve taking into account psychological factors like trust and social network of the user. This could also be one of the key areas of working for the next generation recommender system. In addition, system side also registers a number of changes in terms of its content and goes through phases of updating and evolution. Such changes also affect the type of recommendation provided to the user and can be implemented in the RS. However, the changes in the recommender systems behavior go beyond temporal factors and involve context[20], novelty[21], serendipity[22], real-time[23] dynamics as well as diversity[24]. Each one of these parameters contributes to the dynamic behavior of DRSs. As such Recommender Systems that are involved in dealing with any one of the above parameter are stated as DRSs.

2.1.3. AFFECTIVE RECOMMENDER SYSTEMS

As studied and documented in [25] , affective labeling is supposed to be helpful in recommender system applications. A number of related works include (i) non-affective recommender systems, (ii) Affective labeling based recommender systems, and (iii) various techniques used for affective labeling. This proposed work further advances the affective system to mouse movements and relates it with historical data using TLBO for dynamically covering a wider data set range and improving diversity and mitigate sparsity. Movie recommender systems have been gaining much attention after the Netflix challenge to design a better then before recommender system which had cash price associated and therefore gained attention of the computer programmer comunitites world wide. For a long time research in RS have been focused around open datasets [26] In several case studies [27] , just the movie genre was used as the core lable, taken from the IMDB database for generating user preferences. Whereas the research work by Knavery implemented a CFRS for calculating predictions. The problem associated with all such labeling schemes is that a large part of variance in user interest is not captured by solely using content labels, Therefore the recommendation accuracy is very low. With advancements with time more and more user-oriented labeling approaches have been used . Gonzalez et al, have proposed a framework for exploiting emotions as a parameter for such systems, in a search for providing more genuine results [28]. Some systems document a various number of inputs that can be used for producing affective recommendations including [29] , which documents using keyboard mouse and touch screens for as a method for recording affective feedback from the user .

Affective recommendations is a new area of research, due to which it have high novelty but this makes it difficult to (1) performing in real time, (2) reliable recognition accuracy and (3) applicable on most of available computers.

To solve this challenge we have proposed using just the mouse movements which are going to be present over any system that have a browse-able interface.

2.2 Issues Faced by recommender systems

2.2.1 COLD START

Cold- start [30] is a state where a system do not have sufficient data for making a decision, such state occurs in RS when a new user enters the system, or a new item is added to the list of recommendable items with low levels of information. Cold-start problem can further be distinguished between a number of categories: new item anomaly, new user problem, and community problem. New user is the most common problem resulting in a cold-start scenario. The new community problem [31] refers to the difficulty when a recommender system is started up for the first time and requires some amount of time to capture reliable amount of data . A few common ways used for tackling this problem are to make the recommendation based on different means, i.e. by taking CF –based recommendation only when enough number of users ad ratings are present. The new items problem [32] is faced because new items entering a recommender system do not have enough information attached to them, which can be indexed for prepairing recommendations. Therefore an item which is not properly indexed, is not recommended often and goes un noticed by the users and since they are not aware of the item , it is not rated by the user, which will lead to a circle where no item will impact the performance of the system and the recommendation system will be stagnant. This problem do not have high impact when the items associated can be discovered by a large number of means for example movies, etc. A popular solution is to employ people to explicitly rate new items entering the system. The new user problem [33] represents the biggest difficulty in the RS operations. As the new user entering a RS have not yet provided any ratings or do not have any historical data associated with him, his recommendations cannot be personalized to him.

Once a user enters the system for first time and select ratings , a RS offers him personalized recommendations, but the number of ratings is not sufficient to be able to make reliable recommendations., there new users might view RS as not offering services which have been expected by them and might very well even stop usinf the recommender system. One common strategy is to collect user data from various sources, right when the user eters the system for the fist time , for example data from facebook, twitter, etc [34], and provide him recommendations based on the historical activity of the user and not just on the bases of ratings, that user have provided in the system. Cold – start problem is mostly tackled using hybrid approaches [35]. Leung et al. [36] presented a content focused hybrid novel approach which uses cross level rules for associating information about content of domains items. . Kim et al. [37]
Has employed collaborative tagging as an method for collecting and filtering preferences of system users for various items and they have explored the use of collaboratively tagged data for reducing sparseness and users entering the system in a cold start state. The had collected the data by crawling the delicious site which tag data collaboratively. They also use very implicit association among preferences given by users to various items and the additional preferences given to taxonomy for making better recommendations, which are also good in quality as well as mitigate the problem of cold start . Fig 1. show a scenario where two users are recommender different levels of recommendation based on the time of their entry in the system, as the new users are presented with very crude and low precision recommendations, whereas the precision of recommendations made to an old user are supposed to be high by the virtue of information available about him.

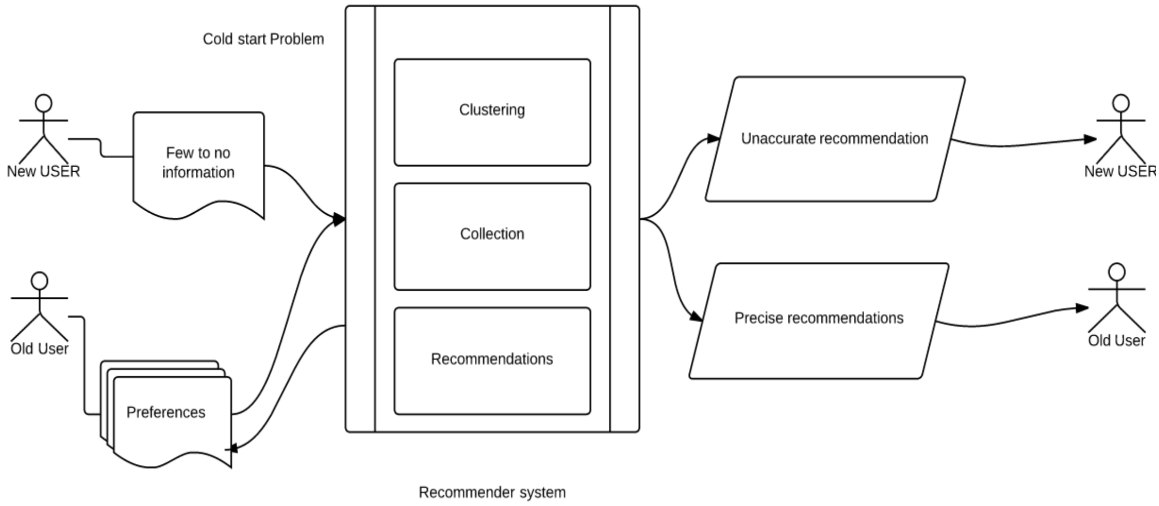


Fig 1. Cold start problem Generalized structure of a

2.2.2 LONG TAIL SEGREGATION

A the name suggests, Long Tail segregation[38] deals with the spreading of data in a long pattern where a major part is constituted as tail, or less visible data. Which means that a number of items are popular and presented to users, multiple times, whereas other items stay dormant and due to their location in heavy tail, do not sell well. The long tail include possibility to discover ad analyze a vast of data using personalised filters in recommender systems. The world has been revolving around the Hit and miss form of categorization, due to the self space limitation. In a movie store the movies placed in teh front rows are often the most sold movies, not the main factor is, if the movies were high sellers because they were placed in the front rows or , they were placed in the front row, because they were already high sellers. This situation clearly references a Long tail segregation scenario since, if the movie was sold because it was placed in the first row, it clearly dwarfed the sales of other movies with higher potential, that were stored in the deeper rows, and the movies should be rotated. But if the movie was placed in the front row because it was already the best seller, then it should not have been in the front row as people already liked it and they would have gone to the deeper rows more frequently for the movie, increasing the chances of discovering a new movie in the behind rows, hence creating the sale of a new item, this phenomenon dictates that none of the movies should be in the front row, which is clearly the true essence of the recommender system. As it is to be attached with time since the movies should be rotated frequently to keep a larger portion of movies in focus.

2.3 .*DATA SPARSITY*

Recommender systems, providing users with personalized recommendations from a plethora of choices, have been an important component for e-commerce applications to cope with the information overload problem. Collaborative filtering (CF) is a widely used technique to generate recommendations. The basic principle is that recommendations can be made according to the ratings of like-minded users. However, CF inherently suffers from two severe issues, which are the problems targeted in this research.

• Data sparsity[39] refers to the difficulty in finding sufficient reliable similar users since in general the active users only rated a small portion of items;

• Cold start refers to the difficulty in generating accurate recommendations for the cold users who only rated a small number of items.

Lacking sufficient ratings will prevent CF from modelling user preference effectively and finding reliable similar users. In particular, the rating scarcity of recommender systems is usually up to 99% and cold users have rated less than five items in general.

One of the resultant issues is the design of similarity measure to better model user correlation, especially in the cold conditions where only few ratings are present. To address these issues, two general strategies have been observed in the literature. The first strategy is to incorporate additional (user or item) information to help model user preference, and the second is to propose new similarity measures and make better use of existent user ratings. Although up to date many algorithms have been proposed, these issues have not been well addressed yet.

PROGRESS TO DATE

We have conducted two different works following the general strategies mentioned in the last section. Specifically, we propose a trust-aware CF method to incorporate social trust1 which is strongly and positively correlated with user similarity. In addition, we also propose a novel Bayesian similarity measure by taking into account both the direction and length of rating profiles whereas traditional methods only consider the direction of rating profiles.

Chapter – 3

FUNDAMENTALS

3.1. SOFT COMPUTING

Soft computing[40] is used to solve NP-complete problems. NP-complete problems do not have any known algorithm for the optimal solution that can be executed in polynomial time. The soft computing is different from hard computing due to its tolerance of imprecision, the solutions can be uncertain and approximate to the optimal solution.

Several algorithms or tools have been introduced in the past that solve NP-complete problems that result in uncertain and unpredictable solutions. Most of them are based on natural biological and physical processes. These include –

- Neural Network
- Back propagation network

3.2. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Networks[41] are computational tool based on soft computing techniques. They model the interconnection of neurons in the nervous system of the human body and other organisms. The basic components used are called neurons. These neurons are trained with the training dataset. The neural network is a non-linear processing system that can be used to solve the clustering or classification problems. The neurons are arranged in a number of layers. The first layer is the input layer. It is the one that has to interact with the environment or the dataset. The last layer is the output layer. It is the one that present the processed data or the classes or clusters. The layers between the first and last layers are called hidden layers. These layers neither interact with environment nor present the processed data. Increasing the number of hidden layers and the number of neurons in each layer increases the computational complexity. The neurons in the all the layers are multiple-input and multiple output systems. The receive signals from all the neurons in the previous layer and process those signals using a non-linear function. The resultant signal is transmitted to all the neurons of the next layer. For the input layer neurons, the input dataset is the received, processed and transmitted to next layer. For all the hidden layers, the input is the outputs of all the neurons of the previous layer which is processed and transmitted to next layer. For output layer neurons, the input is outputs of all the neurons of the previous layer which is processed. The output is the membership value of the input in the classes defined by the neurons of the output layer.

3.3. BACK-PROPAGATION NETWORK

The back-propagation algorithm is used in the multi-layer neural networks [42] to classify the input dataset in different classes. While training the neural network, each sample of the dataset is input to the network, processed and output is generated. The generated output is then compared to the expected output and error is calculated. The error is then propagated backwards i.e. from output layer to the input layer while adjusting the weights between connected neurons. The sample is repeatedly presented and the weights are repeatedly adjusted until the error value is minimised.

Initially, the number of layers and the number of neurons in each layer is set. The number of input layer neurons is equal to the dimensions of the input dataset. The number of output layer neurons is equal to the number of classes in which the dataset has to be classified. The number of hidden layers and the number of neurons in those layers are chosen depending on the problem. The weights between the connected neurons are initially set to a random number or zero.

In each neuron, for processing the inputs, an activation function is used. Without the activation function, the back-propagation neural network would just be a perceptron. The activation function takes the sum of weighted inputs as input and generates the output signal. This function can be linear, threshold, exponential or sigmoid function. Fig.2 documents a generalized structure of a BPN with i, number of input layer neurons, j number of hidden layer neurons and k, number of output layer neurons. The BPN transform i number of inputs to k number of outputs depending on the design purpose and the data the BPN was trained with .

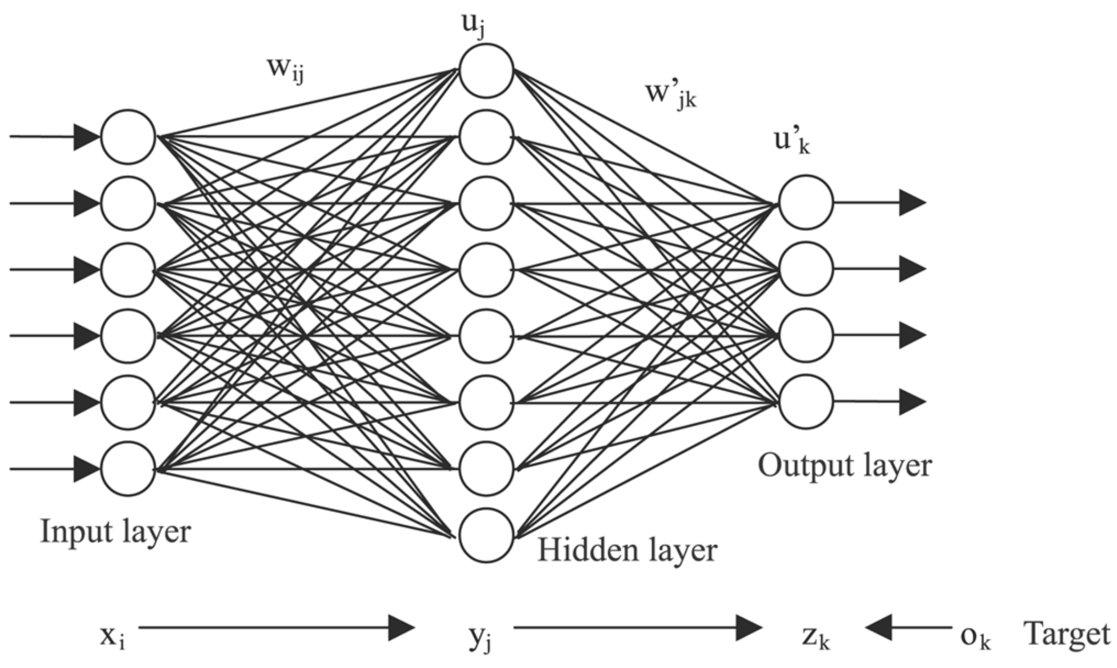


Fig. 2 Generalized structure of a BPN

The algorithm consists of 2 steps –

- Forward Pass
- Backward Pass

3.3.1. FORWARD PASS

The inputs are introduced to the network. At input layer, using the inputs, weights and the activation function, the outputs for each neuron are calculated. These outputs are then used as inputs for the next layer neurons whose outputs are used for its next layer neurons and so on. When all the hidden layer neurons have been processed, the output layer neurons calculate the final outputs for the input data. The calculated values show the membership of the input in the corresponding class. Since the neural networks learn using the supervised learning, each input sample already has an expected output. Using the given expected output and the calculated actual output, the error is calculated. The calculated error will then be used in the backward pass.

3.3.2. BACKWARD PASS

Once the error has been calculated, it will be used in backward propagation and adjusting the weights. The error is firstly propagated from output layer to the last hidden layer. At this point, the change in weights for each last hidden layer neurons is calculated. These values will then be propagated to previous hidden layers and the weights will be updated. The backward propagation will be continued until the first hidden layer.

3.4. TEACHING LEARNING BASED OPTIMIZATION

The Teaching Learning Based Optimization[43] method is based on the effect of the influence of a teacher on the output of learners in a class. Here, output is considered in terms of results or grades. The teacher is generally considered as a highly learned person who shares his or her knowledge with the learners. The quality of a teacher affects the outcome of the learners. It is obvious that a good teacher trains learners such that they can have better results in terms of their marks or grades. The process of TLBO is divided into two parts. The first part consists of the 'Teacher Phase' and the second part consists of the 'Learner Phase'. The 'Teacher Phase' means learning from the teacher and the 'Learner Phase' means learning through the interaction between learners.

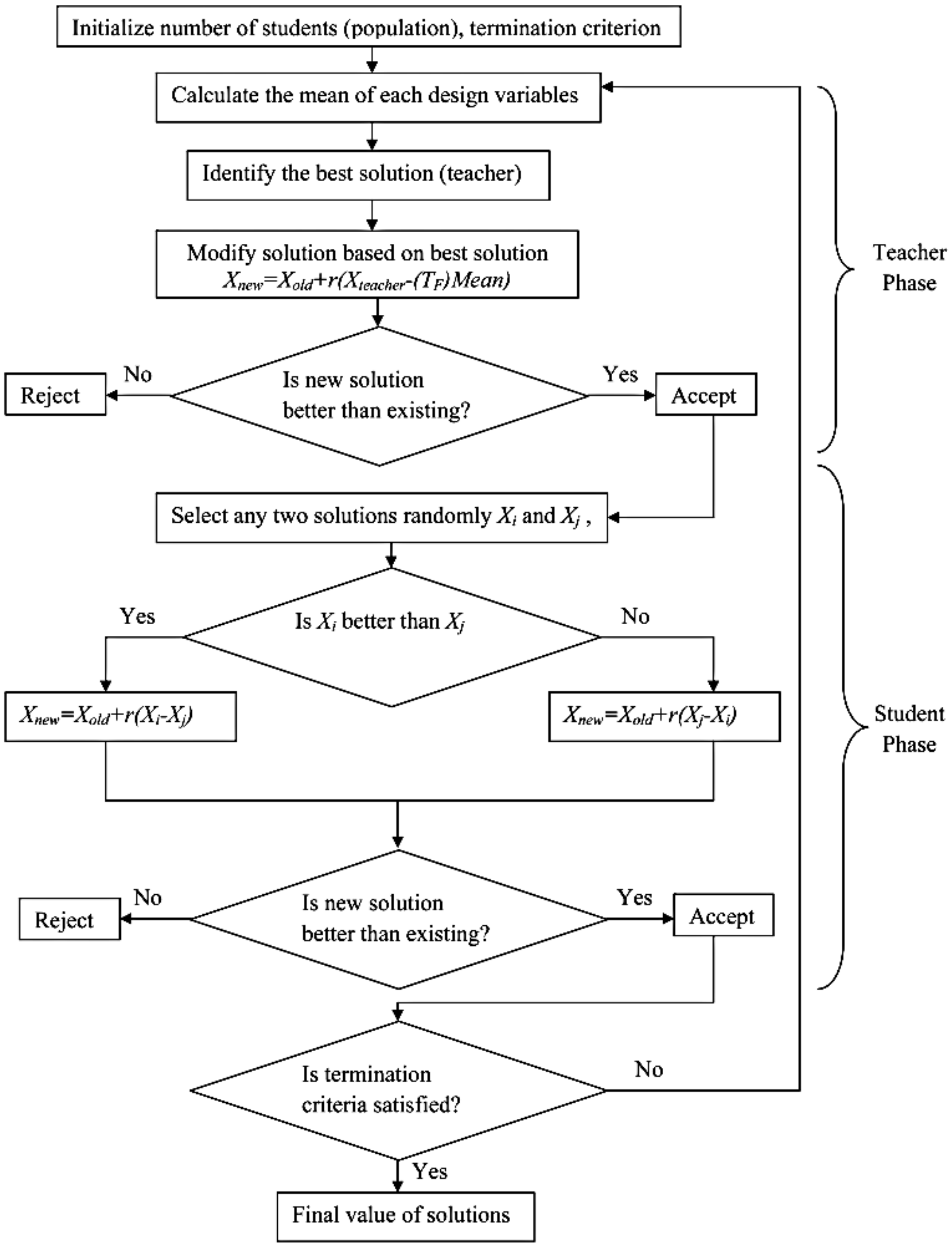


Fig. 3 Generalized TLBO implementtion

3.4.1. TEACHER PHASE

The mean of a class increases from MA to MB depending upon a good teacher. A good teacher is one who brings his or her learners up to his or her level in terms of knowledge. But in practice this is not possible and a teacher can only move the mean of a class up to some extent depending on the capability of the class. This follows a random process depending on many factors.

Let M_i be the mean and T_i be the teacher at any iteration i. T_i will try to move mean M_i towards its own level, so now the new mean will be T_i designated as M_{new}. The solution is updated according to the difference between the existing and the new mean given by

$$Difference_{Mean_i} = r_i(M_{new} - T_F M_i) \quad (6)$$

where T_F is a teaching factor that decides the value of mean to be changed, and r_i is a random number in the range [0, 1]. The value of $T_F$can be either 1 or 2, which is again a heuristic step and decided randomly with equal probability as

$$T_F = round[1 + rand(0,1)\{2 - 1\}]$$

This difference modifies the existing solution according to the following expression

$$X_{new,i} = X_{old,i} + Difference_{Mean_i} \quad (7)$$

3.4.2. LEARNER PHASE

Learners increase their knowledge by two different means: one through input from the teacher and the other through interaction between themselves. A learner interacts randomly with other learners with the help of group discussions, presentations, formal communications, etc. A learner learns something new if the other learner has more knowledge than him or her. Learner modification is expressed as

- For $i = 1: P_n$

- Randomly select two learners X_i and X_j, where $i \neq j$
- If $f(X_i) < f(X_j)$
  - $X_{new,i} = X_{old,i} + r_i(X_i - X_j)$
- Else
  - $X_{new,i} = X_{old,i} + r_i(X_j - X_i)$
- End If

➢ End For

➢ Accept $Xnew$ if it gives a better function value.

The pseudo-code of TLBO –

Step 1. Define the optimization problem and initialize the optimization parameters.

Initialize the population size (P_n), number of generations (G_n), number of design variables (D_n), and limits of design variables (U_L,L_L). Define the optimization problem as –

Minimize $f(X)$.

Subject to $Xi \in xi = 1,2,\ldots,D_n$

where $f(X)$ is the objective function, X is a vector for design variables such that $L_{L,i} \leq X_i \leq U_{L,i}$

Step 2. Initialize the population.

Generate a random population according to the population size and number of design variables. For TLBO, the population size indicates the number of learners and the design variables indicate the subjects (i.e. courses) offered. This population is expressed as population –

$$\begin{bmatrix} x_{1,1} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{P_n,1} & \cdots & x_{P_n,D} \end{bmatrix}$$

Step 3. Teacher phase.

Calculate the mean of the population column-wise, which will give the mean for the particular subject as

$$M_D = [m_1, m_2, \dots, m_D]$$

The best solution will act as a teacher for that iteration

$$X_{teacher} = X_{f(X)=min} \tag{8}$$

The teacher will try to shift the mean from M_D towards $X_{teacher}$ which will act as a new mean for the iteration. So,

$$M_{new_D} = X_{teacher,D} \tag{9}$$

The difference between two means is expressed as

$$Difference_D = r\left(M_{new_D} - T_F M_D\right) \tag{10}$$

The value of T_F is selected as 1 or 2. The obtained difference is added to the current solution to update its values using

$$X_{new,D} = X_{old,D} + Difference_D \tag{11}$$

Accept X_{new} if it gives better function value.

Step 4. Learner phase.

As explained above, learners increase their knowledge with the help of their mutual interaction.

Step 5. Termination criterion.

Stop if the maximum generation number is achieved; otherwise repeat from Step 3.

Chapter – 4

Proposed Recommender System

The algorithm proposed here for recommending movies to users over a web interface that is documented in these thesis targets three major problems faced by almost every recommender system, namely, Cold start, Long tail segregation and data scarcity. The proposed algorithm consists of two different modules combined together using teacher learning based optimization, namely offline module and an online module.

User is presented with an interface consisting of a number of movies and is allowed to check out the movies and selects the movie that seems most appealing to the user. During this course of navigation the user uses his mouse to point on the movie he finds most interesting, i.e. the field of interest. This movement of user's mouse pointer across the web page is captured by a JavaScript system which saves the movement data in a database that is further used for creating movement maps, which are the images that show the users movement across the page. This captured mouse map is then used for training, testing and finally implementing a clustering system based on neural network to cluster users with similar movement patterns together. These clusters are used for categorizing users so that they can be recommended movies using legacy, birds of same feather flocks together strategy.

When a new user comes into the system his mouse map is used for giving him membership in one of the ten clusters used for online clustering. Then the users are mapped to the offline clustering system, which is a BPN that clusters the historical user data from MovieLens database and is used as an historical reference for the recommendation of movies to users that are present on our recommender system. The following segments detail the

TLBO is then applied to calculate the most probable recommendation to the user using fuzzy mapping with the memberships and movie ratings. Fig. 4 Shows a general structure of the recommendation system that we have proposed in this thesis. The system is divided into three parts i) online interface for data collection and presenting recommendations ii) offline system for historical references and improving recommendations iii) recommendation system that utilizes Tlbo for producing recommendations, as described in the following sections.

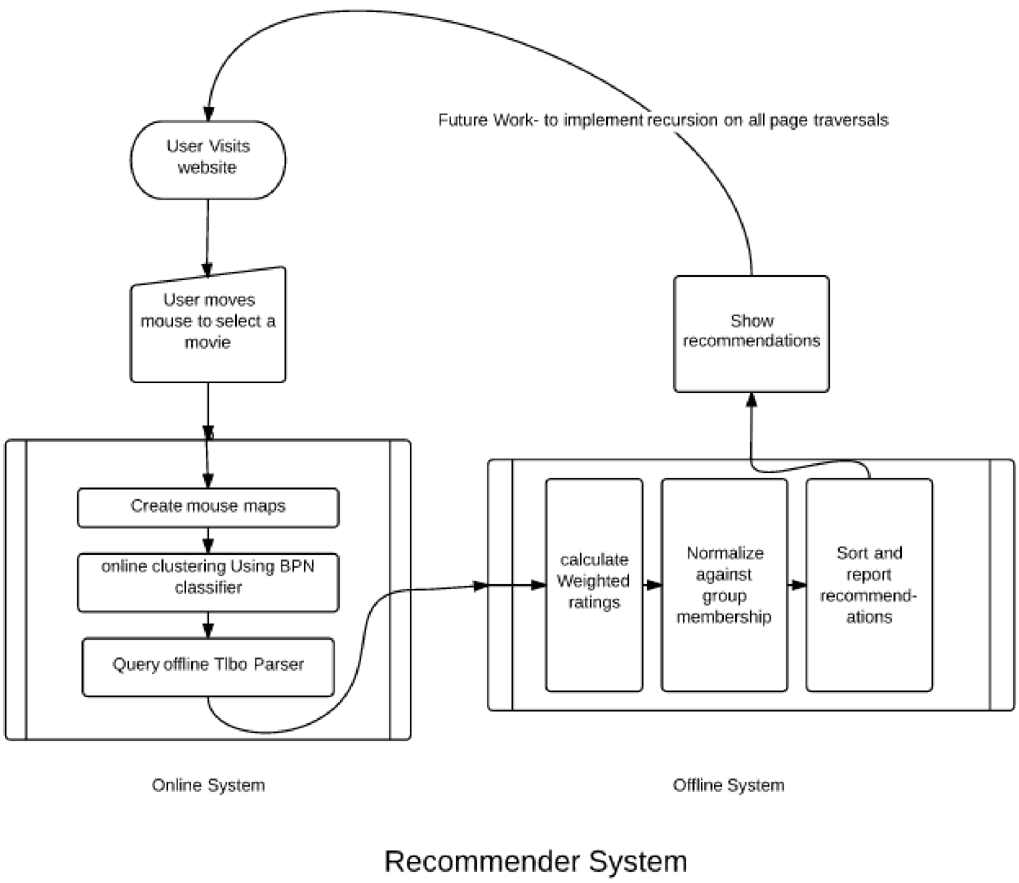


Fig. 4 Proposed recommender system

Different parts on the proposed recommender system, namely :

A. Online System

B. Data collection System

C. Offline System, and

D. Pseudo code,

Of the proposed system are documented in teh following sections.

4.1.1 ONLINE SYSTEM

Online user interface is designed to contain Webpages, listing movies from the movie database MovieLens and every user is asked to find a movie he/she likes the most on that page.

The basic design objectives are:

- To get user to explore movies from available database
- To allow user to rate movies he likes
- To present recommendations to the user

When users enters the system, he is given a userID based on his status, i.e if he is a new user or not, the id is then stored in a cookie at the user's browser to identify him in future. The system is designed so that the mouse movements of user when browsing through the listings are recorded and stored in a database, which is further used for preparing mouse maps, that is, a bitmap image of mouse movements across the webpage. These mouse maps are then utilized for training a neural network which is trained over 80 such mouse maps to classify them into groups based on the similarity. The used neural network is a general purpose image classification neural, and have a implementation similar to the BPN described in previous sections. Fig xx show a demonstration of the online interface , containing a number of movies selected from MovieLens database.

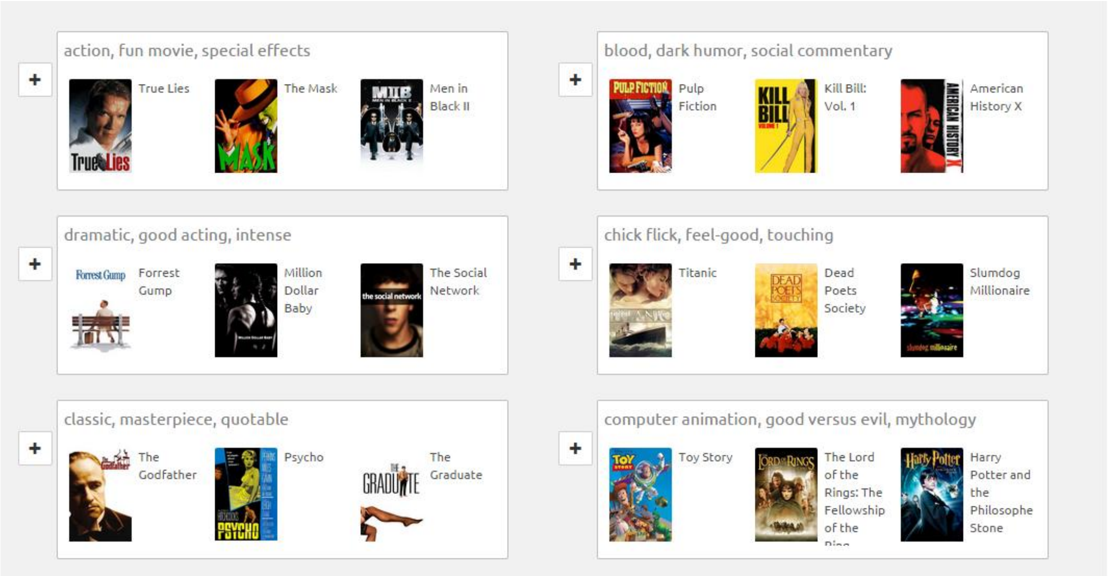


Fig 5. Online interface

4.1.2 RATING SYSTEM

Every movie is assigned a rating on the scale of 1 to 5 in the MovieLens database and the same rating system is used for ratings in our proposed algorithm this keeps the online module in synchronization with the historical MovieLens database, this provides inter compatibility between the two systems and allows for the proposed algorithm to be implemented, without any conflict on the basis of difference in newly generated and historical database.

Rating flow is divided in two important subparts.

- Rate every selected movie

  User is asked to rate every selected movie so that his selection can be mapped to the offline system containing historical MovieLens database statistics and further for clustered using a neural network.

- Feed to the offline network for reference

  The acquired ratings are pushed to the offline network so that the BPN can be updated to accommodate new user entering the system.

4.2 DATA COLLECTION

4.2.1 ONLINE DATA

As described in the above offline and online models, User is presented with a webpage where he can browse movies presented to him and his mouse movements were recorded using JavaScript to be sstored in a database to create image maps as shown in the following table 1 .

The recorded mouse movements tracks the location of pixels visited by the user, therefore the size of the mouse map is highly dependent on the resolution of the user's display and it have to be scaled down to a standard size as shown in Fig. 6. Which I have taken as 256*256 since this size is sufficient for storing enough level of details as required by the BPN network for classifying the user in a cluster.

Fig 6. Mouse movement tracking

4.2.2 Justification for scaling

Since a neural network infrastructure is used for clustering users, the approach in its essence is probabilistic which makes the horizontal or vertical scaling of the patterns unique,i.e the change in the patterns due to horizontal and vertical compression is insignificant and the original pattern still remains identifiable into clusters therefore, making the un-rationalized compression feasible and usable.

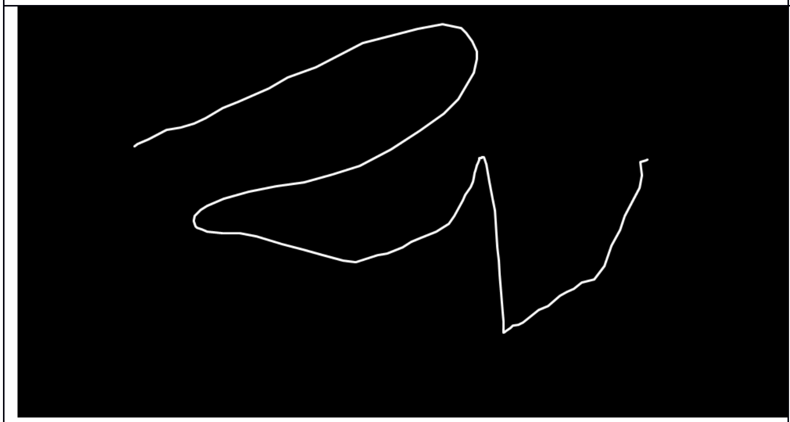
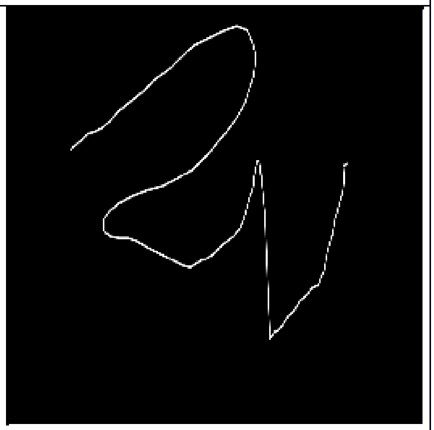
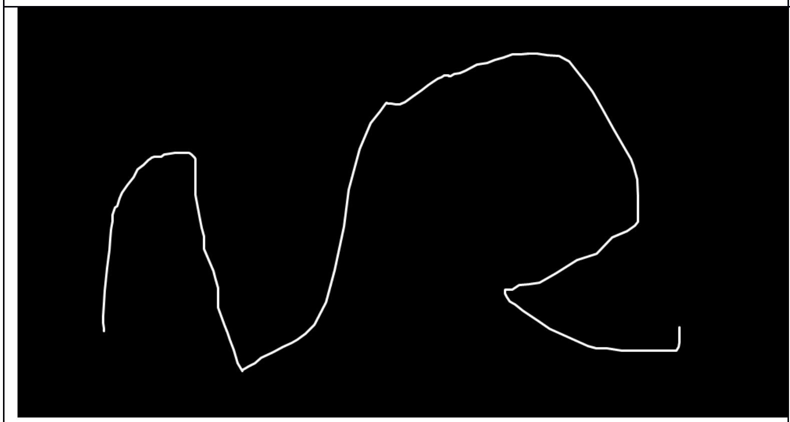
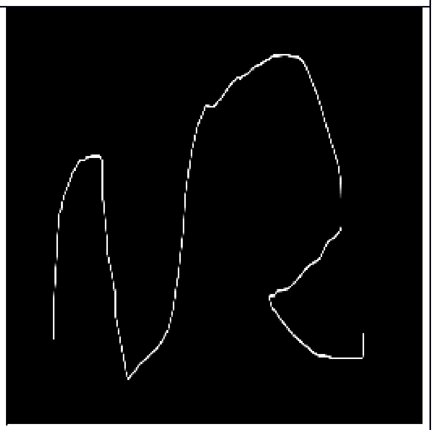
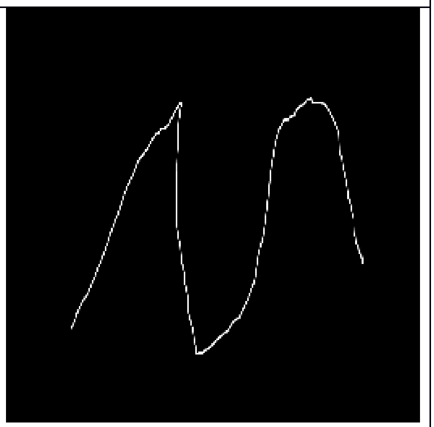
Mouse Track (Raw)		**Mouse Track (256*256)** **scaled**
		
		
		

Table 1 . Captured mouse movement maps and re-scaling

4.2.3 TEST DATA ONLINE

A set of hundred volunteers are asked to use the web interface for the site and divided in subsets of 80 and 20 for training and testing purposes.

Test Data includes:

1) Mouse map for the user
2) Any ratings given by user
3) User ID for reference to both the databases (online and offline)

4.2.4 TEST DATA OFFLINE

For making offline database which contains historical datasets for creating recommendations and making online part feasible, I have used an well renowned dataset freely available and well cited and used in various research journals, i.e., MovieLens datasets .

The MovieLens database is present in following structure:

- User Table : Contains

  user id | age | gender | occupation | zip code

- Movie Table : Contains

  movie id | movie title | release date | video release date| IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi | Thriller | War | Western

- Watch Table

- movie id |user id

❖ User ID: Unique Identification number assigned to user.

- Movie ID: Unique Identification number assigned to Indexed Movies.

4.3 OFFLINE SYSTEM

User data from MovieLens database is used for training a classifier neural net implemented using the discussed back propagation network model, The deployed BPN have the following characteristics –

- Input: movie ratings given to all movies by a user.
- Output: membership of the user in each group.
- No. of inputs: maximum number of movies in the database.
- No. of outputs: no. of groups $\left(log(Totalno. of Movies)\right)$.

Whenever a new rating is given by the user his current and historical ratings are used for updating the structure of the BPN by providing a new feedback, therefore the user is assigned a new class/cluster on the basis of all his ratings. This clustering works as the reference for improving ratings after utilizing the affective/ mouse input from the user.

4.3.1 TLBO

Teacher learner framework of TLBO is used for improving end recommendations by deciding the most important and effective recommendation:

- When a new user enters then system a new mouse movement profile is generated and queried against the existent classification network to ascertain the most similar patterned groups. All this while membership of two groups is assigned to Reference(membership) to each groups users respectively and each user is queried against offline database and movies in the group are assigned a value:

  Recommendation membership: movie-rating * Reff (m)

- Learners are assigned to each group and calculate rating aggregate for all the movies rated by user in that group.
- Teacher receives similarity ratings from all the learners and adjusts recommender membership for all movies in all groups by a normalising similarity factor by the designated learners.
- All movie recommendation memberships are generated and sorted in increasing order. And the sorted list is presented as a recommendation list to the user by the teacher function.
- After a new User rates a movie:

  After a new movie is rated by a user from the recommendations, the event is fed to the offline classifier with all his present rated movies.
- Rating is changed by user for a movie:

  The offline classificatory is fed with the updated preferences.

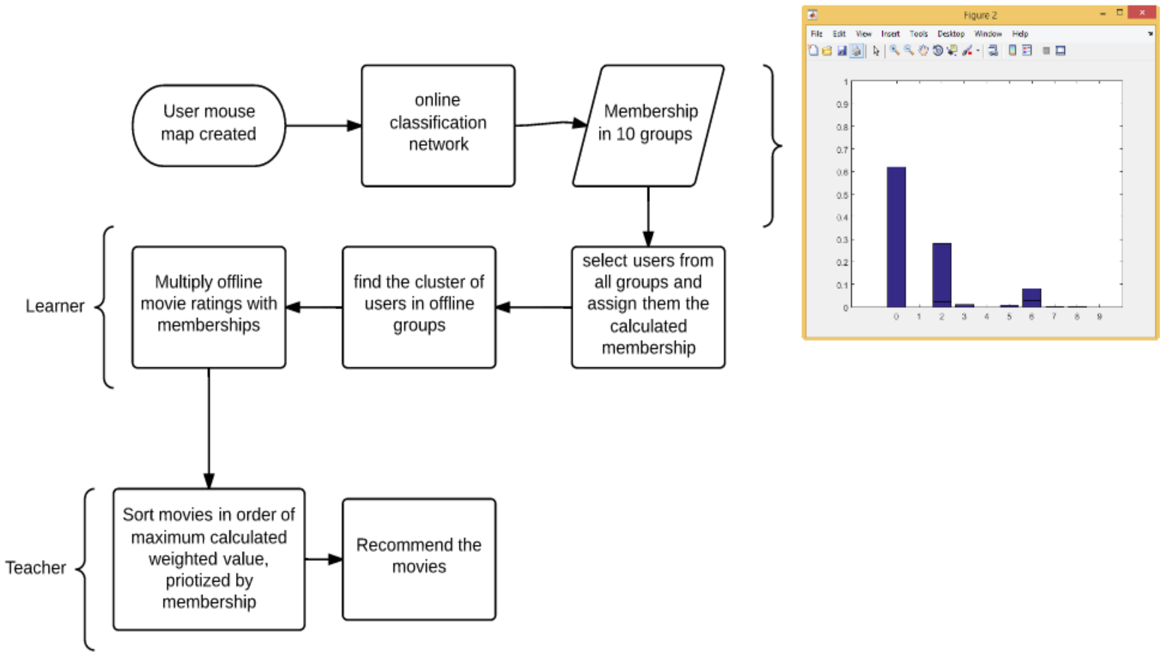


Fig 7 . Offline module with modified TLBO

Chapter – 5

EXPERIMENTAL RESULTS

Finding recommendations for a new user is very since no data or rating is available for making historical references in order to search the vast multinational search space that makes tackling the problem of Cold start, really hard and complex, hence our proposed approach do not require user to bring data to the system rather the basic and affective movements of mouse suffice for providing the data required for making primary recommendations to the User. Since an extra dimension of suggestion parameters is introduced, it enables recommender system to scale from the beginning and mitigate Cold start problem, as depicted in the Fig 9 The exceptionally high recall in the beginning is a proof of the effectiveness of the system, but is also open to further investigation since the no, of users was limited under test conditions and real world situation might vary, but the effectiveness is clear from the comparisons and is attributed primarily to the use of dual neural network infrastructure which allows for learning from previous system scenarios.

In this section, we define the following three most widely used recommendation quality measures: (1) precision, which indicates the proportion of relevant recommended items from the total number of recommended items, (2) recall, which indicates the proportion of relevant recommended items from the number of relevant Items.

As define in [45] precision and recall can b calculated as Let Xu as the set of recommendations to user u, and Zu as the set of n recommendations to user u. We will represent the evaluation precision, recall for recommendations obtained by making n test recommendations to the user u, taking a Θ relevancy threshold. Assuming that all users accept n test recommendations:

$$precision = \frac{1}{\#U}\sum_{u\in U}\frac{\#\{i \in Z_u | r_{u,i} \geqslant \theta\}}{n} \quad (4)$$

$$recall = \frac{1}{\#U}\sum_{u\in U}\frac{\#\{i \in Z_u | r_{u,i} \geqslant \theta\}}{\#\{i \in Z_u | r_{u,i} \geqslant \theta\} + \#\{i \in Z_u^c | r_{u,i} \geqslant \theta\}} \quad (5)$$

Fig 9 and 10 show the performance out proposed system in comparison with the various systems compared in [44] .

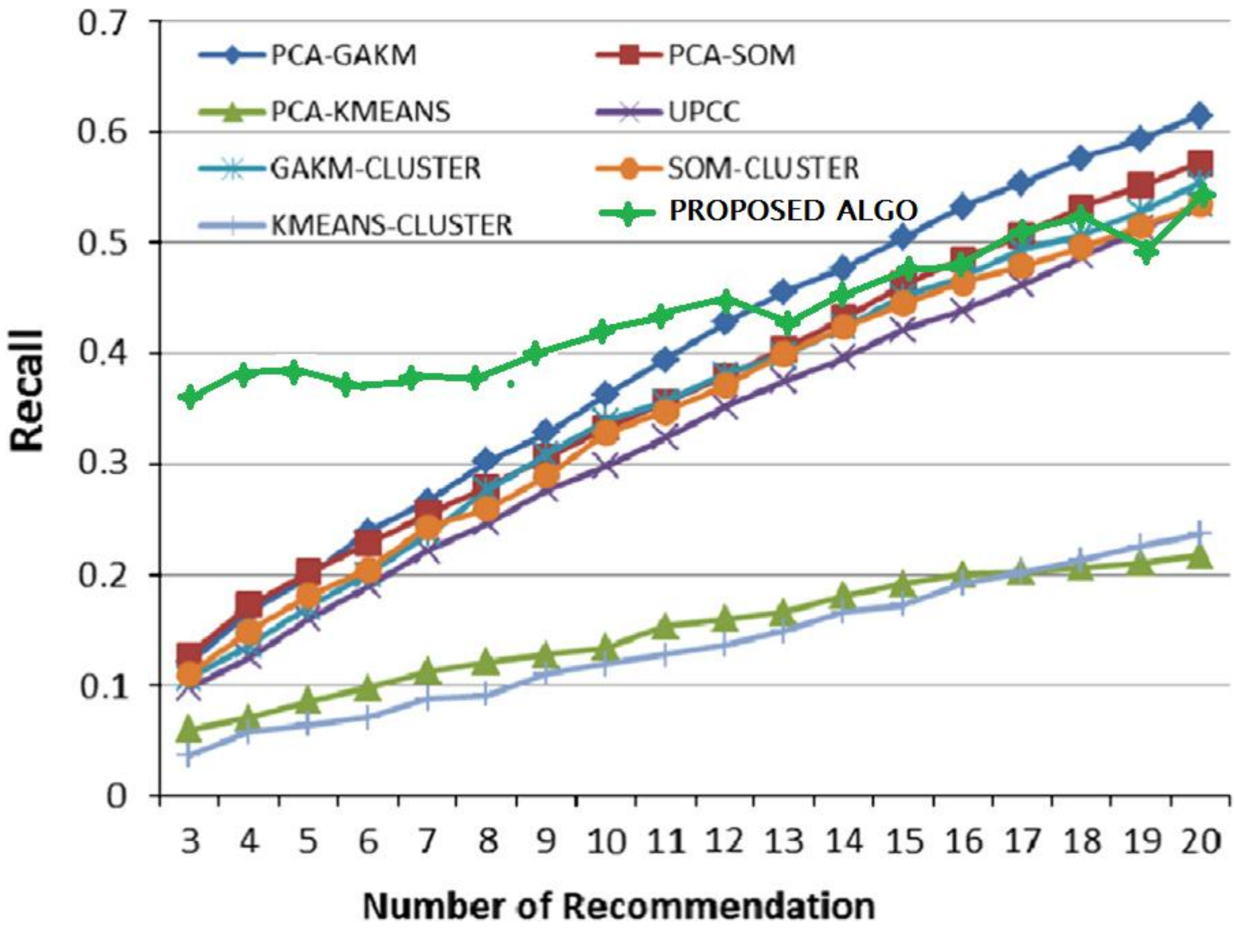


Fig. 9. Recall plot against various systems

The exceptional good recall performance of the system even when the number of recommended movies is very less is attributed to the following facts:

- A neural network infrastructure has the memory property which enables it to probabilistically recall from the past traits and since the network used is a BPN it learns every time a recommendation is made by adjusting according to the choice of the user.
- A double clustering scheme is used in two, offline and online, modes which benefits from both historical and real-time user data enabling the system to mitigate the impacts of cold start and hence, making recommendations with high impact possible, right from the time when our starts using the system.

The slope of the curve produced by our system is comparatively less slant and saturates with time because, the probabilistic nature of the system do not allow for over training of the networks and keeps the system open for new recommendations, making Long tail segregation and Data scarcity in check .

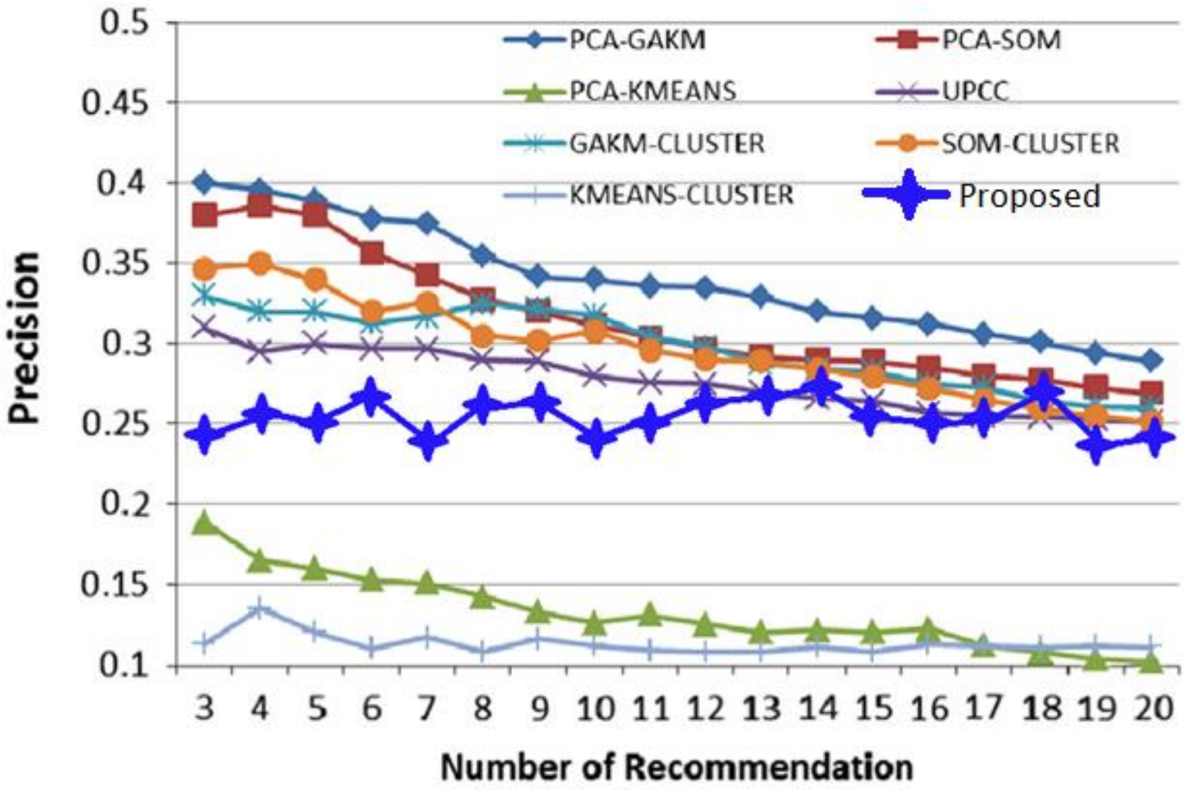


Fig. 10 Precision plot against various systems

The precision stats as shown in fig. 10 of the proposed system approaches an early achieved saturation which enables an overall god user experience irrespective of number of recommendations, since the proposed solution takes into consideration all the data that can be related based on the memberships as suggested, no of recommendations only slightly affect the precision of recommendations .

Other parameters on which the system is compared are calculated as below:

Chapter – 6

CONCLUSION AND FUTURE WORK

The proposed algorithm suggests a method for using mouse inputs from users for enhancing the Recommendations to users and mitigates the problem of Cold start and Long Tail Segregation, Data Scarcity. As shown in the produced results, an improved value of precision in the beginning of recommendation sequence is suggestive of improved cold start and the fact that two layers of clustering are used and are dynamically connected with each other based on the membership values, results in inclusion of all the percipients in the recommendation decisions thereby preventing segregation of results as it happens in the clustering systems across recommendation systems. Data Scarcity is also handled by the TLBO algorithm since all the members are included equally in participation for recommending movies.

Therefore, we conclude that the proposed system seamlessly reduces Cold start impact factor and starts providing users with recommendations as early as a user enters our system and starts receiving recommendations without segregating data in remote cluster tails and sparsely populating data.

The following features can be constructed in future for increasing the efficiency of the whole system:

- Recursive suggestions can be provided by recommending movies based on the selection of movie from among the recommending movies to a given user at a giver frame of time.
- Swarm intelligence can be used for dynamically updating the offline clusters according to the change is membership of user in online groupings.

- Eye tracking can be implemented with mouse tracking using the webcam on users system if available.
- News content Prediction can be done based i=on the mouse movements of the user across the various news published on the news website , since it is expected that the user will focus more on the articles he like and so will be the positions of his eye and mouse which can be tracked for better news predictions .

Chapter – 7

RFERENCES

[1] Balabanovic, M., Shoham Y.: FAB: Content-based, Collaborative Recommendation. Communications
of the Association for Computing Machinery 40(3) (1997) 66-72

[2] Basu, C., Hirsh, H., Cohen W.: Recommendation as Classification: Using Social and Content-Based Information in Recommendation. In: Proceedings of the 15th National Conferenceon Artificial Intelligence, Madison, WI (1998) 714-720

[3] Bell, Robert, Yehuda Koren, and Chris Volinsky. "Modeling relationships at multiple scales to improve accuracy of large recommender systems."Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2007.

[4] Burke, Robin. "Hybrid recommender systems: Survey and experiments." User modeling and user-adapted interaction 12.4 (2002): 331-370.

[5] Trewin, Shari. "Knowledge-based recommender systems." Encyclopedia of Library and Information Science: Volume 69-Supplement 32 (2000): 180.

[6] Rao, Ravipudi V., Vimal J. Savsani, and D. P. Vakharia. "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems." Computer-Aided Design 43.3 (2011): 303-315.

[7] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro. "Content-based recommender systems: State of the art and trends." Recommender systems handbook. Springer US, 2011. 73-105.

[8] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." Knowledge and Data Engineering, IEEE Transactions on 17.6 (2005): 734-749.

[9] Schafer, J. Ben, Joseph Konstan, and John Riedl. "Recommender systems in e-commerce." Proceedings of the 1st ACM conference on Electronic commerce. ACM, 1999.

[10] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. "Collaborative filtering recommender systems." Foundations and Trends in Human-Computer Interaction 4.2 (2011): 81-173.

[11] Schafer, J. Ben, et al. "Collaborative filtering recommender systems." The adaptive web. Springer Berlin Heidelberg, 2007. 291-324.

[12] Barrington, Luke, Reid Oda, and Gert RG Lanckriet. "Smarter than Genius? Human Evaluation of Music Recommender Systems." ISMIR. Vol. 9. 2009.

[13] Ye, Jun. "Cosine similarity measures for intuitionistic fuzzy sets and their applications." Mathematical and Computer Modelling 53.1 (2011): 91-97.

[14] Bäck, Thomas. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford university press, 1996.

[15] Ghazanfar, Mustansar Ali, and Adam Prügel-Bennett. "Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems."Expert Systems with Applications 41.7 (2014): 3261-3275.

[16] Zhang, Zi-Ke, et al. "Solving the cold-start problem in recommender systems with social tags." EPL (Europhysics Letters) 92.2 (2010): 28002.

[17] Schein, Andrew I., et al. "Methods and metrics for cold-start recommendations." Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2002.

[18] Ghazanfar, Mustansar Ali, and Adam Prugel-Bennett. "A scalable, accurate hybrid recommender system." Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on. IEEE, 2010.

[19] Adomavicius, Gediminas, and Alexander Tuzhilin. "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." Knowledge and Data Engineering, IEEE Transactions on 17.6 (2005): 734-749.

[20] Adomavicius, Gediminas, and Alexander Tuzhilin. "Context-aware recommender systems." Recommender systems handbook. Springer US, 2011. 217-253.

[21] Rana, Chhavi, and Sanjay Kumar Jain. "A study of the dynamic features of recommender systems." Artificial Intelligence Review 43.1 (2015): 141-153.

[22] Ge, Mouzhi, Carla Delgado-Battenfeld, and Dietmar Jannach. "Beyond accuracy: evaluating recommender systems by coverage and serendipity."Proceedings of the fourth ACM conference on Recommender systems. ACM, 2010.

[23] Phelan, Owen, Kevin McCarthy, and Barry Smyth. "Using twitter to recommend real-time topical news." Proceedings of the third ACM conference on Recommender systems. ACM, 2009.

[24] McGinty, Lorraine, and Barry Smyth. "On the role of diversity in conversational recommender systems." Case-based reasoning research and development. Springer Berlin Heidelberg, 2003. 276-290.

[25] M. Pantic and A. Vinciarelli, “Implicit human-centered tagging [Social Sciences],” IEEE Signal Process. Mag. vol. 26, no. 6, pp.173–180, Nov. 2009.

[26] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” IEEE Trans. Knowl. Data Eng. vol. 17, no. 6, pp.734–749, Jun. 2005.

[27] M. Pogacnik, J. Tasic, M. Meza, and A. Kosir, “Personal content recommender based on a hierarchical user model for the selection of TV programmes,”User Model.User-Adapted Interact.: J. Personaliz. Res. vol. 15, no. 5, pp. 425–457, 2005.

[28] emotional context in recommender systems,” in Proc. 2007 IEEE 23rd Int. Conf. Data Engineering Workshop, Apr. 2007, pp. 845–852.

[29] Bakhtiyari, Kaveh, Mona Taghavi, and Hafizah Husain. "Hybrid affective computing—keyboard, mouse and touch screen: from review to experiment."Neural Computing and Applications: 1-20.

[30] Schein, Andrew I., et al. "Methods and metrics for cold-start recommendations." Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2002.

[31] Sahebi, Shaghayegh, and William W. Cohen. "Community-based recommendations: a solution to the cold start problem." Workshop on Recommender Systems and the Social Web, RSWEB. 2011.

[32] Sun, Dongting, Zhigang Luo, and Fuhai Zhang. "A novel approach for collaborative filtering to alleviate the new item cold-start problem."Communications and Information Technologies (ISCIT), 2011 11th International Symposium on. IEEE, 2011.

[33] Bobadilla, JesúS, et al. "A collaborative filtering approach to mitigate the new user cold start problem." Knowledge-Based Systems 26 (2012): 225-238.

[34] Shapira, Bracha, Lior Rokach, and Shirley Freilikhman. "Facebook single and cross domain data for recommendation systems." User Modeling and User-Adapted Interaction 23.2-3 (2013): 211-247.

[35] Burke, Robin. "Hybrid web recommender systems." The adaptive web. Springer Berlin Heidelberg, 2007. 377-408.

[36] Leung, Cane Wing-ki, Stephen Chi-fai Chan, and Fu-lai Chung. "An empirical study of a cross-level association rule mining approach to cold-start recommendations." Knowledge-Based Systems 21.7 (2008): 515-529.

[37] Kim, Heung-Nam, et al. "Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation." Electronic Commerce Research and Applications 9.1 (2010): 73-83.

[38] Park, Yoon-Joo, and Alexander Tuzhilin. "The long tail of recommender systems and how to leverage it." Proceedings of the 2008 ACM conference on Recommender systems. ACM, 2008.

[39] Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.

[40] Zadeh, Lotfi A. "What is soft computing?." Soft computing 1.1 (1997): 1-1.

[41] Drew, Philip J., and John RT Monson. "Artificial neural networks." Surgery127.1 (2000): 3-11.

[42] Le Cun, B. Boser, et al. "Handwritten digit recognition with a back-propagation network." Advances in neural information processing systems. 1990.

[43] Rao, Ravipudi V., Vimal J. Savsani, and D. P. Vakharia. "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems." Computer-Aided Design 43.3 (2011): 303-315.

[44] Wang, Zan, et al. "An improved collaborative movie recommendation system using computational intelligence." Journal of Visual Languages & Computing25.6 (2014): 667-675.

[45] Wang, Zan, et al. "An improved collaborative movie recommendation system using computational intelligence." *Journal of Visual Languages & Computing*25.6 (2014): 667-675.