**1.INTRODUCTION**

The recommender system is about to identify the knowledge about the similar user or the event and derive the convenient aspect based on it. It is the criteria of individualized and interesting and useful that separates the recommender system from information retrieval systems or search engines.Recommender systems are an extensively studied and well-established field of research . Recommender systems are the systems which analyses taste and interest of users and recommend services, products, brands or persons as best suited. Users may find it tough to select the best service that meets their individual interest and prerequisite . Most of algorithms and techniques are developed to improve the recommender systems. Recommender systems usually rely on collaborative filtering, content-based filtering, knowledge-based filtering, and hybrid recommendation algorithms. Recommender systems encounter two main challenges for big data application:

1)To make a decision within an acceptable time.

2) To generate ideal recommendations from so many services. In Content-based recommendations, the user will be recommended items similar to the ones preferred in the past. In Collaborative recommendations user will be recommended things that people with similar tastes and preferences liked before. In Hybrid Approaches these methods combine collaborative and content-based methods. Examples of such practical applications include CDs, books, web pages and various other products now use recommender systems .

In many traditional service recommender systems available the ranking and recommendation list provided is the same. Recommender systems have become an important research area. The interest in this area is high because it constitutes a problem- big research area and because of the plenty of practical applications that help users to deal with information. Recommender systems have their relevance to information retrieval in different areas.It analyze the taste, the mood or the context in which the user is at the moment. Based on the analysis, they create an accurate recommendation that suits the particular user . There are various techniques used to create recommendations. The two main categories of recommendation systems are content based and collaborative.

In both categories we should go through the entire entity base to find the correct item to be recommended. In collaborative approach this represents matrix of users and items, in

content-based it is matrix of items and their similarity. Nevertheless, there is no guarantee that the estimation is correct and the recommended item is accurate enough for the user.

Many recommendation systems try to recommend item by pairing the extracted knowledge base with the user’s context and taste. As a result, the recommender systems may suffer

from performance issues what makes them unusable in real time.

In this article we aim to design and evaluate a recommendation method that uses a new approach for recommending items with various attributes such as movies.

Instead of the approaches mentioned above we decided to design a method which makes use of graph structure. We experimented with graph algorithms to provide comparison

and their pros and cons. Graph-based recommender systems have been tested in the past and have shown promising results . Our contribution is in comparison of graph algorithms and their modifications. We applied our method for multimedia recommendation (e.g. movies, TV shows and TV programs) in web application called Televido, which enabled us to experiment with various algorithms and real users.

Recommender systems became very popular due to information overload which we have to face every day. There are approaches such as collaborative filtering or content based. Both approaches could be boosted by using graph representation to speed up the process of finding the most appropriate items. In case of collaborative filtering we naturally need homogeneous

bipartite graph representation since we are using single type connections among users and items.

Mele et al. proposed interesting solution using graph representation of direct connections among users. These connections express a behavior of these users. Connection among user *u*

and *v* is added when user *u* discovered some item earlier thanuser *v*. This temporal modification of collaborative filteringis similar to well-known model of PageRank .

In case of content-based approaches we commonly use text optimized databases to quickly compute the similarity among items . However, a graph could be one way to hold information on items. Especially if we use known metadata (categories, keywords, attributes) as the content to calculate similarity.

* 1. **MOTIVATION**

Recommendation systems help users find and select items (e.g., books, movies, restaurants) from the huge number available on the web or in other electronic information sources . Given a large set of items and a description of the user's needs, they present to the user a small set of the items that are well suited to the description. Recent work in recommendation systems includes intelligent aides for filtering and choosing web sites ,news stories , TV listings ,and other information. The users of such systems often have diverse, conflicting needs. Differences in personal preferences, social and educational backgrounds, and private or professional interests are pervasive. The ability of computers to converse with users in natural language would arguably increase their usefulness and flexibility even further. Research in practical dialogue systems, while still in its infancy, has matured tremendously in recent years . Today's dialogue systems typically focus on helping users complete a specific task, such as planning, information search, event management, or diagnosis. In this paper, we describe a personalized conversational recommendation system designed to help users choose an item from a large set all of the same basic type. Our goal is to support conversations that become more efficient for individual users over time. To our knowledge, this is the first personalized spoken dialogue system for recommendation, and one of the only conversational natural language interfaces that includes a personalized, long-term user model. Second, it introduces a novel model for acquiring, utilizing, and representing user models. Third, it is used to demonstrate a reduction in the number of system-user interactions and the conversation time needed to find a satisfactory item. The combination of dialogue systems with personalized recommendation addresses weaknesses of both approaches. Most dialogue systems react similarly for each user interacting with them, and do not store information gained in one conversation for use in the future. Thus, interactions tend to be tedious and repetitive. By adding a personalized, long-term user model, the quality of these interactions can improve drastically. At the same time, collecting user preferences in recommendation systems often requires form filling or other explicit statements of preferences on the user's part, which can be difficult and time consuming.

* Many of the top commerce sites use recommender systems to improve sales.
* Users may find new books, music, or movies that was previously unknown to them.
* Also can find the opposite for e.g.: movies or music that will definitely not be enjoyed.

**2.LITERATURE SURVEY**

**2.1 OUTCOME OF LITERATURE SURVEY**

In the domain of media content recommendation research usually faces the problem calculating the similarity among items (in content-based). Authors often propose modifications of collaborative filtering. Biancalana et al. used neural networks to train the model once. This actually reduced the multidimensional nature of their data but disabled

them to make on-the-fly and up-to-date recommendations without retraining the model.

Another interesting work on using graph algorithm is presented by Chen et al.. They used tripartite graph of users, items and queries used in the video search. This algorithm is very similar to spreading activation which could be promising in case of graph-based recommending. Similar approach was also presented by Bogers et al. . In their work they used more contextual features (genre, language, director, actor, etc.). They used probabilistic algorithm which uses the matrix holding the intensity of transitions among entities while user browses them .

In [4] this paper addresses the difficulty of retrieving relevant, pertinent, and novel information for a large system that involves fusion of data in different formats such as, text, barcode, and pictures. We come up with a schema to combine an intelligent image retrieval and intelligent information retrieval (IIR) along with the user profile learning to develop a recommender system. In [1] Sherla1 It aims at presenting a customized service recommendation list and recommending the foremost applicable services to the users effectively. Specifically, keywords area unit won’t indicate users’ preferences, and a user-based cooperative Filtering algorithm is adopted to get applicable recommendations. In [3] here summarizes the various aspects of RS, problems /challenges. It also discusses certain issues specific to context aware systems and the long tail problem of RS.

In [4] authors explore the different characteristics and potentials of different prediction techniques in recommendation systems in order to serve as a compass for research and practice in the field of recommendation systems. In [4] the system can recommend interesting document files to users by collaborative filtering. In the system, we employ regional similarity between users and general similarity between groups. Using this system, users can find out necessary knowledge and reuse knowledge effectively. In [5] author proposes text indexing system based on the assignment of an appropriate weighted single term which produces superior retrieval result as evidence accumulated over the past 20 years. This result depends crucially on the choice of effective term weighted system. In [6] author proposes weighted least square method is utilized to obtain the weights. This method has the advantage that it involves the solution of a set of simultaneous linear.

**2.2 PROBLEM STATEMENT**

Recommendation of movies based on graph traversal algorithm.

**2.3 OBJECTIVES**

* To provide movie recommendation to users.
* To propose an approach to recommendation with a focus on the natural change of user’s interests in movies.
* To make use of a graph representation and experimented with modified graph algorithms.
* To design a representation of the data about movies in a graph structure and a method which uses our data model for recommendation.
* To propose four recommendation algorithms which are capable to find recommendations based on initial nodes, which selection is based on the user’s current interests.
* To have a system that is user friendly and easy to understand and use .
* To create a dataset that has all relevant informations about a particular movie.
* To make our system diversifiable so that it can satisfy users of different geographical location.
* To give ways to different attributes.

**3. METHODOLOGY**

Our recommendation algorithms are graph algorithms, which traverse the graph structure from initial nodes. The initial nodes represent user’s interest. An initial node might

be a movie or a genre the user likes. These initial nodes are either selected explicitly by the user as a query or implicitly based on the user model and the feedback from the user

using standard methods. We actually do not need user model to be connected to nodes. We only need the list of items which are relevant for user and use it as a query on the fly.

The algorithms try to find the nodes which are the closest in the graph to all initial nodes, which are then returned as recommendations. There are multiple ways of looking

at the problem of finding the closest nodes, especially in a very complex graph, which is why we designed and implemented four separate algorithms to compare.

*A. Union Colors Algorithm*

The Union Colors algorithm is based on the basic

*Breadth-first search (BFS)* graph algorithm. The algorithm

works in the following way:

1) Mark each initial node with a different color (naturally,

colors are represented by numbers).

2) Perform a simultaneous BFS from each initial node:

a) Enqueue all initial nodes.

b) Dequeue a node and visit it.

c) Add all neighbors of the visited which are yet to be visited node into the queue.

d) Repeat from step b)

3) When a node is visited:

a) If it is yet to be colored, color it with the color of the initial node

b) If it is already colored (with a different color),merge the two colors into one - remember that

one color equals the other

4) Keep merging colors until the last two colors are merged and only one color remains

5) Return the current and the next required number of nodes in the queue as the result.

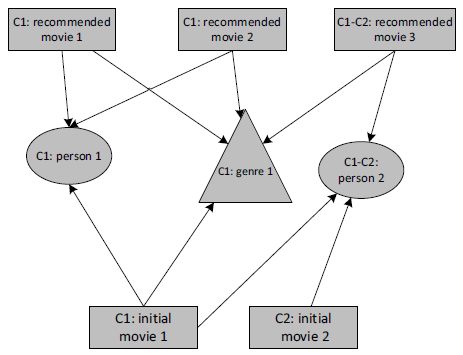


Figure 1: Visualisation of union colors algorithm

*B. Mixing Colors Algorithm*

1) Mark each initial node as visited by a different color.

2) Perform a simultaneous BFS from each initial node:

a) Enqueue all initial nodes.

b) Dequeue a node and visit it.

c) When visiting a node, mark it with the colors of all the nodes which enqueued it.

d) Try to add all neighbors of the visited node into the queue.

i) If the node is not in the queue, add it and remember, that it was added by the visited node.

ii) If the node is already in the queue, do not add it but remember that it has also been added

by the visited node.

e) Repeat from step b)

3) Continue until the required number of nodes is colored by every color, return them.

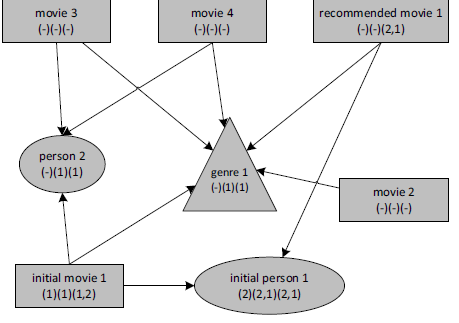


Figure 2: Visualisation of mixing colors algorithm

*C. Energy Spreading Algorithm*

1) Set the energy of each initial node to some constant value.

2) Perform a simultaneous BFS from each initial node

3) When a node is visited its energy increases by value *E*, *E* = *Ep* *n* , where *Ep* is the energy of the parent node which enqueued the visited node and *n* is the number of nodes the parent node enqueued.

4) A node’s energy can increase multiple times, but it only spreads it when it receives energy for the first time.

5) Continue until the required number of nodes is visited from each initial node.

6) Order the nodes by their energies and return the required number of the nodes, the more energy it has the higher it is.

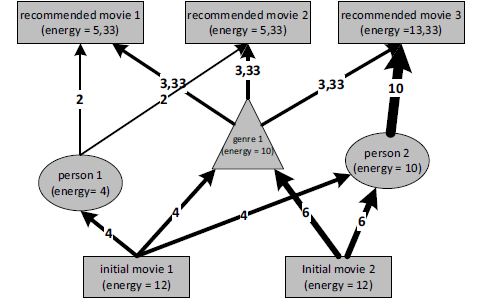


Figure 3:Visualisation of energy spreading algorithm

*D. Modified Dijkstra’s Algorithm*

1) Run the Dijkstra’s shortest path algorithm from each node to (some constant) maximal depth – calculate the shortest path from the starting node to each visited node.

a) If the algorithm is ran from the first initial node, put all visited nodes with the value of the shortest path as their total value into a results set.

b) If the algorithm is ran from other initial nodes, check if the node is in the results set. If yes, add the shortest path value to its total value.

2) Check if each node in the results set was visited from all initial nodes.

3) Order the nodes in the results set by their total values of the shortest paths.

4) Return the required number of nodes from the ordered results set.

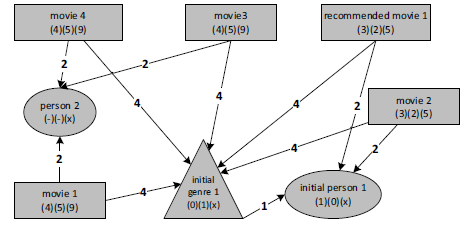


Figure 4:Visualisation of modified dijikstra’s algorithm

The goal of the experimental evaluation we performed was to determine the best algorithm of the four algorithms we designed and to determine the accuracy of each algorithm.Based on the results of the experiment we intend to work on the most successful algorithms in the future and further modify them to make their results even better. We filled the databases with a dataset consisting of information about a large amount of real movies and TV programs.

In the end, the graph database consisted of roughly 165 000 nodes and 870 000 relations.

The experiment used to test our methods based on collecting explicit feedback from users. The user’s task was to pick some initial nodes – movies, people or genres.The system generated 4 different sets of recommendations based on the initial nodes using the four recommendation algorithms designed.

**4.WORK DONE**

In this project,we analysed the performance and accuracy of all four algorithms given in the paper such as – union colors ,mixing colors,energy spreading and modified dijikstra’s algorithms. We tried to implement the algorithms with our basic knowledge of graph traversal algorithms and found out that there is still chances of modification.Hence we again propose a new method which can be defined under weighted hybrid recommendation technique.

In this technique, we are combining a graph traversal algorithm with maximum heap data structure and trying to sort the movies according to their ascending order of ratings given by various users.

The proposed algorithm is as follows:

**Input:** a number of movies: m

**Output:** a number of clusters: K

**Step 1** Select n movies from m movies n<m

**Step 2** If n>20 then select top 20 movies from n movies based on ratings.

Else display the output movies sorted by rating.

**Step 3** If rating of movies x, y are equal i.e. If

Rx= Ry

Then select those movies which have greater number of user votes.

**Step 4** Assume K=4.

**Step 5** REPEAT (6, 7)

**Step 6** Chose initial centroid C1, C2, C3, C4.

**Step 7** UNTILL centroid does not change

**5. RESULTS AND ANALYSIS**

The results show that the users usually picked the Union Colors algorithm as the first. This results might be biased- because of the way the experiment user interface was designed the first algorithm was always the same. Since the rating of the algorithm was the lowest, we assume the users left the first algorithm picked when they weren’t very happy with the recommendation or were simply lazy.

Next in line are the Mixing Colors and the Energy Spreading algorithms. The first had a slightly better position but the second had a better rating. These algorithms appear to be

the best of all, although they require some improvement.

The Modified Dijkstra’s rating is not bad, but it was not picked as good nearly as often as other algorithms. The experiment we performed also had an auxiliary goal which was to find out information about the performance times of the algorithms. The only algorithm which had

performance issues was the Modified Dijkstra’s algorithm, which needs to keep a priority queue in order to work. The other algorithms’ performance was satisfactory considering

the scale of the data model graph, the recommendation was usually done in less than a second.

**6. CONCLUSION AND FUTURE WORK**

We designed a graph-based data model and proposed four recommendation

algorithms. The evaluation shows promising results, but it also shows the need to further research. Observing four different algorithms helped to achieve an improvement especially

in the precision of recommendation. The experiment showed that we can disregard the Modified Dijkstra’s algorithm and probably also the Union Colors algorithm and focus on improving the rest. The data model can also be improved by instructing new types of relationships, for example keywords.

The main advantage of the graph representation for the task of item recommendation is performance. The implemented methods seem to be fast enough to work in real time. Our proposal is relevant for recommendation of huge number items and variable interests of users. Our approach is applicable also in other domains.

Proposed traversal algorithms could be used in domains where we need to represent entities with variety of attributes and relations. For instance, music recommendation, news recommendation, book recommendation or even recommendation in e-shops.

**REFERENCES**

[1] Sitalakshmi Venkatraman, Senior Member, IACSIT and Sadhana J. Kamatkar Intelligent Information Retrieval and Recommender System Framework International Journal of Future Computer and Communication, Vol. 2, No. 2, April 2013.

[2] Dept. of CA, Vasavi College of Engineering, Hyderabad-31, India Dr.T.Adi Lakshmi, Dept. Of CSE, Vasavi College of Engineering, Hyderabad-31, India Recommendation Systems: Issues and challenges Soanpet .Sree Lakshmi et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5771-5772 Soanpet .Sree Lakshmi.

[3] F.O. Isinkaye a,\*, Y.O. Folajimi b, B.A. Ojokoh c Recommendation systems: Principles, methods and evaluation.

[4] Profiles Akihito NAKAGAWA and Takayuki IT0 An Implementation of a Knowledge Recommendation System based on Similarity among Users

[5] G. Salton, Automatic Text Processing. Addison-Wesley, 1989.

[6] A. Chu, R. Kalaba, and K. Spingarn, A Comparison of two Methods for Determining the Weights of Belonging to Fuzzy Sets, Optimization Theory and Applications, Springer vol. 27, no. 4, pp. 531-538,1979.