Enhancing Movie Recommender System

By

Patel Ronakkumar N.

11MCEC13

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2013

Enhancing Movie Recommender System

By

Patel Ronakkumar N.

11MCEC13

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2013

Enhancing Movie Recommender System

Major Project(Part-II)

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology in Computer Science and Engineering

By

Patel Ronakkumar N.

(11MCEC13)

Guided By

Prof. Priyank B. Thakkar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2013

Certificate

This is to certify that the Major Project entitled "**"Enhancing Movie Recommender System"** submitted by Patel Ronakkumar N. (11MCEC13), towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Priyank B. Thakkar
Guide and Assistant Professor,
Department of C.S.E.,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Vijay Ukani
Associate Professor and PG-Coordinator(C.S.E.),
Department of C.S.E.,
Institute of Technology,
Nirma University, Ahmedabad.

Dr Sanjay Garg
Professor and Head,
Department of C.S.E.,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Professor and Director,
Institute of Technology,
Nirma University, Ahmedabad.

Undertaking for Originality of the Work

I, Patel Ronakkumar N., Roll. No. 11MCEC13, give undertaking that the Major Project entitled **"Enhancing Movie Recommender System"** submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in Computer Science and Engineering of Nirma University, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:18^{th} May 2013
Place:Ahmedabad

Endorsed By:

Prof.Priyank Thakkar

Abstract

Recommender Systems are used by customers to buy items more efficiently. Business is also benefited simultaneously. There are various approaches of Recommender Systems, like: (1) Collaborative Filtering (2) Content based Filtering and (3) Hybrid Filtering. In Collaborative Recommender System, ratings of the most similar users(in user based collaborative filtering) or items(in item based collaborative filtering) are used to predict the rating of a new item. In Content Based Filtering, user profile is constructed based on the content of the items liked by the user in the past and then based on similarity between user and item profile, prediction is made. Hybrid Filtering combines collaborative and content based approach. In this dissertation, we focus on movie recommendation task. We propose a new hybrid approach which combines usage, tag and other content data of items. We model prediction task as classification problem where our aim is to predict whether the item will be liked or disliked by the user. In our work, we propose item based Recommender which combines usage, tag and movie specific data such as genres, star cast and directors to improve the accuracy of the Recommender System. We have tested our approach using Hetrec2011-movielens-2k dataset. We have used Accuracy, Precision, Recall and Fmeasure to evaluate the performance of our proposed algorithm.

Acknowledgements

My deepest thanks to **Prof. Priyank B. Thakkar**, Assistant Professor and the Guide of the project that I undertook, for giving valuable inputs and correcting various documents of mine with attention and care. He has taken the pain to go through the project and make necessary amendments as and when needed.

My deep sense of gratitude to **Prof. Vijay Ukani**, Associate Professor and PG-Coordinator(C.S.E.), and **Dr Sanjay Garg**, Professor and Head of C.S.E. Department, for an exceptional support and continual encouragement throughout the Major project.

I would like to thank **Dr K Kotecha,** Professor and Director, for his unmentionable support, providing basic infrastructure and healthy research environment.

I would also thank my Institute and all faculty members in Department of Computer Science & Engineering.

Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom I am able to complete my dissertation work successfully.

- Patel Ronakkumar N.

11MCEC13

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 General

In these days, as the e-commerce industry is growing and becoming complex, the information about the products is increasing with exponential rate. In such environment, customers have difficulty to find optimal information about products from the tremendous amount of information. To help the buyers, the major e-business companies are also developing their Recommender systems (RS) to help their customers to choose products/items more efficiently. The customers can get benefit by receiving some useful information about the products which they are likely to buy and at the same time, the business can get benefit with an increase of its sales.

Recommender systems emerged in the mid-90s in order to filter out irrelevant information and select content that meets user needs. Recommender system has been described as "An information filtering technology, that produces individualized recommendations as output or have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options"[2]. These systems can be used for different purposes in several domains from offering products to consumer in e-commerce, to finding proper information in research etc.. Averaging the opinions of a large group of people captures the really more useful information than legitimate experts.

An e-commerce Recommender system will gather current customers information and past purchases and uses these information to recommend new products. With the use of some standard recommender algorithms, an e-commerce system provides customers with accurate recommendations. There are recommendation systems in different domains

such as films, television programs, video, music, books, news, images, web pages[20]. Many advanced systems try to help users to find right data according to their interest. Personalized recommendations are a key method for information retrieval. System like Amazon (www.amazon.com) help customers to find the best products online. Pandora (www.pandora.com) offers a wealth of information and services related to many aspects of music. It also helps users to find songs that they might be interested in by capturing the user behavior. MovieLens (www.movielens.org) and IMDB (www.imdb.com) tries to guide users to identify the movies they might like. YouTube(www.youtube.com) is known as the biggest collection of online videos and a system that can leverage the user browsing history for recommendation.

E-commerce websites utilize one or more Recommender systems to suggests the best solution to their customers. Various movie businesses like Netfilx, IMDB, Hulu etc. recommend the movies by creating a relationship with the customers. Customers retention is very important to such websites, this relationship will in benefit of the customer as well as the websites. Although there are several factors which affect the quality of Recommender system, recommendations based on common viewpoints of user become more and more trustworthy and widely-used. Movie recommendation is an open research area with unanswered problems and with growing social networking data. There is a need of solutions to those problems to keep the customers en-tact[18].

Generally, recommendation systems can be categorized as content-based, collaborative or hybrid [20], as described below:

1.1.1 Content-Based Recommender (CBR)

Systems recommend items similar to that one the user has preferred in the past. However, there are some limitations to the CB technique, like the data scarcity problem. Modeling the user's interest is limited to extracting features from their browsing or purchasing history. Another limitation is that CB systems cannot identify new and different items that the user may enjoy, as it is prone to finding only those that are highly similar to the items in the history of that user[2].

1.1.2 Collaborative Filtering (CF)

It will collect the preferences from multiple users. Collaborative recommendations are based on the similarity. The system will recommend those items that are liked by most

similar users. Similarly to the CB approach, CF techniques have shortcomings as well: a new item cannot be recommended until someone references it. Moreover, a user with unusual preferences may not receive recommendations unless there are other users that exhibit the same interests[2].

1.1.3 Hybrid Recommender System

It is combination of the CBR and CF techniques, referred as the hybrid approach to overcome some of the problems that each approach has individually, and can achieve good results. Hybrid Recommender system could offer good performance even with little or no user data[2].

1.2 Motivation

How to find relevant information from large available resources is the main issue now a days. Significant research efforts have been made to automated filtering systems that provide humans with desirable and relevant information only. Search engines count among these filtering systems and have gained wide-spread acceptance. During the last 10 years, Recommender systems have been gaining more popularity as another efficient means of reducing complexity when searching for relevant information. Recommenders intend to provide people with suggestions of products they will appreciate, based upon their past preferences, history of purchase, or demographic information. Significant research has been made in Recommender system. Movie Recommender is also an emerging area and still requires an improvement. How to use efficiently movie information and how to model it in classification are some of the issues.

1.3 Scope of Project

By combing usage, tag and content data of movies, we can improve the quality of recommendations. Movie recommendation task can be modeled as classification problem where our aim is to predict whether the movie will be liked or disliked by the user. We also exploits movie specific data such as movie genres, star cast and directors.

Chapter 2

Literature Review

E-commerce websites help customers to find interesting items from huge data available over the Internet. These websites use Recommender systems to find relevant items from the large number of choices. In the past decade, lot of work has been done on Recommender systems and various techniques have been developed. This chapter aims to discuss general concepts and terminology on recommendation techniques. We discuss, in more detail, major types of recommendation techniques and algorithms, pointing out their advantages and disadvantages.

2.1 Recommender Systems

It is estimated that by the end of year 2012, one third of the world population will be using the Internet and about 1.2 billion people will join social networking websites. People spend 6.7 billion hours on social networking websites in a month, constantly pouring the information on the web. As a result, we are drowning in information, but continue to starve for knowledge[17]. Thus, it is becoming difficult for people to find the resources of their interest. Many techniques and systems have been developed in the last decade to help users to find the right information. One of the most successful technologies is known as Recommender Systems, which are based on personalized information filtering, and predict relevant items for a particular user. Recommender systems generate item sets, that may be liked by specific users.

Given the enormous amount of data and diverse users, Recommender systems can not simply rely on users or community for recommendations. We can use different techniques from different areas such as Natural Language Processing, Artificial Intelligence, Human-

Computer Interaction, and Information Retrieval are incorporated in advanced systems for better predictions [20]. During the past decade significant progress has been made in the field of Recommender systems, though this is still a very active and popular area, as there are many open ended problems that need to be addressed. To choose the right recommendation algorithm for a problem, offline experiments need to be conducted over the same set of real data[2]. Typically, Recommender systems are broadly classified in three categories:[20]

1. Content-based recommendations: Items recommended to the users, similar to the ones the user preferred in the past.

2. Collaborative recommendations: Items recommended to the users, by opinion from the people with similar tastes and preferences.

3. Hybrid approaches: Hybrid approaches combine collaborative and content-based techniques.

2.2 Content Based

Content-Based Recommender (CBR) systems suggest items from huge available options based on similarity between item features and user's[22]. Typically, these Recommender systems suggest items similar to the items preferred in the past by the user. In content-based systems, the items are represented by their associated features. The algorithm compares the collective user information against the feature content of a new item, and items with high similarity score are recommended. CBR approaches construct a user profile from the features used to describe the items that the user has rated. CBR models are evolving models which keep updating the user profile based on user item preference and activity. Sometimes, CBR approaches rely on users feedback to learn their (users) preference. Typically, information about user choices or interaction with the recommendation system, can be used to construct the user profile. User activity and user queries may help Recommender systems to filter out the items that are already viewed by the user. The information needed can be captured in two ways, through implicit feedback or through explicit feedback.

Explicit feedback: On various websites, users are asked to provide general information about themselves this type of information collected from users directly is considered to be explicit feedback. Such information includes: age, gender, location, interests, etc.. Another option that most websites opt for is to ask feedback on a product. Users are suppose to provide feedback either in the form of surveys or general forms. This technique is bothering and most of the time users are not interested in filling out the forms, thus the resulting profiles are imprecise. Hence, nowadays e-commerce websites are requesting users to express their opinions by selecting a value in a range of explicit ratings. This is usually less troublesome to users than filling out the forms.

Implicit feedback: Monitoring the user activity over the web can provide implicit feedback. Usually, users are not aware that they are providing feedback to the system. One such example is YouTube, where direct relation between the amount of time spent on a single video per session and user interest is captured. This type of feedback may not be as accurate as the feedback in the form of explicit ratings by users, but users are not disturbed as discussed earlier. CBR systems work best when recommending mostly text-based items, where the content is extracted in form of keywords.
One such popular content-based technique is the Fab system [24], which is designed to recommend the web pages to the users. In the Fab system about 100 most important keywords are chosen to represent the web pages in the corpus. Similarly, documents are represented by 128 most distinct and frequent words in and the importance of each word in a document is measured by some weighting scheme. There are many weighting schemes to calculate the importance of a keyword, but the term frequency/inverse document frequency (TF-IDF) measure shown in [20] is one of the most popular and known measures for specifying keyword weights. TF correlates to the term's frequency, defined as the number of times term t appears in the currently scored document d($TF_{t,d}$). Documents that have more occurrences of a given term receive a higher score. Normalized term frequency is introduced to capture more accurately the importance of a term t in document d. Normalized TF is a ratio between the frequency of a terms occurring in a document and the maximum term frequency in that document (tf -max(d)). Thus,

$\text{nTF}_{t,d}$ is normalized term frequency of a term t in a document d and is given by:

$$nTF_{t,d} = \frac{TF_{t,d}}{tf - max(d)} \tag{2.1}$$

The inverse document frequency for a term t is given as:

$$IDF_t = log\frac{N}{n_t} \tag{2.2}$$

where N is the total number of documents in the corpus and n_t is the number of documents in which term t appears. TF-IDF weight for term t in document d is defined as:

$$w_{t,d} = nTF_{t,d} \times IDF_t \tag{2.3}$$

To capture the TF-IDF in the content based component of our approach, we have used the modified version of TF-IDF which moves one step further and eliminates the noise and synonymy problems.

CBR techniques have been successfully used in many areas, including in the field of biological and medical sciences. PURE is an article recommendation system, where a user has to feed initial preferred articles into the system, which then iteratively captures the user preferences from their inputs[20].

Advantages of content-based Recommender systems:

1. Implicit feedback from users is enough to construct the user profile and with increase of database content over the time, performance of CBR gradually improves.

2. CBR can help to recommend new or unpopular items to users based on their taste. Hence, new items do not starve for users explicit feedback.

Disadvantages of content-based Recommender systems:

1. CBR systems are limited to features that are explicitly associated with items. For better performance of the system, a sufficient set of features is required, so either automatic feature extraction from content is needed or manually annotation of items is required. Applications of feature extraction methods are somewhat limited to text based approaches and harder to apply in domains such as image or video rec-

ommendation. Also, it is impractical to identify features manually due to limitation of resources

2. Another problem with CBR is that when items are represented by the same set of features, they can be indistinguishable.

3. CBR systems suffer from an insufficient number of ratings at very early steps.

4. Over-specialization can occur when the CBR system only recommends items scoring highly against a user's profile. In such cases, the user is restricted to seeing items similar to those already rated. Often this is addressed by injecting some randomness in the predictions.

5. New users may not be able to get accurate predictions according to their taste. For better predictions, users need to have a sufficient number of ratings, which is not always possible.

Techniques used:[18]

1. Memory Based:

   - TF-IDF
   - clustering

2. Model Based:

   - Bayesian classifiers
   - clustering
   - Decision trees
   - Artificial Neural Network

2.3 Collaborative Filtering

Collaborative filtering (CF) is a technique for producing personalized recommendations by computing the similarity between the current user and other users with similar choices. Thus, the current user choice is predicted by gathering choice information from other users with similar preferences. If choices matched in the past, it is assumed that they will

match in future as well. Prediction and recommendation are the two main parts of a CF. Collaborative prediction uses the current preferences that are available and the other users preferences relation to predict the current user preferences. Developing a set of items which are more likely to be of interest to the current user is known as collaborative recommendation.

2.3.1 Types of Collaborative Filtering Algorithms

- Memory Based collaborative filtering algorithms

  Memory-based algorithms use the entire item-user dataset. A set of similar users are identified for the current user, and rating predictions are generated based on ratings in the neighborhood of the current user.

  Techniques used:[18]

  - Nearest Neighbour(cosine, correlation)
  - Clustering
  - Graph Theory

- Model Based collaborative filtering algorithms

  Model-based CF algorithms use the pure rating dataset to construct a model to make prediction. Well known model-based techniques include Bayesian models which use the naive Bayes strategy to make predictions. Clustering CF chunk the big dataset in a set of clusters, then recommendations are made independently for each cluster to achieve better scalability. Model-based CF methods can deal with the sparsity, scalability and other problems in a better way than memory based methods. The prediction performance is improved and an intuitive rationale for recommendations is given. Irrespective of several advantages, this technique also has some shortcomings. Model building is usually expensive, furthermore model-based CF performs a trade-off between scalability and performance prediction. At last, useful information can be lost when dimensionality reduction techniques are employed [2].

  Techniques used:[18]

  - Bayesian Network
  - Clustering

- Artificial Neural Network
- Linear Regression
- Probabilistic Model

2.3.2 Similarity Measure [20]

Similarity computation is the backbone of collaborative filtering as neighborhood formation is done based on these values. To evaluate the similarity between any two users or items, Various approaches have been used to compute the similarity sim(c,$c^{'}$) between users in collaborative Recommender systems. In most of these approaches, the similarity between two users is based on their ratings of items that both users have rated. The two most popular approaches are correlation and cosine-based[20]. To present them, let S_{xy} be the set of all items rated by both users x and y, i.e., $S_{xy} = \{s \in S | r_{x,s} \neq \emptyset \ \& \ r_{y,s} \neq \emptyset\}$. In collaborative recommender systems, S_{xy} is used mainly as an intermediate result for calculating the nearest neighbors of user x and is often computed in a straightforward manner, i.e., by computing the intersection of sets S_x and S_y. However, some methods, such as the graph-theoretic approach to collaborative filtering [17], can determine the nearest neighbors of x without computing S_{xy} for all users y . In the correlation-based approach, the Pearson correlation coefficient is used to measure the similarity.

$$sim(x,y) = \frac{\sum_{\forall_s \in S_{x,y}} (r_{x,s} - \bar{r_x})(r_{y,s} - \bar{r_y})}{\sqrt{\sum_{\forall_s \in S_{x,y}} (r_{x,s} - \bar{r_x})^2} \sqrt{\sum_{\forall_s \in S_{x,y}} (r_{y,s} - \bar{r_y})^2}} \tag{2.4}$$

where $S_{x,y} = \{s \in S | r_{x,s} \neq \varnothing, r_{y,s} \neq \varnothing\}$

Similarity between two item is described

$$sim(i,j) = \frac{\sum_{\forall_u \in T} (r_{u,i} - \bar{r_i})(r_{u,j} - \bar{r_j})}{\sqrt{\sum_{\forall_u \in T} (r_{u,i} - \bar{r_i})^2} \sqrt{\sum_{\forall_u \in U} (r_{u,j} - \bar{r_j})^2}} \tag{2.5}$$

where T is a set of users who has rated both the item i and j.

2.3.3 Prediction Computation [20]

Memory-based algorithms essentially are heuristics that make rating predictions based on the entire collection of previously rated items by the users. That is, the value of the

unknown rating $r_{c,s}$ for user c and item s is usually computed as an aggregate of the ratings of some other (usually, the N most similar) users for the same item s:

$$r_{c,s} = aggr_{c' \in \hat{C}} \quad r_{c',s} \tag{2.6}$$

where $\hat{C}$ denotes the set of N users that are the most similar to user c and who have rated item s (N can range anywhere from 1 to the number of all users). Some examples of the aggregation function are:

$$r_{c,s} = \frac{1}{N} \sum_{c' \in \hat{C}} \quad r_{c',s} \tag{2.7}$$

$$r_{c,s} = K \sum_{c' \in \hat{C}} \quad sim(c, c^{'}) \times r_{c',s} \tag{2.8}$$

$$r_{c,s} = \bar{r_c} + K \sum_{c' \in \hat{C}} \quad sim(c, c^{'}) \times (r_{c',s} - \bar{r_{c'}}) \tag{2.9}$$

Where

k=normalizing Factor and selected as

$$k = \frac{1}{\sum_{c' \in \hat{C}} |sim(c, c^{'})|} \tag{2.10}$$

$\hat{r_c}$= average rating of user C.

for item based:

$$r_{c,s} = \frac{\sum_{s' \in \hat{S}} \quad sim(s, s^{'}) \times r_{c,s'}}{\sum_{s' \in \hat{S}} \quad |sim(s, s^{'})|} \tag{2.11}$$

Where, $\hat{S}$ denotes set of N most similar item to item s. Advantages of Collaborative Filtering Recommenders:

1. Collaborative filtering can perform well in cases where there is not much data or content associated with items.

2. Recommender systems built solely on collaborative filtering approach have the ability to provide recommendations that are relevant to the user, but do not contain content from the user's profile.

Disadvantages of Collaborative Filtering Recommenders:

1. New User: There needs to be enough other users already in the system to find a match for a particular user.

2. Sparsity: Most users do not rate most items and hence the user-item matrix is typically very sparse. If there are many items to be recommended and even if there are many users, because the user-ratings matrix is sparse, it is hard to find users that have rated the same items.

3. First Rater: It is not possible to recommend an item that has not been rated before. This problem comes for new items mostly.

4. Popularity Bias: Collaborative filtering cannot recommend items to someone with unique taste. In general, there is a tendency to recommend the popular items.

2.3.4 Use Tag Information In Collaborative Filtering

With the increasing popularity of the collaborative tagging systems, tags could be interesting and useful information to enhance RS algorithms. Unlike attributes which are global descriptions of items, tags are local descriptions of items given by the users. For example popular web services such as Flickr,Last.fm,Gmail etc..., provide possibility for users to tag or label an item of interest. Content information used in attribute aware RS algorithms is typically attached to the items and is usually provided by domain experts. Therefore, an item always has the same attributes among all users. On the other hand, tags are provided by various users[15]. Thus, tags are not only associated to the items but also to the users. There is two different approach that can be implemented are:User based and Item based.
Weighted Tag Recommender(WTR) [16] exploits tag data but does not use ratings data and other information available about the items. Weighted Tag Rating Recommender (WTRR)[2] combines tag and ratings. However being user based approach, it can not use other information available about the movies. In WTRR while predicting the rating it uses only ratings of those movies, which are tagged also.

2.4 Hybrid Techniques

In general, hybrid Recommenders are systems that combine multiple recommendation techniques together to achieve better performance and to eliminate disadvantages in

recommendation system. Hybrid systems aim to take advantage of all techniques (that are combined together) and obtain more accurate predictions[9]. Different ways to combine collaborative and content-based methods into a hybrid Recommender system can be classified as follows:[20]

1. Implementing collaborative and content-based methods separately and combining their predictions.

2. Use some content-based characteristics into a collaborative approach.

3. Use some collaborative characteristics into a content-based approach.

4. Make a general unifying model that use both content-based and collaborative characteristics.

Chapter 3

Related work

This chapter focuses on the related work in the area of the recommendation systems. Several applications will be reviewed from different types of resources in three main categories; these are content-based, collaborative filtering , Hybrid.

3.1 Content-Based Recommendation Systems

The content-based algorithms use information about an item to suggest recommendations to the users.It is capable of using ratings and other forms of information about the preference of each user. The effect of multiple information sources gives more accurate recommendations rather than exploiting a narrower amount of content[23].

Feature weighting system selects item attributes from IMDB database for movie recommendations such as genre, cast, writer, etc.. Due to the interest of users for these features, several weight values are set by using regression analysis. Therefore, this regression equation is used for computing similarity of items. It is seen in the empirical analysis; the proposed method gives better results than other state-of-art content-based recommendation systems[2].

3.2 Collaborative Filtering Recommendation Systems

One of the first recommendation systems Tapestry[1] is an experimental mail system which recommends messages to the user when selecting relevant document.

Another pioneering and ongoing effort is GroupLens (www.grouplens.org) that uses user ratings for the weight computations of users or items and then predicts ratings of new items based on those weights. This application is also one of the first memory-based

recommendation systems. Based on the similar news reader clients for an active user, the recommendation engine presents articles to the actual user.

A pure collaborative filtering approach can use an algorithm based on neighborhood. This kind of application selects the most similar users for a test user. Then weights of their ratings are calculated and lastly, combination of them is used for prediction process to the test user.

Furthermore, MovieLens, is web-based application that serves movie recommendations to the users.

Amazon.com, item recommendation system is an example for e-commerce web applications. It focuses on matching similar users to an actual user. Amazon first explores similarity between other items and each purchased or rated products of the user, afterwards it gives a list of recommendations based on the combination of those similar items. One of the main strengths of item-based collaborative filtering approaches is that it decreases large amount of dataset when computing similarity weights. For instance, MovieLens dataset has approximately 7-fold more users than items. In this case, item-based collaborative filtering can naturally decrease the scalability which is a well-known problem in collaborative filtering techniques.

Last.fm(www.last.fm/), is another web-based application and aims to recommend tracks to the end users. Each user can listen or scrobble (The Scrobbler sends little note about which song the listener is playing at this moment) a track recorded by a musician. This system has collected data in two ways; Storing music content and generating user profile. Hence, both user and item-based similarity values can easily be calculated.

Models are developed using data mining, machine learning algorithms to find patterns based on training data. These are used to make predictions for real data. There are many model-based CF algorithms. These include Bayesian networks, clustering models, latent semantic models such as singular value decomposition, probabilistic latent semantic analysis, Multiple Multiplicative Factor, Latent Dirichlet allocation and markov decision process based models[11].

3.3 Hybrid Recommendation Systems

There are several hybrid applications such as MOVIES2GO (www.movies2go.net/) which combines ratings of the items with the movie feature and uses semantic and web content.

Hydra [12] is a web-based movie recommendation system, is a hybridization of collaborative and content-based filtering approaches. MovieLens user item rating dataset is used in Hydra and it is also supplemented by content features of movies served in IMDB. In addition, Hydra focuses on reducing system runtime cost in terms of supporting by singular value decomposition (SVD) algorithm. SVD is one of the methods passed in the dimensionality reduction in recommendation systems. The main aim is to factorize matrix and to obtain narrower effective dataset as an input for prediction process. Further, demographic information of users are considered when retrieving data from MovieLens. It is clear that privacy policies are paid attention by MovieLens researchers while they have collected data such as age, gender, etc. for relevant users.

Another hybrid method is the content-boosted collaborative filtering approach also used for movie recommendations. Further, this system is contributed by missing data prediction and local and global similarity. The proposed method mainly concerns in the data sparsity problem. ReMovender [5] is an example of web-based recommendation system for providing movies which satisfies users' taste. It collects several features of movies from IMDB database to calculate item similarity. The current movie's language, country, cast or writer features have different importance for the end users.

Chapter 4

Problem Discussion and Approach

4.1 Problem Discussion

The book Recommender system proposed by Liang [16] is built from tag information only. The authors state that tags can capture the content information of items. However, tags are sometimes meaningful only to the users that assigned them. They can be ambiguous and can also have a lot of synonyms. In paper [16] authors developed a way of addressing these problems by expanding the tag set.

Our approach is extension to Weighted Tag Recommender (WTR)[16] and Weighted Tag Rating Recommender (WTRR) [2]. WTR exploits tag data but does not use ratings data and other information available about the items. WTRR combines tags and ratings. Tags may not always capture the true preferences of users, so the actual ratings are used. One main difference in WTR and WTRR is, instead of simply counting the number of times a user u_i has tagged an item with the tag t_x (as done in [16]),ratings are added of those movies which are tagged by tag t_x by user u_i. However being user based approach, it can not use other information available about the items. In WTRR while predicting the rating it uses only ratings of those movies, which are tagged also.

The other information like Genre,Star cast and Director, which capture content information can also be helpful for better prediction. So, we also incorporate these information to find similar items. Rather than taking only rating information of those movies, which are tagged also, our approach uses all available rating information.

4.2 Approach

The user set U =$\{u_1, u_2, ..., u_{|U|}\}$ contains all the users that tagged movies in hetrec2011-movielens-2k dataset [7]. The movie set M =$\{m_1, m_2, ..., m_{|M|}\}$ contains all movies from the corpus, the tag set T =$\{t_1, t2, ..., t_{|T|}\}$ contains all the tags used by the users in U to label movies in M. Finally, we denote by R =$\{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5\}$ the set of all possible ratings that users can give.

There are four steps that we need to consider in order to construct a tag Movie profile.

1. Calculate the relevance of a tag to a movie as a weight.
2. Calculate relevance of a tag to a user as a weight.
3. Using such weights, we next estimate relatedness between two tags.
4. Finally use the relatedness information to construct Item profiles.

Relevance of a Tag to a Movie

As our approach is to capture ratings, in addition to tags. The modified movie tag relevance weight equation is based on ratings rather than simple counts. To be able to use both ratings and tags, we must ensure that the user who tagged a movie, also rated that movie. In other words, a movie must be both tagged and rated by a particular user. The adapted formula for calculating movie tag relevance weight using the approach is:

$$w_{m_i}(t_x) = \frac{\sum_{u_j \epsilon U_{m_i,t_x}} r_{u_j,t_x}(m_i)}{\sum_{u_j \epsilon U_{m_i,t_y \epsilon T_{m_i}}} r_{u_j,t_y}(m_i)} \tag{4.1}$$

where the numerator is a summation of the ratings $r_{u_j,t_x}(m_i)$ assigned to the movie m_i by all the users u_j who used t_x to annotate it. The set of users who used t_x to tag m_i is denoted by U_{m_i,t_x}. The denominator represents a summation of all the ratings from the users who tagged m_i. The value of $w_{m_i}(t_x)$ now captures the true popularity of the tag$t_x$with respect to a movie m_i.

Relevance of Tag to a User

It signifies how strongly the user feels about a tag. The Equation is:

$$w_{u_i}(t_x) = \frac{\sum_{m_j \epsilon M_{u_i,t_x}} r_{u_j,t_x}(m_j)}{\sum_{m_j \epsilon M_{u_i,t_y \epsilon T_{u_i}}} r_{u_i,t_y}(m_j)} \tag{4.2}$$

where the numerator is a summation of the ratings assigned to the movie m_j by all the users who used$t_x$to annotate it, and the denominator is the summation over all ratings assigned to the movie m_j by all the users who tagged it.

Tag Relatedness Metric for a Movie

Given the relevance of a tag with respect to a user, we can calculate the relatedness of two tags with respect to a movie, as explained below. The relatedness metric is used when constructing the Movie profiles, in order to avoid semantic ambiguity.

The relatedness metric between two tags is denoted by $c_{m_i}(t_x, t_y)$ and represents the degree of correspondence (or connection) between t_x andt_y with respect to movie mi. It measures how similar tag t_y is to a given tag t_x, in the content of a movie mi. The equation to calculate the tag relatedness metric is given by:

$$c_{m_i}(t_x, t_y) = \frac{1}{|U_{m_i,t_x}|} \sum_{u_j \epsilon U_{m_i,t_x}} w_{u_j}(t_y) \tag{4.3}$$

Where U_{m_i,t_x} set of users that tagged movie m_i with t_x. We should note that the tag relatedness metric is not symmetric, in the sense that $c_{m_i}(t_x, t_y)$ is not always equal to $c_{m_i}(t_y, t_x)$, as the set U_{m_i,t_x} can be different from the set U_{m_i,t_y} .

Tag Movie Profiles

As tags related to t_y are believed to be representative for movie m_i, the weight (relevance) of tag t_y for a movie m_i is calculated as summation of relatedness between the tags used by movie m_i (i.e., $t_x \epsilon T_{m_i}$) and target tag t_y;$W_{u_i}(t_y)$ is the total relevance weight of t_y for the movie m_i and is given by:

$$W_{m_i}(t_y) = \sum_{t_x \epsilon T_{m_i}} w_{m_i}(t_x) c_{m_i}(t_x, t_y) \tag{4.4}$$

Similar to inverse document frequency in information retrieval, a tag's occurrence for all movies must be taken into consideration in order to measure the general importance of a tag in the topic preference identification of a movie; imf(t_y) is the inverse movie frequency

of tag t_y and is given by:

$$imf(t_y) = \frac{1}{log(e + \left|M_{t_y}\right|)} \tag{4.5}$$

Where $\left|M_{t_y}\right|$is the number of movies that is tagged with t_y and e is Euler's number thus, $0 < imf(t_y) < 1$.

The tag representation of each movie is defined as:

$$M_i^T = \{W_{m_i}(t_y) \quad imf(t_y)\} \tag{4.6}$$

Neighborhood Formation

In order to predict how much a user will enjoy an unseen movie, in other words to predict their rating for it, we first set out to find the list of movies sharing similar taste. The main goal is to identify for each movie m, an ordered list of N most similar movie, M =$\{m_1, m_2, ..., m_{|M|}\}$ such that $m \epsilon M$ and sim(m, m1) is maximum, sim(m,m2) is the second highest and so on. The N-nearest movies are selected based on the similarity value.

Each movie is encoded with their own topic preferences (captured by tags) and user preferences (captured by ratings). The similarity between two movies based on movie topic preference is denoted as $sim_m^T(m_i, m_j)$ where T is the sets of tags.

We use cosine similarity to compute the angle between the two movie vectors, which are represented by the set of all tags with weights representing them. Let m_i and m_j be the two weighted tag vector, then $sim_m^T(m_i, m_j)$ is given by:

$$sim_m^T(m_i, m_j) = \frac{\sum_{y=1}^{|T|} \; m_{i,y} m_{j,y}}{\sqrt{(\sum_{y=1}^{|T|} m_{i,y}^2)(\sum_{y=1}^{|T|} m_{j,y}^2)}} \tag{4.7}$$

Whereas, the similarity between two movies based on movie user preference is denoted as $sim_m^U(m_i, m_j)$ where U is the set of all users. Movie User preference takes the popularity of movie into the consideration for two users and is given by:

$$sim_m^U(m_i, m_j) = \frac{\sum_{U_k \epsilon U_{m_i} \cap U_{m_j}} imf(U_k)}{\sqrt{|U_{m_i}| \quad \left|U_{m_j}\right|}} \tag{4.8}$$

Where, $|U_{m_i}|$ is number of user who tagged movie mi,$imfU_k$ is the inverse movie

frequency of user U_k and is defined as $imfU_k = \frac{1}{log(e+|M_{U_k}|)}$ where M_{U_k} is the number of movies which is tagged by user U_k.

Given the topic and user profiles, the similarity between two movies is given by:

$$sim(m_i, m_j) = \omega \; sim^T(m_i^T, m_j^T) + (1-\omega) sim^U(m_i^U, m_j^U) \tag{4.9}$$

ω is a weighting parameter such that $0 < \omega < 1$. This parameter omega controls the extent of the collaborative dimension of the algorithm. As we decrease the value of ω, the algorithm will be predominantly collaborative, as the contribution of the movies user preferences will dominate. During the experimental phase, we vary omega.

Rating Prediction Formula

$$r_{u,m} = \frac{\sum_{v \epsilon N(m)} sim(m,v) \; r_{u,v}}{\sum_{v \epsilon N(m)} |sim(m,v)|} \tag{4.10}$$

where sim(m,v) is similarity between movie m and v ,N(m) is set of most similar movies to movie m.

Block Diagram of Proposed Algorithm

Two Matrices:(1)Sub Rating Matrix (2)User Movie Tag matrix

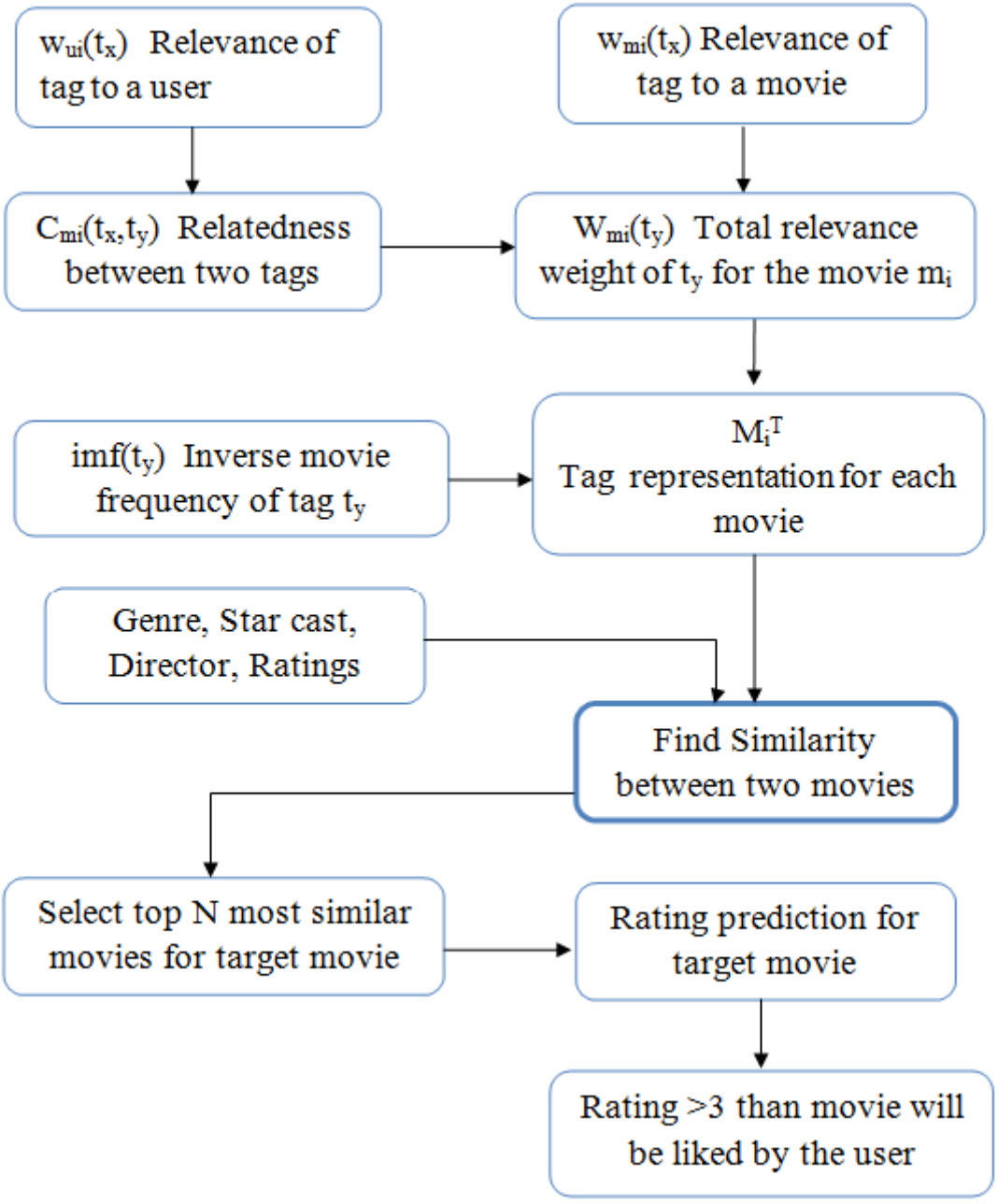

Figure 4.1: Block Diagram of Proposed Algorithm

Two matrices Sub Rating matrix (ratings of those movies are there which are tagged also) and User Movie Tag matrix is used to calculate relevance of a tag to a user ($w_{u_i}(t_x)$) and relevance of tag to a movie ($w_{m_i}(t_x)$). Relatedness between two tags $C_{m_i}(t_x, t_y)$ is calculated using ($w_{u_i}(t_x)$). Using $w_{m_i}(t_x)$ and $C_{m_i}(t_x, t_y)$ we calculate Total relevance weight of t_y for the movie m_i ($W_{m_i}(t_y)$). To take popularity of movie into account we multiply $W_{m_i}(t_y)$ with inverse movie frequency of tag (imf(t_y)) and Tag representation is ready for each movie (M_i^T). Next step is to find similarity between movies using M_i^T, Genre, Star cast, Director, Ratings. We select top N most similar movies for target movie and than predict the rating for target movie. If predicted rating is >3 than movie will be liked by the user.

Chapter 5

Experimental Setup

5.1 Dataset

The data set used in our experiments, hetrec2011-movielens-2k dataset [7] published by the GroupLens research group. The movies in this data set are also referencing their corresponding web pages at the IMDB website. Information about the format as well as statistics regarding the data are available at the hetrec2011-movielens-2k website. There are 2,113 users, 10,197 movies and a total of 13,222 unique tags that fall into 47,957 tag assignment,shows as tuples of the form [user, tag, movie]. There are also 855,598 user ratings ranging from 0.5 to 5.0, in increments of 0.5, thus a total of 10 distinct rating values. There is an average of 405 ratings per user, and 85 per movie. We have 20 Genre type. 20,809 movie genre assignments. 4060 directors and 95321 actors, average 22 actors per movie. We have preprocessed the data and get the user-item rating matrix. We have made sub uirating matrix in which we keep ratings of only those movies which are tagged also. We consider only those actor who worked in more than 2 movies.

5.2 Evaluation Metrics

The performance evaluation matrix are Accuracy, Precision, Recall, Fmeasure. To understand these we first see the confusion matrix, is a specific table layout that allows visualization of the performance of an algorithm. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.

where,

	Actual Class	
Predicted Class	TP	FP
	FN	TN

Table 5.1: Confusion Matrix

TP=case was positive and predicted as positive.

FP=case was negative but predicted as positive.

FN=case was positive but predicted as negative.

TN=case was negative and predicted as negative.

Based upon this the accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5.1}$$

Precision: is the fraction of retrieved instances that are relevant.

$$Precision = \frac{TP}{TP + FP} \tag{5.2}$$

Recall: is the fraction of relevant instances that are retrieved.

$$Recall = \frac{TP}{TP + FN} \tag{5.3}$$

Fmeasure: is the harmonic mean of precision and recall.

$$Fmeasure = 2.\frac{Precision\ .\ Recall}{Precision\ +\ Recall} \tag{5.4}$$

5.3 Experimental Setup

For Item based CF we take 10 items which have been rated by minimum and maximum 20 and 50 users respectively. For evaluation of our algorithm, the original rating dataset is splitted into two sets: training set and testing set. It is ensured that these two subsets are disjoint. For testing, 33% of the ratings are ignored, and we run our algorithm to predict class for testing set. The training set is used to predict all the hidden ratings in the testing set. The ratings of the testing set is kept intentionally hidden so, performance of algorithm can be measured. The Movie which has rating value greater than 3 will be liked by user and less than or equal to 3 will be disliked. Based on this Accuracy, Precision, Recall and Fmeasure can be calculated. The experiment has been performed on different combination and perfomance is measured.

Chapter 6

Results & Discussion

From the graph shown below it has been seen that the accuracy is better when we use uiratings matrix than sub uiratings matrix(keep ratings of only those movies which are tagged also) for prediction task. It has also been seen that when we give different weight to content information of items, means different values of omega(between 0.9 to 0.1), to find top N nearest neighbors and than prediction from uiratings matrix gives better result than conventional approach. Tag information capture the true preference of the users so, the more weight should be given to tag. The proposed Hybrid approach ,which is item based because we want to use features of item (like tag, genre, star cast, director), improves the accuracy of the Recommender system.

Accuracy

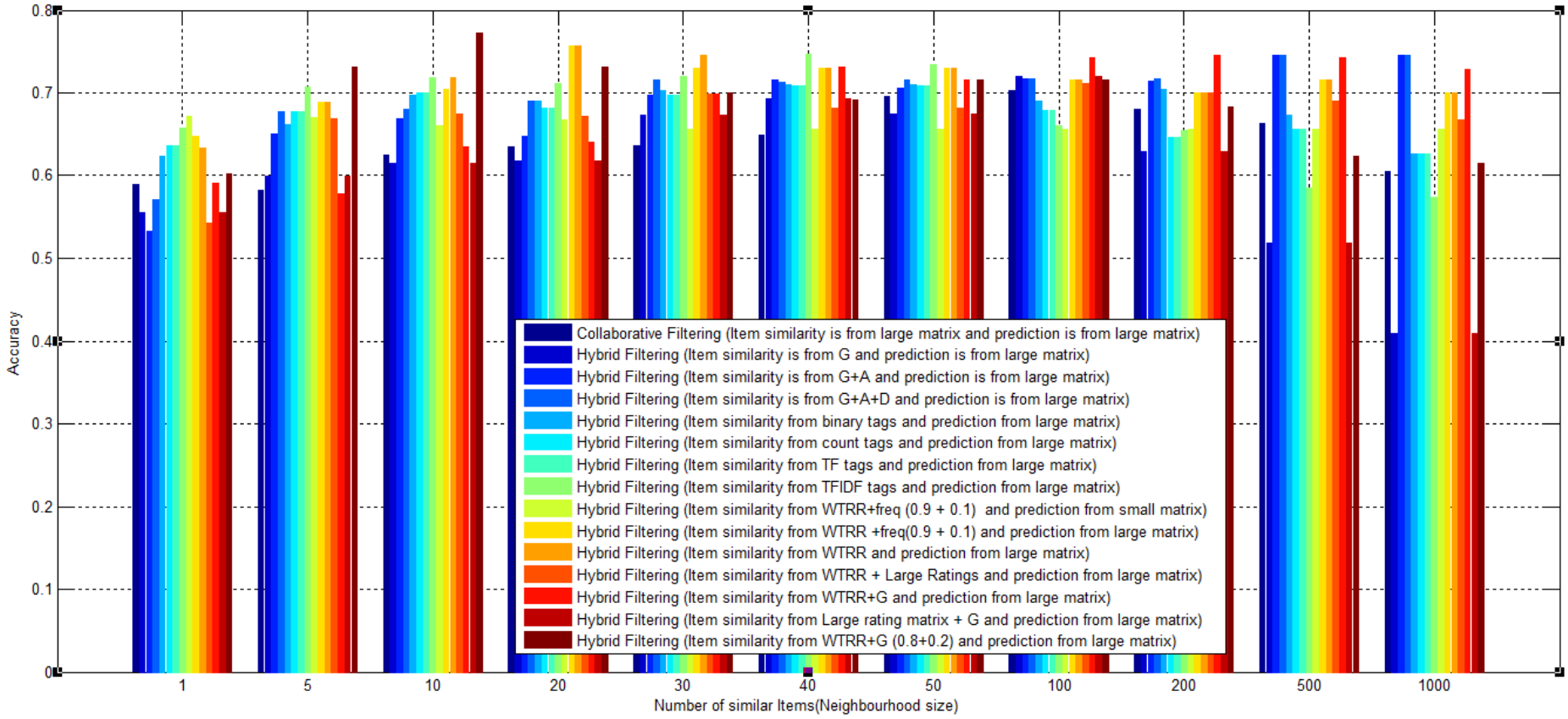

Figure 6.1: Accuracy

Precision

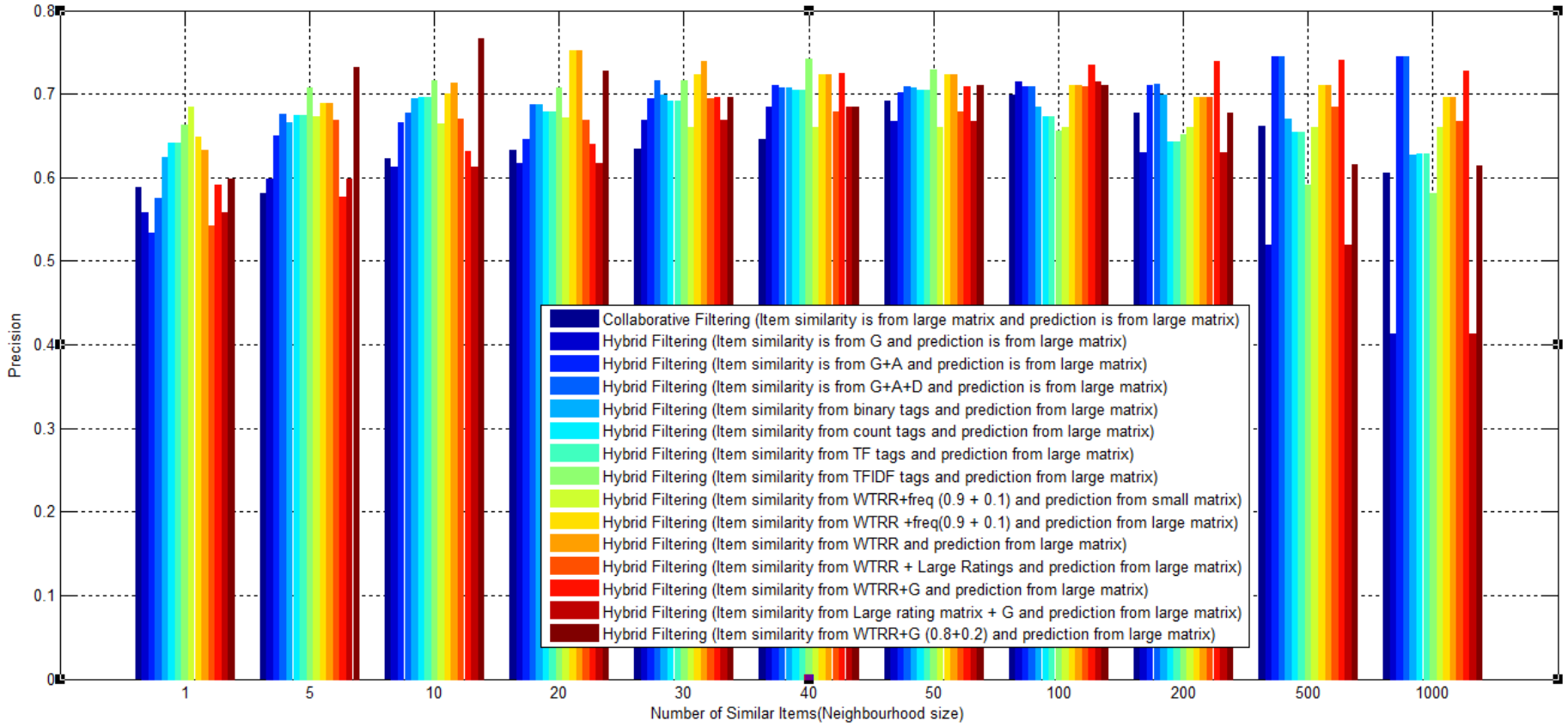

Figure 6.2: Precision

Recall

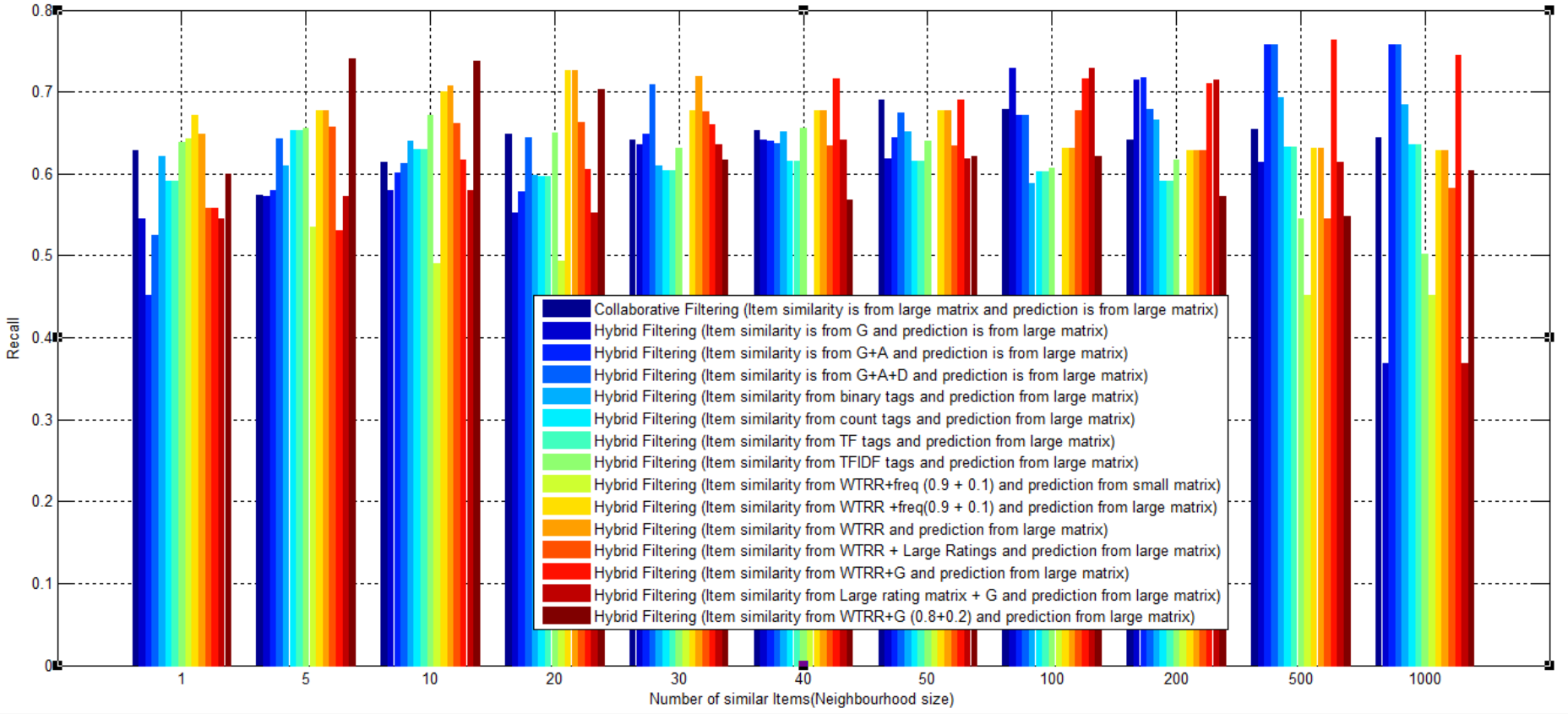

Figure 6.3: Recall

Fmeasure

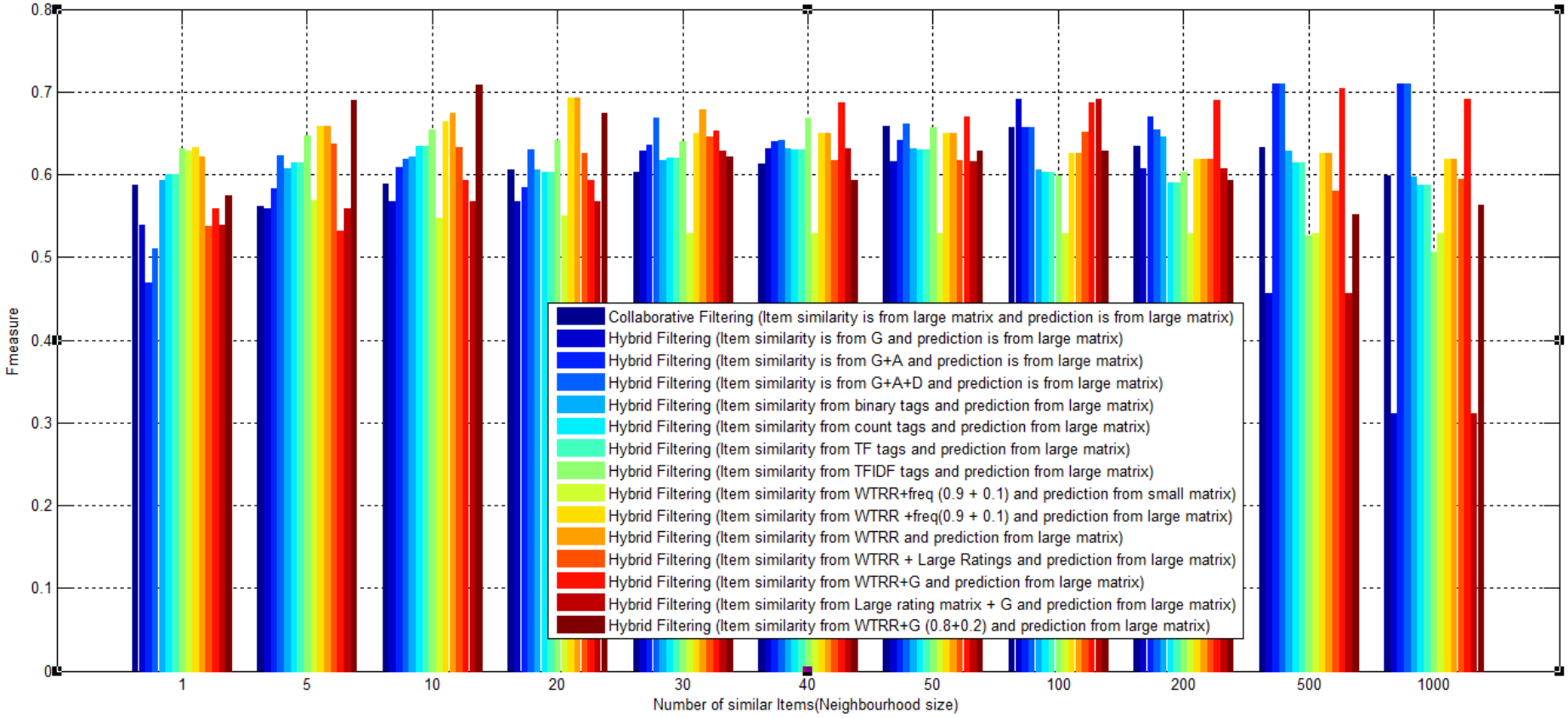

Figure 6.4: Fmeasure

6.1 Collaborative Filtering

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.5898	0.5888	0.6279	0.5879
5	0.5824	0.5806	0.5731	0.5613
10	0.6249	0.6222	0.6142	0.5893
20	0.6346	0.6322	0.6485	0.6061
30	0.6365	0.6340	0.6416	0.6031
40	0.6490	0.6462	0.6522	0.6134
50	0.6960	0.6917	0.6908	0.6588
100	0.7024	0.6981	0.6781	0.6570
200	0.6797	0.6774	0.6416	0.6339
500	0.6630	0.6617	0.6548	0.6324
1000	0.6055	0.6058	0.6442	0.5983

Table 6.1: Collaborative Filtering

6.2 Hybrid Filtering(Item similarity is from Genre and prediction is from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.5560	0.5575	0.5448	0.5395
5	0.5995	0.5984	0.5719	0.5584
10	0.6150	0.6126	0.5790	0.5672
20	0.6182	0.6171	0.5527	0.5674
30	0.6735	0.6684	0.6355	0.6290
40	0.6923	0.6848	0.6411	0.6313
50	0.6746	0.6673	0.6186	0.6156
100	0.7198	0.7151	0.7291	0.6909
200	0.6297	0.6300	0.7142	0.6066
500	0.5180	0.5198	0.6145	0.4560
1000	0.4095	0.4128	0.3685	0.3104

Table 6.2: Hybrid Filtering(Item similarity is from Genre and prediction is from rating matrix)

6.3 Hybrid Filtering(Item similarity is from Genre+Star cast and prediction is from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.5324	0.5336	0.4518	0.4695
5	0.6502	0.6495	0.5789	0.5830
10	0.6682	0.6656	0.6014	0.6086
20	0.6473	0.6453	0.5776	0.5847
30	0.6971	0.6941	0.6482	0.6358
40	0.7152	0.7106	0.6399	0.6404
50	0.7057	0.7019	0.6445	0.6417
100	0.7176	0.7092	0.6712	0.6575
200	0.7144	0.7105	0.7180	0.6691
500	0.7454	0.7446	0.7575	0.7097
1000	0.7454	0.7446	0.7575	0.7097

Table 6.3: Hybrid Filtering(Item similarity is from Genre+Star cast and prediction is from rating matrix)

6.4 Hybrid Filtering(Item similarity is from Genre+Star cast+Director and prediction is from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.5704	0.5755	0.5245	0.5105
5	0.6770	0.6757	0.6429	0.6234
10	0.6807	0.6778	0.6120	0.6189
20	0.6894	0.6872	0.6439	0.6297
30	0.7162	0.7159	0.7088	0.6678
40	0.7128	0.7070	0.6369	0.6414
50	0.7152	0.7085	0.6738	0.6610
100	0.7176	0.7092	0.6712	0.6575
200	0.7176	0.7113	0.6792	0.6540
500	0.7454	0.7446	0.7575	0.7097
1000	0.7454	0.7446	0.7575	0.7097

Table 6.4: Hybrid Filtering(Item similarity is from Genre+Star cast+Director and prediction is from rating matrix)

6.5 Hybrid Filtering(Item similarity from binary tags and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6232	0.6236	0.6213	0.5931
5	0.6613	0.6664	0.6092	0.6076
10	0.6976	0.6945	0.6401	0.6213
20	0.6899	0.6868	0.5985	0.6062
30	0.7024	0.6990	0.6091	0.6165
40	0.7101	0.7067	0.6507	0.6317
50	0.7101	0.7067	0.6507	0.6317
100	0.6899	0.6849	0.5880	0.6050
200	0.7042	0.6992	0.6662	0.6450
500	0.6726	0.6696	0.6934	0.6289
1000	0.6257	0.6270	0.6843	0.5971

Table 6.5: Hybrid Filtering(Item similarity from binary tags and prediction from rating matrix)

6.6 Hybrid Filtering(Item similarity from count tags and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6361	0.6417	0.5910	0.5998
5	0.6768	0.6740	0.6533	0.6139
10	0.7007	0.6958	0.6299	0.6336
20	0.6822	0.6792	0.5965	0.6024
30	0.6965	0.6922	0.6046	0.6195
40	0.7090	0.7043	0.6152	0.6299
50	0.7090	0.7043	0.6152	0.6299
100	0.6781	0.6733	0.6027	0.6027
200	0.6465	0.6432	0.5905	0.5894
500	0.6566	0.6549	0.6321	0.6141
1000	0.6257	0.6287	0.6356	0.5879

Table 6.6: Hybrid Filtering(Item similarity from count tags and prediction from rating matrix)

6.7 Hybrid Filtering(Item similarity from TF tags and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6361	0.6417	0.5910	0.5998
5	0.6768	0.6740	0.6533	0.6139
10	0.7007	0.6958	0.6299	0.6336
20	0.6822	0.6792	0.5965	0.6024
30	0.6965	0.6922	0.6046	0.6195
40	0.7090	0.7043	0.6152	0.6299
50	0.7090	0.7043	0.6152	0.6299
100	0.6781	0.6733	0.6027	0.6027
200	0.6465	0.6432	0.5905	0.5894
500	0.6566	0.6549	0.6321	0.6141
1000	0.6257	0.6287	0.6356	0.5879

Table 6.7: Hybrid Filtering(Item similarity from TF tags and prediction from rating matrix)

6.8 Hybrid Filtering(Item similarity from TF*IDF tags and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6581	0.6626	0.6385	0.632 0
5	0.7066	0.7067	0.6563	0.6476
10	0.7191	0.7157	0.6718	0.6541
20	0.7114	0.7080	0.6493	0.6408
30	0.7197	0.7164	0.6318	0.6400
40	0.7465	0.7416	0.6550	0.6690
50	0.7340	0.7294	0.6399	0.6571
100	0.6596	0.6558	0.6073	0.5991
200	0.6548	0.6518	0.6172	0.6030
500	0.5846	0.5912	0.5446	0.5262
1000	0.5739	0.5805	0.5015	0.5061

Table 6.8: Hybrid Filtering(Item similarity from TF*IDF tags and prediction from rating matrix)

6.9 Hybrid Filtering(Item similarity from WTRR+freq with omega 0.9 and prediction from sub rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6723	0.6850	0.6424	0.6287
5	0.6698	0.6723	0.5347	0.5686
10	0.6596	0.6637	0.4911	0.5469
20	0.6680	0.6714	0.4930	0.5508
30	0.6555	0.6594	0.4513	0.5287
40	0.6555	0.6594	0.4513	0.5287
50	0.6555	0.6594	0.4513	0.5287
100	0.6555	0.6594	0.4513	0.5287
200	0.6555	0.6594	0.4513	0.5287
500	0.6555	0.6594	0.4513	0.5287
1000	0.6555	0.6594	0.4513	0.5287

Table 6.9: Hybrid Filtering(Item similarity from WTRR+freq with omega 0.9 and prediction from sub rating matrix)

6.10 Hybrid Filtering(Item similarity from WTRR +freq with omega 0.9 and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6472	0.6484	0.6721	0.6332
5	0.6891	0.6890	0.6769	0.6587
10	0.7036	0.6995	0.7005	0.6639
20	0.7570	0.7517	0.7255	0.6931
30	0.7292	0.7228	0.6765	0.6498
40	0.7292	0.7228	0.6765	0.6498
50	0.7292	0.7228	0.6765	0.6498
100	0.7149	0.7098	0.6316	0.6258
200	0.7006	0.6959	0.6283	0.6184
500	0.7149	0.7098	0.6316	0.6258
1000	0.7006	0.6959	0.6283	0.6184

Table 6.10: Hybrid Filtering(Item similarity from WTRR +freq with omega 0.9 and prediction from rating matrix)

6.11 Hybrid Filtering(Item similarity from WTRR and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6329	0.6332	0.6484	0.6212
5	0.6891	0.6890	0.6769	0.6587
10	0.7179	0.7134	0.7074	0.6747
20	0.7570	0.7517	0.7255	0.6931
30	0.7459	0.7395	0.7188	0.6779
40	0.7292	0.7228	0.6765	0.6498
50	0.7292	0.7228	0.6765	0.6498
100	0.7149	0.7098	0.6316	0.6258
200	0.7006	0.6959	0.6283	0.6184
500	0.7149	0.7098	0.6316	0.6258
1000	0.7006	0.6959	0.6283	0.6184

Table 6.11: Hybrid Filtering(Item similarity from WTRR and prediction from rating matrix)

6.12 Hybrid Filtering(Item similarity from WTRR + Rating matrix and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.5428	0.5418	0.5585	0.5367
5	0.6693	0.6692	0.6576	0.6368
10	0.6739	0.6695	0.6607	0.6331
20	0.6721	0.6687	0.6628	0.6258
30	0.6989	0.6946	0.6758	0.6450
40	0.6822	0.6780	0.6336	0.6168
50	0.6822	0.6780	0.6336	0.6168
100	0.7117	0.7081	0.6769	0.6507
200	0.7006	0.6959	0.6283	0.6184
500	0.6899	0.6847	0.5452	0.5802
1000	0.6678	0.6666	0.5828	0.5947

Table 6.12: Hybrid Filtering(Item similarity from WTRR + Rating matrix and prediction from rating matrix)

6.13 Hybrid Filtering(Item similarity from WTRR+Genre and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.5907	0.5905	0.5583	0.5592
5	0.5781	0.5769	0.5314	0.5323
10	0.6341	0.6309	0.6163	0.5928
20	0.6408	0.6392	0.6053	0.5927
30	0.6991	0.6957	0.6599	0.6523
40	0.7316	0.7246	0.7155	0.6866
50	0.7162	0.7092	0.6906	0.6691
100	0.7430	0.7353	0.7165	0.6869
200	0.7448	0.7384	0.7100	0.6902
500	0.7430	0.7409	0.7637	0.7041
1000	0.7287	0.7279	0.7441	0.6912

Table 6.13: Hybrid Filtering(Item similarity from WTRR+Genre and prediction from rating matrix)

6.14 Hybrid Filtering(Item similarity from rating matrix + Genre and prediction from rating matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.5560	0.5575	0.5448	0.5395
5	0.5995	0.5984	0.5719	0.5584
10	0.6150	0.6126	0.5790	0.5672
20	0.6182	0.6171	0.5527	0.5674
30	0.6735	0.6684	0.6355	0.6290
40	0.6923	0.6848	0.6411	0.6313
50	0.6746	0.6673	0.6186	0.6156
100	0.7198	0.7151	0.7291	0.6909
200	0.6297	0.6300	0.7142	0.6066
500	0.5180	0.5198	0.6145	0.4560
1000	0.4095	0.4128	0.3685	0.3104

Table 6.14: Hybrid Filtering(Item similarity from rating matrix + Genre and prediction from rating matrix)

6.15 Hybrid Filtering (Item similarity from WTRR+Genre (0.8+0.2) and prediction from large matrix)

Neighborhood Size	Accuracy	Precision	Recall	Fmeasure
1	0.6015	0.5976	0.5995	0.5746
5	0.7316	0.7313	0.7407	0.6892
10	0.7726	0.7659	0.7369	0.7085
20	0.7316	0.7268	0.7029	0.6737
30	0.7006	0.6964	0.6173	0.6215
40	0.6913	0.6848	0.5686	0.5935
50	0.7149	0.7103	0.6206	0.6289
100	0.7149	0.7103	0.6206	0.6289
200	0.6836	0.6776	0.5720	0.5930
500	0.6235	0.6161	0.5473	0.5513
1000	0.615 0	0.6139	0.6039	0.5627

Table 6.15: Hybrid Filtering (Item similarity from WTRR+Genre(0.8+0.2) and prediction from large matrix)

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this dissertation, We propose a new hybrid approach which combines usage, tag and content data of movies to improve the quality of recommendations. Movie recommendation task is modeled as classification problem where our aim is to predict whether the movie will be liked or disliked by the user. Item based Recommender system proposed by us exploits movie specific data such as movie genres, star cast and directors. Results show that combining the data in right manner, improves the accuracy of Recommender.

7.2 Future Work

Machine learning technique such as decision tree induction will be used to learn a classifier. Feature selection technique such as Information Gain will also be used to control the dimensionality of the data.

Bibliography

[1] Panagiotis Symeonidis, Alexandros Nanopoulos, Apostolos N, Papadopoulos, Yannis, and Manolopoulos. *Collaborative recommender systems: Combining effectiveness and efficiency*, volume 34. Elsevier, 4 edition, 2007.

[2] Swapnill Nagar. A hybrid recommender: User profiling from tags/keywords and ratings. Master's thesis, Rajiv Gandhi Technical University, 2012.

[3] Chhavi Rana Mukta kohar. Survey paper on recommendation system. *(IJCSIT) International Journal of Computer Science and Information Technologies*, 3(2), 2012.

[4] Raminder Jeet Singh Sodhi, Vaibhav Gaur, and V.K. Panchal. Recommendation based on prominent items. *International Conference on Image Information Processing with IEEE*, pages 1–4, 2011.

[5] Gozde Ozbal, Hilal Karaman, and Ferda N Alpaslan. A content-boosted collaborative filtering approach for movie recommendation based on local and global similarity and missing data prediction. *The Computer Journal*, 54(9):1535–1546, 2011.

[6] Liang, Huizhi, Xu, Yue, Li, Yuefeng, Nayak, Richi, Shaw, and Gavin. A hybrid recommender systems based on weighted tags. *International Conference on Data Mining (SDM2010)*, May 2011.

[7] Ivan Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2nd workshop on information heterogeneity and fusion in recommender systems (hetrec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011, New York, NY, USA, 2011. ACM.

[8] Rubi Boim and Tova Milo. Methods for boosting recommender systems. *Data Engineering Workshops (ICDEW), IEEE 27th International Conference*, pages 288–291, 2011.

[9] Daniar Asanov. Algorithms and methods in recommender systems, 2011.

[10] Mustansar Ali Ghazanfar and Adam Prugel-Bennett. A scalable, accurate hybrid recommender system. *Knowledge Discovery and Data Mining, WKDD'10. Third International Conference*, pages 94–98, 2010.

[11] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 1(421425):19, 2009.

[12] Stephan Spiegel, Jerome Kunegis, and Fang Li. Hydra: A hybrid recommender system. *ACM*, 1:75–80, 2009.

[13] Giovanni Semeraro, Pasquale Lops, Pierpaolo Basile, and Marco de Gemmis. Knowledge infusion into content-based recommender systems. *Proceedings of the third ACM conference on Recommender systems, ACM: New York, New York, USA*, pages 301–304, 2009.

[14] Dhoha Almazro, Ghadeer Shahatah, Lamia Albdulkarim, Mona Kherees, Romy Martinez, and William Nzoukou. A survey paper on recommender systems. *arXiv preprint arXiv:1006.5278*, 2009.

[15] Karen H. L. TsoSutter, Leandro Balby Marinho, and Lars SchmidtThieme. Tagaware recommender systems by fusion of collaborative filtering algorithms. *In Proceedings of the 2nd ACM Symposium on Applied Computing*, (9781595937537/08/0003):1995–1999, March 2008.

[16] Huizhi Liang, Yue Xu, Yuefeng Li, and Richi Nayak. Collaborative filtering recommender systems using tag information. *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference*, 3:59–62, 2008.

[17] Toby Miller. Drowning in information and starving for knowledge: 21st century scholarly publishing. *International Journal of Communication*, 1:123–135, 2007.

[18] Shaohong Fu Kangning Wei, Jinghua Huang. A survey of e-commerce recommender systems. *Service Systems and Service Management, 2007 International Conference*, June 2007.

[19] Christina Christakou and Andreas Stafylopatis. A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools*, 16(05):771–792, 2007.

[20] Alexander Tuzhilin Gediminas Adomavicius. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on knowledge and Data Engineering*, 17(6), June 2005.

[21] Kwok-Wai Cheung, James T Kwok, Martin H Law, and Kwok-Ching Tsui. Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35(2):231–243, 2003.

[22] Robin van Meteren and Maarten van Someren. Using content-based filtering for recommendation. Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop, 2000.

[23] William Cohen Chumki Basu, Haym Hirsh. Recommendation as classification:using social and content-based information in recommendation. *Proceedings of the national conference on artificial intelligence*, pages 714–720, 1998.

[24] Marko Balabanovic and Yoav Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, 1997.