REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A HYBRID RECOMMENDATION SYSTEM

AHMED ADEEB JALAL

M.Sc. THESIS
DEPARTMENT OF COMPUTER ENGINEERING
PROGRAM OF COMPUTER ENGINEERING

ADVISER

ASSIST. PROF. DR. OĞUZ ALTUN

ISTANBUL, 2016

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

A HYBRID RECOMMENDATION SYSTEM

A thesis submitted by AHMED ADEEB JALAL in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 29.11.2016 in Department of Computer Engineering, Computer Engineering Program.

Thesis Adviser

Assist. Prof. Dr. Oğuz Altun

Yildiz Technical University

Approved By the Examining Committee

Assist. Prof. Dr. Oğuz Altun

Yildiz Technical University _____

Prof. Dr. Nizamettin Aydın, Member

Yildiz Technical University _____

Assist. Prof. Dr. Cemal Okan Şakar, Member

Bahcesehir University _____

This study supported by the Ministry of Higher Education and Scientific Research of Iraq.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my advisor Dr. Oğuz Altun, for his support and guidance throughout graduate study and research. Dr. Oğuz Altun always tries his best to provide help and encouragements. Also, I would like to thank the department of computer engineering for providing me this precious studying opportunity.

I dedicate this thesis to my family and friends. In particular, I dedicate this thesis to my wife for her tireless love and encouragement over the years. Also, I can't forget the support of Mona Mohamed Wafy employee at Al-Iraqia University, so I would like to express my sincerely appreciation to her.

Special thanks to Al-Iraqia University for giving me this opportunity to get the master's degree.

November, 2016

AHMED ADEEB JALAL

TABLE OF CONTENTS

LIST OF SYMBOLS

Y	Number of items that rated by both user i and user j.
y	Number of items in the data set.
r	Rating values.
$\bar{r}$	Average ratings for all items that rated by user.
n	User's item.
K	Number of users that similar to the user.
u	Number of items that rated by user in the data set.
S	Training sample of item i.
$\vec{I}$	Input vector for all item features.
$\vec{V}$	Output vector for all attribute values of the selected feature.
$\vec{D}$	Output vector for unrepeated attribute values of the selected feature.
TF	Number of times that the attribute values of the selected feature appear in user's profile for rated items.
W	Average ratings based on $\vec{D}$ and TF.
p	Predicted ratings.
T	Total number of predictions generated for all active users.
TP	Acceptable items recommended to user.
TN	Unacceptable items not recommended to user.
FP	Unacceptable items recommended to user.
FN	Acceptable items not recommended to user.

LIST OF ABBREVIATIONS

CAGR	Compound Annual Growth Rate
CFP	Collaborative Filtering Pearson Correlation approach.
CWOCF	Confidence Weighted Online Collaborative Filtering.
HRS-1	Hybrid Recommender Systems, First Technique.
HRS-2	Hybrid Recommender Systems, Second Technique.
MAE	Mean Absolute Error.
RMSE	Root Mean Squared Error.

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABSTRACT

A HYBRID RECOMMENDATION SYSTEM

AHMED ADEEB JALAL

Department of Computer Engineering

M.Sc. Thesis

Adviser: Assist. Prof. Dr. Oğuz Altun

Web growth, especially in social networks, is continuously increasing every day. Multiplicity of products offered and web pages has made picking up relevant items a tedious job. On the other hand, different tastes and behaviors of users is creating the probability to find a similar user among a large group of users difficult. As a result, automated software systems have difficulty to discover what is interesting to users.

We have proposed a new approach to adapt to this flow. We will exploit domain knowledge of training data set to create a summary matrix. The summary matrix consists of new and few columns according to the attribute values of the selected feature. We fill the summary matrix with the average ratings based on the number of times that the attribute values appear in the user's profile for rated items.

We use the summary matrix in two hybrid recommender systems. In our approach, we use meta-level technique which is one of the pipelined hybridization techniques.

The proposed approach will reduce the effects of sparsity, cold start, and scalability which are common problems with the collaborative recommender systems. Also, the proposed approach will improve the recommendation accuracy when there is comparison with the Collaborative Filtering Pearson Correlation approach and it will be faster.

Key words: Data Mining, Big Data, Recommender Systems, Feature Engineering, Hybrid Recommender Systems, Meta-level, Collaborative Filtering, Content-Based Filtering, Sparsity, Cold Start, Scalability.

ÖZET

BİR HİBRİT TAVSİYE SİSTEMİ

AHMED ADEEB JALAL

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Oğuz Altun

İnternet büyümesi her geçen gün gözlenmekte, özellikle de sosyal ağlarda bu durum fazlası ile hissedilmektedir. Sunulan ürünlerin, web sayfalarının fazlalığı, belli bir konuda araştırma yapan kişi için kendi konusu ile alakalı hususları bulmayı oldukça meşakkatli hale getirmektedir. Öte yandan, kullanıcıların farklı tutumları ve tercihleri, büyük bir kullanıcı grubu arasında bir "komşu" kullanıcı bulmayı zorlaştırmaktadır. Bu nedenle otomatik yazılım sistemleri, kullanıcılar için neyin ilgi çekici olduğunu tespit etmekte zorlanmaktadırlar.

Bu akışa uyum sağlamak için yeni bir yaklaşım önerdik. Bir özet matris oluşturmak için eğitim verisi alan bilgisini kullanacağız. Özet matrisi, seçilen özelliğin özellik değerlerine göre yeni ve birkaç sütundan oluşur. Özet matrisini, derecelendirilmiş öğeler için kullanıcının profilinde görünen özellik sayısına göre ortalama derecelendirmelerle doldururuz.

Özet matrisini iki hibrit tavsiye sisteminde kullanırız. Yaklaşımımızda, boru hattı hibritizasyon tekniklerinden biri olan meta-seviye tekniğini kullanıyoruz.

Bizim önerdiğimiz yaklaşım sayesinde, katılımcı tavsiye sistemlerinde sıklıkla görülen seyreklik, soğuk başlangıç ve ölçeklenirlik gibi sorunların etkileri azalacaktır. Ek olarak, tam katılımcı filtreleme yaklaşımı Pearson Korelasyonu yaklaşımı ile karşılaştırıldığında, tavsiyelerin doğruluğunu yükseltmektedir ve daha hızlı olacak.

Anahtar Kelimeler: Veri Madenciliği, Büyük Veri, Tavsiye Sistemleri, Özellik Mühendisliği, Hibrit Tavsiye Sistemleri, Meta-seviye, Işbirlikçi Filtreleme, İçerik bazlı Filtreleme, Seyreklik, Soğuk Başlangıç, Ölçeklenirlik.

CHAPTER 1

INTRODUCTION

In this chapter, we introduce the literature review of data mining, recommender system, and some examples of hybrid recommender systems. We explain the objective of the thesis and the reason that makes recommender systems interesting. We discuss the key idea of our approach, the aim of our approach, and how our approach contributes to solving problems.

1.1 Literature Review

Data mining has attracted a great deal of interest in the digital information industry and in the social network as a whole in recent years. Data mining has gained that significance due to the increase in the amount of available data and the need to discover useful information and knowledge from this data. In this context the data mining term is used to describe the collection of analysis techniques used to infer recommendation rules or build recommendation models from large data sets [41], [42].

Recommender systems are popular intelligent software systems that are applied in various domains such as movies, music, books, jokes, restaurant, financial services [8], and Twitter followers [9]. It recommends interesting items to users [4], [6], [7], [10], [11], [13]. These personalized suggestions are a useful alternative to searching algorithms.

Recommender systems rely on discovering the historical profiles of users. These profiles include information such as rates, item features, tags, and shared files. This profile is compared with other users. It can be distinguished from other information retrieval systems by semantics and systematic analysis to user interactively. Recommendations resulting from recommender systems are interpreted as responding to a user's query at information retrieval systems, therefore the recommender systems can be seen as an information agent [60], [61], [62], [63].

We review some examples of hybrid recommender systems that are applied in various domains. Netflix Inc. [26] for the movie recommendation, combines Collaborative and Content-Based Filtering through similar habits of users and higher rates of shared movies characteristics. Netflix Inc. released a challenge in 2006 and offered a grand prize of one million US dollars to enhance the recommender system of the company [26]. The person or team who could succeed to decrease RMSE for data set by 10 percent, would get the Netflix Inc. prize [1], [2], [5]. Bellkor's Pragmatic Chaos team succeed in achieving an RMSE of 0.8554 with a 10.06% improvement over the Netflix Inc. system [2].

Lawrence et al. [20] described a personalized recommender system to shoppers in supermarkets. This recommender system relies on shoppers previous behavior towards the purchases to suggest new products for them. The IBM researchers developed this recommender system to implement it as a part of SmartPad which was developed as a personal digital assistant for remote shopping.

Paula Cristina and David Martins [22] presented a hybrid book recommender systems based on Collaborative Filtering and author's rankings by users. This hybrid recommender systems improves book recommendations through sending proposals for book readers to decide which book to read next.

MovieLens data set [31] is the online movie recommendations data set that we used in our approach. MovieLens proposes some of the most popular movies to new users to evaluate it. These ratings are exploited to recommend other movies to the user. In addition, MovieLens uses Collaborative Filtering based on these ratings to create personalized recommendations.

We can apply several techniques in the same recommender systems to get the recommendations. For example, two different Content-Based Filtering could work together in hybrid recommender systems such as News Dude. News Dude uses both Naive Bayes and K-Nearest Neighbor classifiers in its news recommendations [16].

Consequently, hybrid recommender systems become increasingly interesting for researchers. Theoretical work focused on how to hybridize the algorithms and which situations can expect to benefit from hybridization [1]. Hybrid recommender systems represent the door to improving the recommendations, overcome some of the problems, and improve the performance of algorithms.

1.2 Objective of the Thesis

Due to the ubiquity of e-commerce, recommender systems have become an exciting area to work on recently. There are many different recommender systems. However, the researchers have yet to create and develop algorithms to reach satisfactory results for users.

Often, users don't have a clear idea about what items which are good for them. Also, the competition between companies makes recommender systems of special interest. These companies compete with one another to market their various items to satisfy consumers in their daily life needs. However, these companies do not know what is acceptable to users. Therefore it is important these companies have the best recommender systems to show the right recommendations for consumers to increase their revenue by increasing their sales.

Many companies develop recommender systems to guide the consumers. Examples of such companies include Netflix Inc. [26] for movie recommendations, Amazon [27] for product recommendations, Last.fm [28] for radio recommendations, and LinkedIn [29] for friend recommendations.

1.3 Hypothesis

Our approach depends on the attribute values of the selected feature in the training data set. This approach, succeeds in modeling to produce the summary matrix with new and few dimensions to be input for hybrid recommender systems. Creating new features is one of feature engineering results [34]. Choosing the right feature is still important because every item has many selectable features.

Our approach reduces the amount of time used to provide the recommendation list for the user. Also, our approach increases the ratings density, which leads to providing greater opportunites for recommender systems to find likenesses between users.

The main structure of the data sets is a two-dimensional matrix that consists of a user-Item ratings matrix. We apply our approach on two data sets MovieLens 1M and HetRec 2011. MovieLens 1M data set consists of 1,000,209 rates that are represented by a matrix of 6040 users and 3883 items. HetRec 2011 data set has 855,598 rates that are represented by a matrix of 2113 users and 10197 items.

In most environments of e-commerce where recommender systems apply, the number of users and items is huge. Therefore, we can consider this problem as a scalability problem. Also, many of the users do not rate their items. Even popular or favorite items are perhaps still unrated, which minimizes opportunities to find a similar person. This problem is called sparsity problem. Sparsity problem effects the full data set matrix. Another problem is related to a single row or column of the data set which has few ratings or none, this problem is considered a cold start problem. We can consider the cold start problem a special case of sparsity problem because, cold start problem effects a single row or column instead of the entire data set. The rows and columns in selected data sets represent the number of users and items.

Our approach tries to reduce these above-mentioned problems. Many researchers over the past several years have come up with different solutions to resolve scalability, cold start, and sparsity problems. These problems are inherent in collaborative recommender systems. Reducing the data set dimensionality is one solution approaches. Sarwar et al. [33] applied singular value decomposition for matrix factorization that provides lowest rank approximations of the original matrix. Singular value decomposition expresses the matrix as the product of three “simple” matrices, which result in the singular values in decreasing order.

Jing Lu et al. [55] proposed Confidence Weighted Online Collaborative Filtering (CWOCF) approach. The key idea of the CWOCF approach is to follow the low-rank matrix factorization and exploit confidence weighted classification in optimizing the low-rank matrixes. The CWOCF approach will update the distributions of matrix factorization vectors.

YiBo Chen et al. [76] proposed to compute the similarity matrix based on relative distance between user ratings to solve the sparsity problem in recommender systems.

Siavash Ghodsi Moghaddam and Ali Selamat [38] proposed a clustering method to solve scalability problem. This method is a hybrid recommender system, which comprises of users' demographic information and Collaborative Filtering.

Iván Cantador et al. [77] proposed a hybrid recommendation model which combines Content-Based and Collaborative Filtering according to relations among users. The proposed approach is based on clusters that are used to find similarities among individuals at multiple semantic layers.

CHAPTER 2

GENERAL INFORMATION

In this chapter, we view the headlines of general information that contributes to our approach. We also note the effect of data growth and the increase of users on the Internet. We can extract useful information from this data by data mining algorithms. Recommender systems are one of the data mining algorithms. We can reduce the data by feature engineering if it is huge to improve performance of recommender systems.

2.1 Data Mining

Data mining is a broad spectrum of computational processes, mathematical modeling techniques [47], and software tools [46] to discover patterns in a large data set. These patterns, previously unknown, represent the summary of the data entered and perhaps are used in further analysis. Data mining term refers to extracting or “mining” knowledge through either automatic or manual methods [43], [44]. Data mining is involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems [45].

Data mining is useful in a discovery scenario of items due to the absence of predetermined principles about what the outcomes will be. Data mining is exploring and mining new, valuable, and nontrivial "interesting" information among large volumes of data. Best results can be achieved by a joint effort of the intelligence of humans in describing problems and goals and computer efficiency. It is possible to determine the primary goals of data mining in one of two categories: prediction and description of patterns [43]. Prediction of patterns exploits some variables or fields in the data set to predict unknown or future values of other variables of interest. Description of patterns focuses on finding the description of data that interprets by humans.

Data mining steps can divide into three simplified processes: pre-processing (data cleaning, integration and selection), data mining, and results validation. Also, we can divide data mining steps into a number of ramified processes, as in [43], [44], [48], [57], [58].

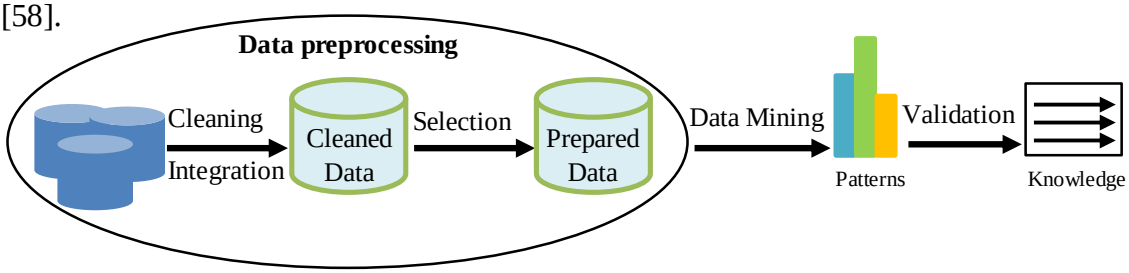


Figure 2.1 Data mining process steps [1].

There are many algorithms used in data mining to find the patterns such as decision tree, classification rules, and clustering techniques [64], [65], [66], [67]. These algorithms can be suitable for a particular type of data set and the results are satisfactory. But, perhaps these algorithms cannot reproduce on a new sample of the data set and bear little use. The final step of data discovery, verifies that the patterns produced by data mining algorithms meet the need of users before turning patterns into knowledge. If the results are not satisfactory, we will need to re-evaluate and change the pre-processing or data mining algorithm.

Several researchers and organizations conducted reviews of data mining tools and surveys to identify some of the strengths and weaknesses of the software packages [46], [49], [50], [51], [52], [54]. The researchers and organizations provide an overview of the behaviors, preferences, challenges, and views of data mining.

2.2 Big Data

Big data or a large-scale data are the outcome of the qualitative boom in computing, communications, and digital storage technologies. This qualitative boom is accompanied by an increase in high-resolution throughput data. Big data refers to data set that are growing rapidly, because of the spread of digital computers, mobile, and expanding the Internet.

Digital information storage capacity, doubles every 40 months, roughly since the 1980s [68]. The storage capacity reached to 2.5 Exabyte (the sixth power of 1000 bytes) every day in 2015 [69], it nearly reached to 3 Exabyte in 2016 [69]. Cisco forecasts indicate to a steady increase in the storage capacity of the coming years [69].

In Figure 2.2, Cisco forecasts of data growth to nearly triple from 2015 to 2020 are illustrated. Cisco expects the data growth will reach to 194 EB per month by 2020, up from 72 Exabyte per month in 2015, with a Compound Annual Growth Rate (CAGR) of 22 percent.

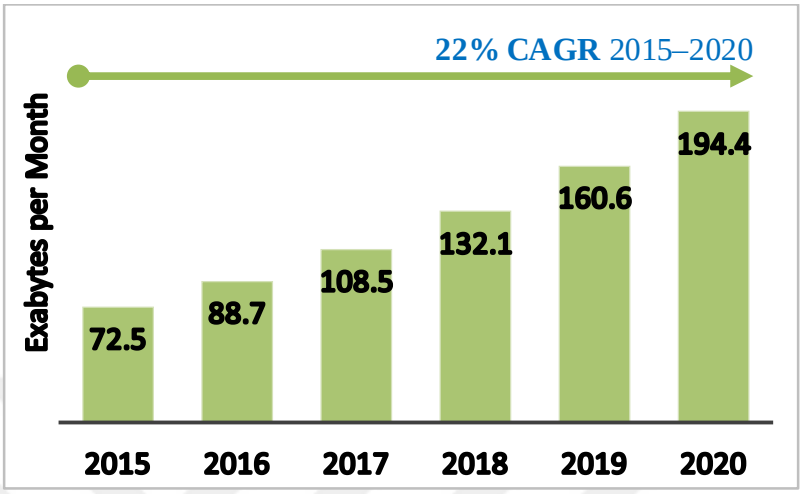


Figure 2.2 Cisco forecasts of data growth [69].

Big data includes several challenges such as storage, transfer, visualization, querying, and updating. These challenges require more predictive analytics, user behavior analytics, or other advanced data analytics methods to discover a useful pattern [70]. Big data is characterized through the quantity and quality of generated and stored data [71], [72].

In Figure 2.3, increasing of Internet users over the past decades for several times is illustrated. Increasing the number of users directly proportional to increasing in the amount of data.

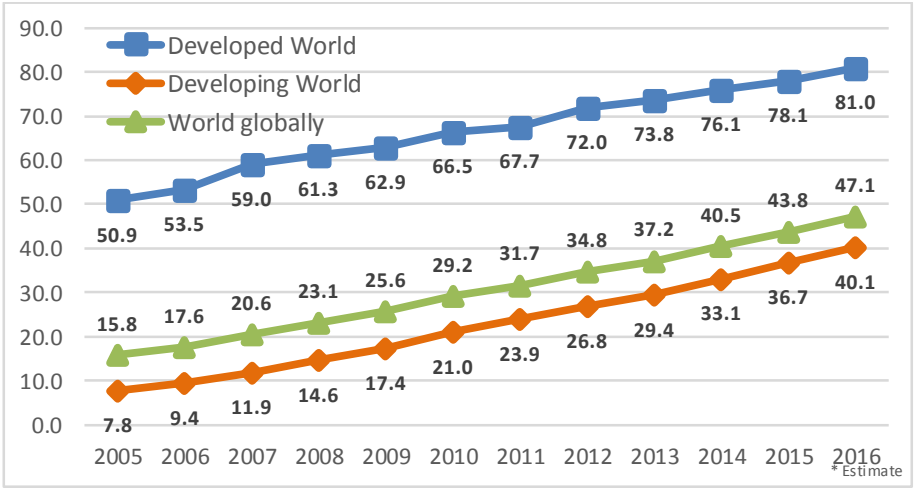
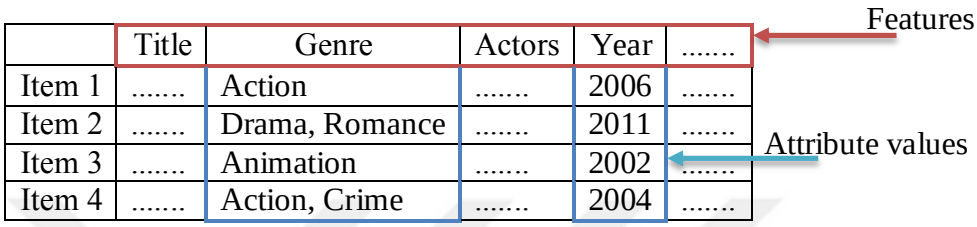


Figure 2.3 Internet users per 100 inhabitants [73].

2.3 Feature Engineering

Feature engineering exploits the domain knowledge of the data set to create new features. These features contribute to make machine learning algorithms work properly. The feature is a piece of information in the data set. This piece might contain many attribute values which are useful for prediction and will influence recommendations. Any attribute values could be a feature, as long as it is useful to the model [24].

In Figure 2.4, some features of movies and its attribute values are illustrated.



	Title	Genre	Actors	Year	.....
Item 1	.....	Action	.....	2006	.....
Item 2	.....	Drama, Romance	.....	2011	.....
Item 3	.....	Animation	.....	2002	.....
Item 4	.....	Action, Crime	.....	2004	.....

Figure 2.4 Features and its attribute values.

The feature is a distinguishing characteristic that might help when analyzing the problem to solve it [17]. The quality and quantity of the features will have great influence on whether the model is good or not [18]. We can clarify the feature engineering steps to create new features, by the following sequential steps [34]:

- Testing features.
- Deciding what features to create.
- Creating features.
- Checking how the features work with your model.
- Improving your features if needed.
- Go back to testing/creating more features until the work done.

Right features chosen require extensive testing to pick up a relevant feature to achieve better results. Right features represent the most important part in machine learning [56]. Right features make a model simpler and more flexible, and they often yield better outcomes [17]. However, the success of the algorithm doesn't only depend on selected features. The model and data set represents an important role in the success of the algorithm to achieve a satisfactory result.

The purpose of choosing right feature in our approach is to reduce the effects of sparsity, cold start, and scalability problems. Alongside, right feature chosen improved the recommendations in our approach and improves it. As a result, we can get user satisfaction.

In Figure 2.5, the features of some types of data sets are illustrated. Any feature has many attribute values depend on the type of feature either numeric or textual.

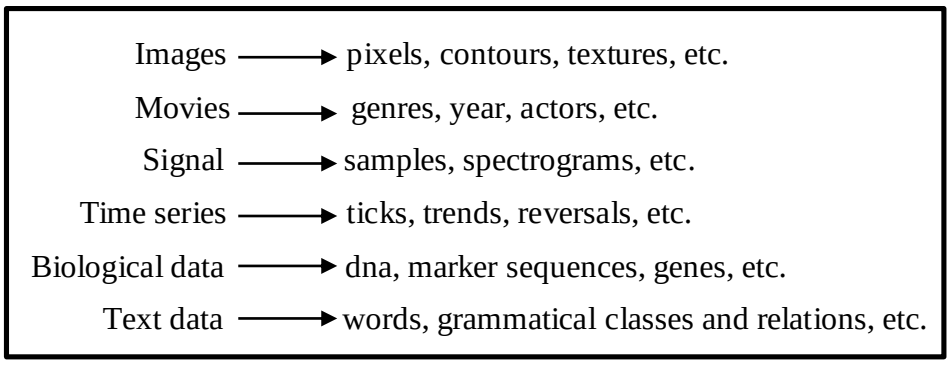


Figure 2.5 Features of some types of data sets [56].

2.4 Recommender Systems

Recommender systems are personalized information agents that have become interesting in recent years. It applies in the domains of academia and industry increasingly. Recommender systems are a subclass of software information filtering systems, which analyzes user profile to predict what the user preference is.

Recommender systems that incorporate data mining techniques, get its recommendations by using knowledge learned from actions and attribute values of users and items. Recommender systems are based on previous information about interaction of the users with items to get the recommendations [59]. The past user concerns determine the user future choices.

There are four techniques of recommender systems: collaborative, content-based, knowledge-based, and demographic [16]. Two main categories are most popular: content-based and collaborative recommender systems. Most recommender systems that apply hybrid recommender systems is a combination of content-based and collaborative recommender systems.

Recommender systems techniques can use feedback on different knowledge sources such as user ratings database, item database, and user's ratings. Knowledge sources depend on information repositories that are stored in a companies' sites online. Information repository contains a user's personal information, the item's information, and ratings. The rating values indicate to the user's preferences.

In Figure 2.6, knowledge sources of two main techniques of recommender systems are illustrated.

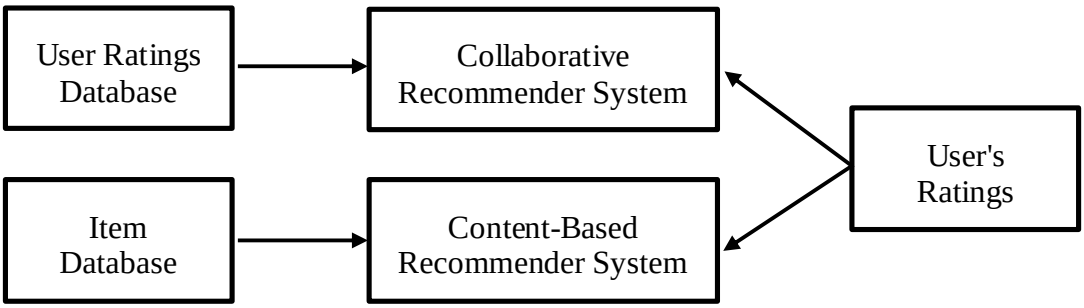


Figure 2.6 Knowledge sources of recommender systems techniques [16].

2.4.1 Collaborative Recommender Systems

Collaborative Filtering approach is the most popular method of recommender systems [1], [10]. Collaborative Filtering generates the recommendations based only on the past users database of ratings that represents full information about users' past rates. Collaborative Filtering predicts preferable items for users by calculating the similarity scores between users. These scores will be interpreted according to the used algorithms. An example of this is, Pearson Correlation approach which interprets the scores based on positive "like" and negative "unlike", on this basis, it will provide a list of recommendations.

Collaborative Filtering avoids semantics and systematically analyzes for items. Therefore, Collaborative Filtering is characterized by quickly and accurately recommendations without considering to the concept of the item itself.

Typically Collaborative Filtering is classified into two methods: memory-based and model-based method. Memory-based methods use the ratings directly to find the likeness between users or items to predict the recommendations. Memory-based methods are based only on ratings, which reflect positively on the efficiency and ease of a method implementation. Model-based methods use ratings to model user-item interactions with latent characteristics. Latent factor indicates latent features that are stored in the database of items and users. Latent factor is sometimes called latent variables that cannot be observed directly. Instead, Latent factor can be inferred from other observed variables. For example, in the movie recommender system, the latent features could be comedy, action, or children topics. Therefore, these latent factor models can be used to predict new items for users.

Many algorithms are applied in measuring user-user similarity and item-item similarity in recommender systems. We apply the algorithms that measure user-user similarity when there are more items than users and item-item similarity when the situation is reverse [53].

Pearson Correlation approach is a one of Collaborative Filtering approaches. The Pearson Correlation approach used to calculate the similarity of two users i and j, and is defined as:

$$S(i,j) = \frac{\sum_{y \in Y}(r_{y,i} - \bar{r}_i) \times (r_{y,j} - \bar{r}_j)}{\sqrt{\sum_{y \in Y}(r_{y,i} - \bar{r}_i)^2} \times \sqrt{\sum_{y \in Y}(r_{y,j} - \bar{r}_j)^2}} \quad (2.1)$$

Where, Y is the number of items that rated by both user i and user j, y is the number of items in the data set, r is the rating values, and $\bar{r}$ is the average ratings.

Pearson Correlation approach depends on identical opinions on ideas and behaviors among users, which is reflected in the ratings.

Three formulas are used to predict ratings depending on the similarity score and the best recommendation is recommended to the user. These formulas are based on the weighted average of all ratings for similar users. Where, K is the number of similar users.

$$r_{i,n} = \frac{\sum_{j \in K} r_{j,n}}{K} \quad (2.2)$$

$$r_{i,n} = \frac{\sum_{j \in K} S(i,j) \times r_{j,n}}{\sum_{j \in K} S(i,j)} \quad (2.3)$$

$$r_{i,n} = \bar{r}_i + \frac{\sum_{j \in K} S(i,j) \times (r_{j,n} - \bar{r}_j)}{\sum_{j \in K} S(i,j)} \quad (2.4)$$

In Figure 2.7, a simple example of Collaborative Filtering approach for groups of users and items is illustrated. We used Eq. (2.1) to get similarity scores between users and Eq. (2.3) to get ratings.

	I1	I2	I3	I4	I5	S(i,j)
U1	**5**	**4**	**?** ← — —		**3** —	— — —
U2	**1**		**3**	**2**		**-0.5**
U3	**4**		**4**	**3**		**0.29**
U4	**4**	**4**	**5**		**3**	**0.5**
U5	**2**	**1**			**1**	**0.87**

$r_{1,3} = 4.6$

Figure 2.7 Collaborative Filtering process [74].

Collaborative Filtering is based on the assumption that a consensuses of people in the past will agree on in the future. Therefore, users will like similar kinds of items that they liked in the past. The advantage of Collaborative Filtering among other recommender systems is that it can recommend a different and unknown item from what the user already knows. This recommendation represents a surprise and is attractive to users.

Nevertheless, Collaborative Filtering often suffers from three common problems: sparsity, cold start, and scalability. These three challenges of the Collaborative Filtering are described below. We try to reduce the impact of these challenges in our approach. There are also many recommender systems proposed [35], [36], [37] to address these problems.

2.4.1.1 Scalability

In many of environments, we need much time to find a similar neighbor when we use Collaborative Filtering. Because, data sets contain millions of users and items. Further, the number of users and items are increasing, so it becomes computationally difficult to find similar neighbors. This increasing in the number of users and items is called scalability problem.

2.4.1.2 Sparsity

Mostly, users don't rate items. Even popular items that user liked or bought still unrated. Because of, increasing number of users and items with few ratings, most entries of data sets remain zero. This situation is called sparsity problem. The level of sparsity is determined by the ratio of the number of zeros to the total number of matrix.

2.4.1.3 Cold Start

We can consider the cold start problem as a special case of the sparsity problem [12]. The cold start problem happens because the user doesn't have enough rating or any rating at all. To avoid this problem, some companies offer to the consumers some of popular items to evaluate it when they login to the company's accounts at first time. Otherwise, it is difficult for recommender systems to provide an accurate recommendation to users.

2.4.2 Content-Based Recommender Systems

Content-Based Filtering approaches are based on a description of item features and user preferences in his/her profile [15], [59]. Content-Based Filtering recommends items similar to the same type of items that a user already liked. Content-Based Filtering may be defined as an algorithm of searching and comparing therefore it is similar to processes that are used in information retrieval systems, but without needing user queries.

Content-Based Filtering obtains the information from two knowledge sources: item features and its rating that is given by users. Simple approach uses average values of items that are rated. Also, there are more advanced techniques to discover what is desirable for the user, such as decision trees, Bayesian classifiers, and cluster analysis algorithms. An example, the data sets that are used in our approach about movies. So, if the user has given a preferred rating toward action movies, Content-Based Filtering will recommend more action movies to him, as shown in Figure 2.8. Content-Based Filtering, unlike Collaborative Filtering lacks the property of being able to be surprising.

Often, getting common attribute values are not easy in the different items, rather than similar items [3], [13]. In addition, a Content-Based Filtering depends on well-structured and reasonable distribution of the attribute values across items [14].

Figure 2.8 Content-Based Filtering process [74]

2.4.3 Hybrid Recommender Systems

Hybrid recommender systems are defined as a combination of various knowledge sources and different techniques together to obtain the outputs. Knowledge sources consist of user profile, community data, and item features. Hybrid recommender systems could be suitable in some cases in different application domains to get the right

recommendations to the user in a timely manner. There is one output for whatever number of techniques or recommender systems that contributed in hybrid recommender systems.

Collaborative Filtering uses user profile (user's ratings) together with community data to derive recommendations. Content-Based Filtering relies on textual descriptions of item features and user's ratings. Thus, the recommender system type that chosen determine which kind of knowledge sources will be needed. However, none of basic approaches can use all of these knowledge sources.

In Table 2.1, knowledge sources which represent the feedback of content-based and collaborative recommender systems are listed.

Table 2.1 Knowledge sources of recommender systems

Recommender system	User profile	Community data	Item features
Collaborative	Yes	Yes	No
Content-Based	Yes	No	Yes

We can clearly view in Table 2.1, Collaborative Filtering approach relies on user profile together with community data. Content-Based Filtering approach relies on user profile and item features to get the recommendations for users.

In Figure 2.9, several knowledge sources for a hybrid recommender systems are illustrated. Hybrid recommender systems can be like a black box that combines several knowledge sources with different techniques or recommender systems to get a recommendation list. The recommendation list in Figure 2.9 represents the best recommendations that will be recommended to the user.

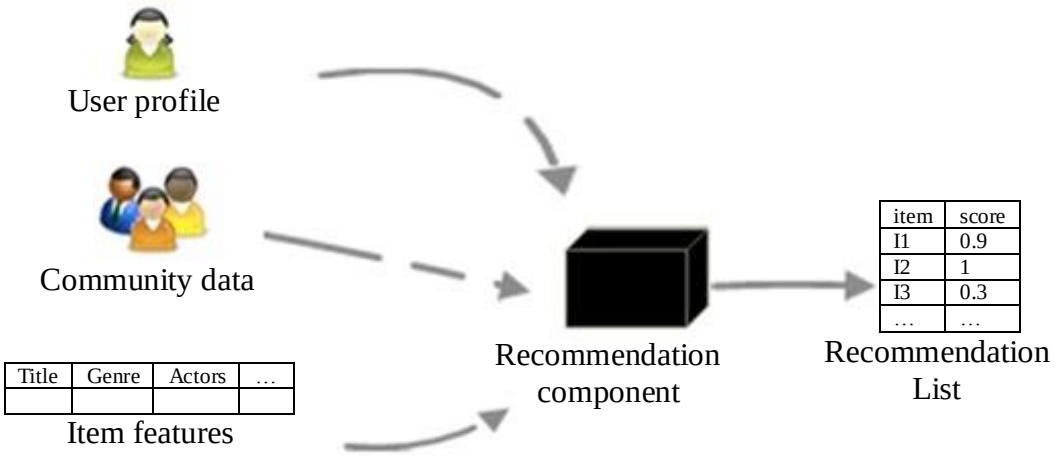


Figure 2.9 Hybrid recommender systems process [1], [75].

Hybrid recommender systems can be divided into three different major categories mentions seven hybridization techniques [1]. Below we give short summaries of these techniques. More information can be found in [1] and [16].

2.4.3.1 Monolithic Hybridization Design

Monolithic hybridization implements and combines several recommender systems in one algorithm to produce the final set of recommendations. Feature combination and feature augmentation technique belongs to this category.

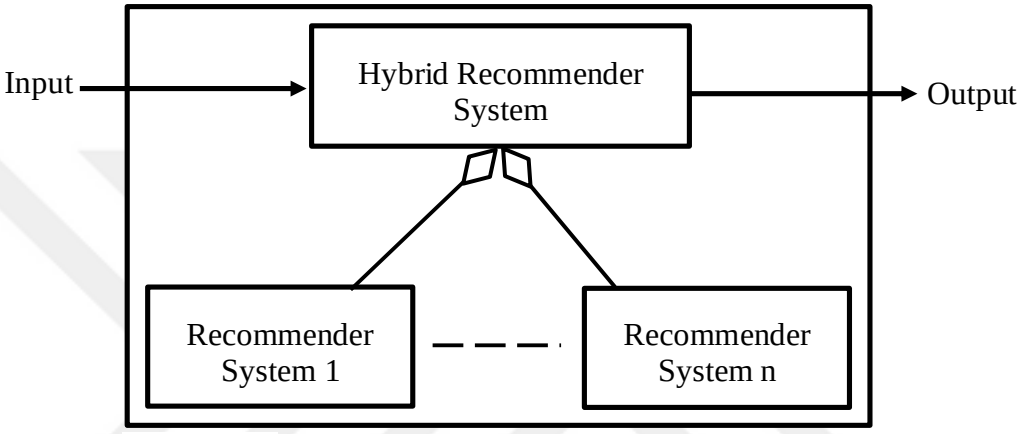


Figure 2.10 Monolithic hybridization design [1], [16], [75].

2.4.3.2 Parallel Hybridization Design

In parallel hybridization, each recommender system that participate operates independently of others and each one has its own outcomes (i.e. separate recommendation list). The outcomes of these several existing systems are combined to generate the final set of recommendations. The mixed, weighted, and switching techniques are classified as in this design.

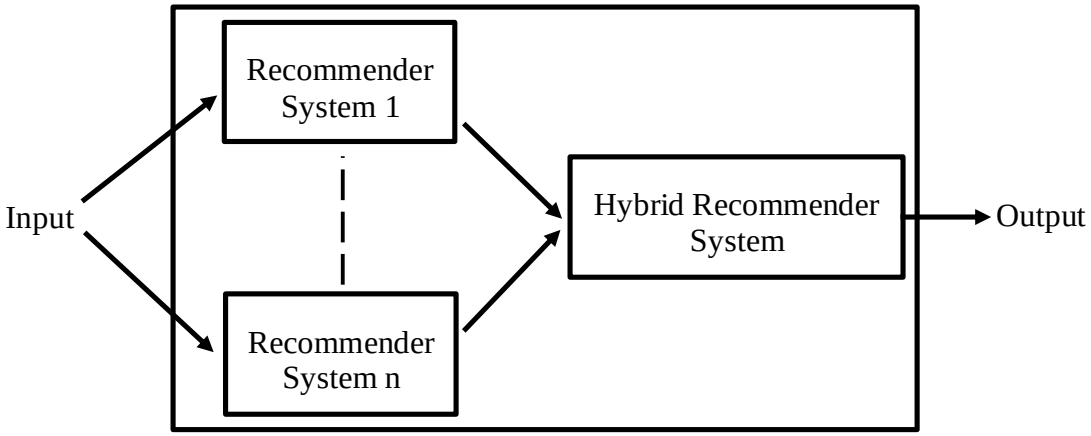


Figure 2.11 Parallel hybridization design [1], [16], [75].

2.4.3.3 Pipelined Hybridization Design

In pipelined hybridization, outputs of previous recommender system become inputs of subsequent one and the final system produces recommendations for users. So, the outputs of the first recommender system affects all the later chain of recommender systems that are in the pipeline. Optionally, subsequent recommender components may use parts of original input data, too [1]. The cascade and meta-level techniques are examples of such pipeline design.

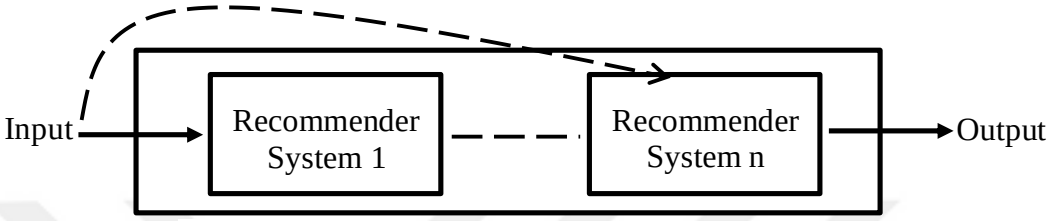


Figure 2.12 Pipelined hybridization design [1], [16], [75].

CHAPTER 3

METHODOLOGY

In this chapter, we introduce the outline concerning the meta-level technique, data sets, feature learning and computing the summary matrix that helps in speed. We divide the data sets that are used in our approach into two data sets one for training (60%) and another for testing (40%). We apply our approach on the training data set.

3.1 Meta-Level Technique

Meta-level technique is one-of the seven hybridization recommendation techniques under the pipelined hybridization design category. Meta-level technique makes the output of previous approaches as inputs of the next approach. As a result, the contributing recommender completely replaces raw data with the learned models. The resulting data are used as inputs in the calculation of the actual recommender.

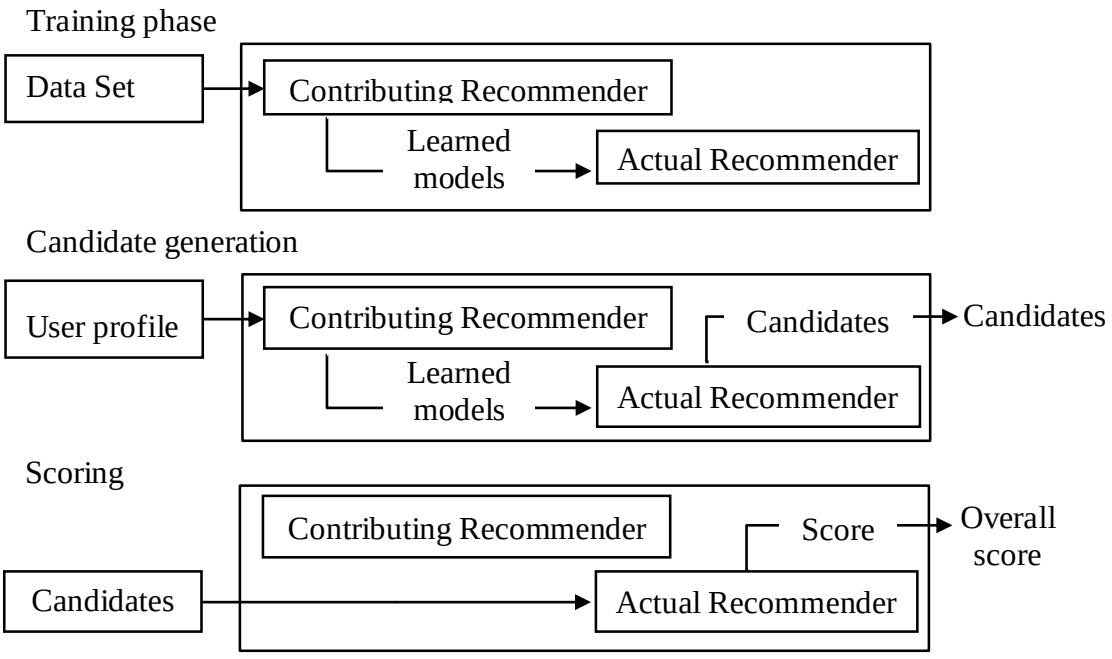


Figure 3.1 Meta-level technique [16].

3.2 Data Sets That We Used

In this part, we introduce each data set that we used in our approach. As well as, we describe some basic statistics of the training data sets. The two data sets that used in this study are available to download from the GroupLens Research website [30].

- MovieLens 1M data set: GroupLens Research collected and made available rating data sets from the MovieLens website [31]. The data set is collected over various periods of time. The rating values range between 0.5 and 5. The data set consist of around 6,040 users and 3,883 items.
- HetRec 2011 data set: The 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011) [32] released data set from Delicious, Last.fm Web 2.0, MovieLens, IMDb, and Rotten Tomatoes. This data set contains social networking, tagging, and individual information from sets of around 2,113 users. The rating values range between 0.5 and 5. The data set consist of around 2,113 users and 10,197 items.

In Table 3.1, the statistics of training data sets: HetRec 2011 and MovieLens 1M are listed. Average number of users who gave ratings for items and average number of items that rated by users can be seen from Table 3.1.

Table 3.1 Statistics of training data sets.

Statistics	HetRec 2011	MovieLens 1M
Number of users	2113	6040
Number of items	10197	3883
Number of ratings	515000	598209
Average number of ratings by users	243.73	99.04
Average number of ratings for items	50.5	154.06
Density	2.4%	2.55%

The ratings density is defined as the fraction of ratings over the total multiplies number of user and items in the matrix.

$$Density = \frac{No.\ of\ Ratings}{No.\ of\ Users \times No.\ of\ Items} \tag{3.1}$$

3.3 Feature Learning and Computing the Summary Matrix

In machine learning, feature learning, or representation learning, is a set of techniques that learns features [19], [23]. The new representation should make machine learning algorithms simpler and more flexible.

The data set in our thesis consists of two major categories: users and items (movies). Each one of the data set contains many features which include many attribute values. For example, user's category contains gender, occupation, age and Zip-code, item's category contains title, genres, actors, and year of release. Gender feature contains two attribute values: male and female. Genre feature contains many attribute values such as action, comedy, and drama.

	Year	Genre
Item1	2006	Action
Item2	2011	Drama
Item3	2002	Animation
Item4	2004	Action
Item5	2015	Drama
Item6	1999	Romance
Item7	2000	Action, Crime

	Action	Drama	Animation	Action, Crime	Romance
User1	3.5		4.5	3.7	2.5
User2	2.5	3	1.5		
User3		3.4		2.2	1.8
User4	2.1		3.6	4.8	3.3
User5	4.4	4.1	2	3	
…….					

Figure 3.2 Overview of the summary matrix.

Feature creation is a process to generate new features based on existing attribute values. For example, say, we have genre (action, comedy, crime, romance) as an input values in a data set. We can generate new feature like action, comedy, crime, and romance that may have a better relationship. This step is used to highlight the hidden relationship in the attribute values.

Feature engineering is the science of extracting more information from existing data [18]. We are not adding any new data here, but we are actually making the data we already have more useful. There are various techniques to create new features, as in [18]. The summary matrix is based on selected feature for movie data set, in our approach the genre feature is good.

An illustration of obtaining the summary matrix is given in Figure 3.2 and the techniques that we will apply on this matrix to get the recommendations. We also explain it below:

- Extract all attribute values of the selected feature.

- Extract the attribute values of the selected feature without repetition.
- Create the summary matrix with new columns based on attribute values of the selected feature.
- Fill the summary matrix with the average ratings based on attribute values of the selected feature.
- Compute similarity scores between users in the summary matrix by using Collaborative Filtering, as in Eq. (2.1).
- Get the recommended item by using Eq. (2.3).
- Get top K similar users ($K = 80$). These users will be the candidates to Collaborative Filtering approach. The similarity measure used is the Euclidean distance, is defined as:

$$d(i,j) = \sqrt{\sum_{L=1}^{y}\left(r_{L,i} - r_{L,j}\right)^2} \tag{3.2}$$

Below we explain our feature learning method. In this method we learn a "summary matrix" that has average rating values for a user on attribute values of a selected column. For this work we selected the genre column as an example.

Let $S(i)$ denote training sample item i, then $S(i)$ can be represented as:

$$S(i) = \left\{\vec{\boldsymbol{I}}(\boldsymbol{i}), \vec{\boldsymbol{V}}(\boldsymbol{i}), \vec{\boldsymbol{D}}(\boldsymbol{i})\right\} \tag{3.3}$$

Where, $\vec{\boldsymbol{I}}(\boldsymbol{i})$, $\vec{\boldsymbol{V}}(\boldsymbol{i})$ and $\vec{\boldsymbol{D}}(\boldsymbol{i})$ stand for the input vector and the two output vectors for training sample item i, respectively. $\vec{\boldsymbol{I}}(\boldsymbol{i})$ represents all item features (i), whose structure can be shown as:

$$\vec{\boldsymbol{I}}(\boldsymbol{i}) = \begin{cases} Title\ (i) \\ Year\ (i) \\ Genre\ (i) \\ Location\ (i) \\ Director\ (i) \\ Actors\ (i) \\ Country\ (i) \end{cases} \tag{3.4}$$

All entries are either textual or integers. Genre (i) is represented as the genre feature (i) that will be extract from other item features, and it is textual.

Likewise, $\vec{\boldsymbol{V}}(\boldsymbol{i})$ represents all the attribute values of the selected feature, which can be shown as:

$$\vec{V}(i) = \begin{cases} \text{Adventure, Children, Fantasy} \\ \text{Comedy, Romance} \\ \text{Comedy} \\ \text{Action, Crime, Thriller} \\ \text{Adventure, Children, Action} \\ \text{Comedy} \\ \text{Adventure, Children, Action} \end{cases} \quad (3.5)$$

Likewise, $\vec{D}(i)$ represents unrepeated attribute values of the selected feature that will be the new columns of the summary matrix, which can be shown as:

$$\vec{D}(i) = \begin{cases} \text{Adventure, Children, Fantasy} \\ \text{Comedy, Romance} \\ \text{Comedy} \\ \text{Action, Crime, Thriller} \\ \text{Adventure, Children, Action} \end{cases} \quad (3.6)$$

We can define $W(i,j)$ as the average ratings based on Eq. (3.4) and TF (explained next paragraph), for each user's items in the summary matrix. i represents the users, j represents the items in the summary matrix. Then $W(i,j)$ can be obtained as:

$$W(i,j) = \frac{\sum_{\vec{v} \in \vec{D}} r_i}{TF} \quad (3.7)$$

Term Frequency (TF) denotes the number of times that the attribute values of the selected feature $\vec{D}$ appears in user's profile for rated items.

Algorithm 3.1 Building the summary matrix

1: input:
2: $\vec{v} \leftarrow < v_1, \ldots, v_L >$ // $\vec{v}$ is the vector of attribute values.
3: $\vec{d} \leftarrow < d_1, \ldots, d_K >$ // $\vec{d}$ is the vector of unrepeated attribute values.
4: for i=1:K
5: for j =1:L
6: if $v_j \in d_i$
7: $u \leftarrow u + r$ // r is the rating values of the data set.
8: $TF \leftarrow TF + 1$.
9: end
10: end
11: $W = u/TF$ // average ratings for each item of users in the summary matrix.
12: end

The summary matrix will be filled with average ratings for items that are rated by a user in the data set. The summary matrix consists of the same number of users (rows) in the data set, but new items (columns).

In Table 3.2, the statistics of the summary matrix after implementing Algorithm 3.1 are listed. The number of items in the summary matrix is reduced. Therefore, the rating density is increased, which contributes to solve the problems: scalability, sparsity and cold start.

Table 3.2 Statistics of the summary matrix.

Statistics	HetRec 2011	MovieLens 1M
Number of users	2113	6040
Number of items	788	301
Number of ratings	218293	252394
Average number of ratings by users	103.31	41.79
Average number of ratings for items	277.02	838.52
Density	13.1%	13.9%

In Figure 3.3, the amount of decrease in the items before and after implementing Algorithm 3.1 for HetRec 2011 and MovieLens 1M data set is illustrated.

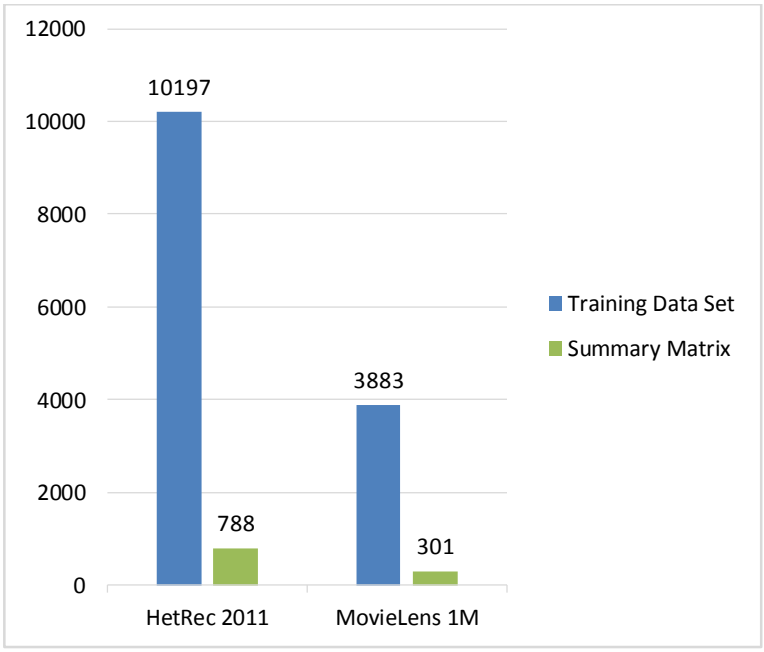


Figure 3.3 Aamount of decrease in the items.

In Figure 3.4, the rating density of HetRec 2011 and MovieLens 1M data set before and after implementing Algorithm 3.1 is illustrated.

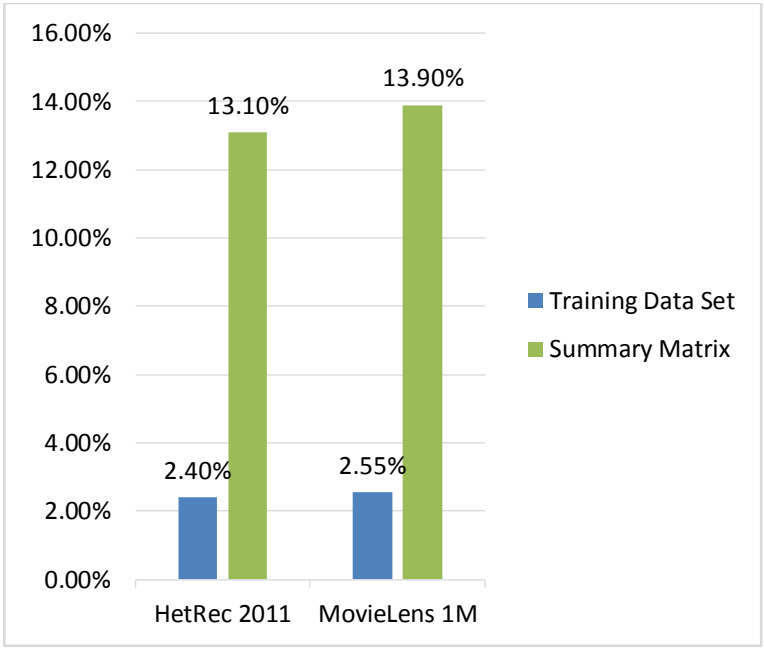


Figure 3.4 The rating density.

Today the increasing growth in the web with thousands of users who interact with thousands of items, slows down the work of recommender systems. Therefore, we need to reduce the amount of time used and the number of similar users to make software systems faster to get recommendations so our approach focuses on reducing the items in the summary matrix to get satisfactory results quickly. In addition, our approach focuses on increasing the rating density that makes recommender systems operations to discover similar users easy.

In Figure 3.5, the percentage of the ratings that are given by one user to all items in the training data sets versus the summary matrix is illustrated.

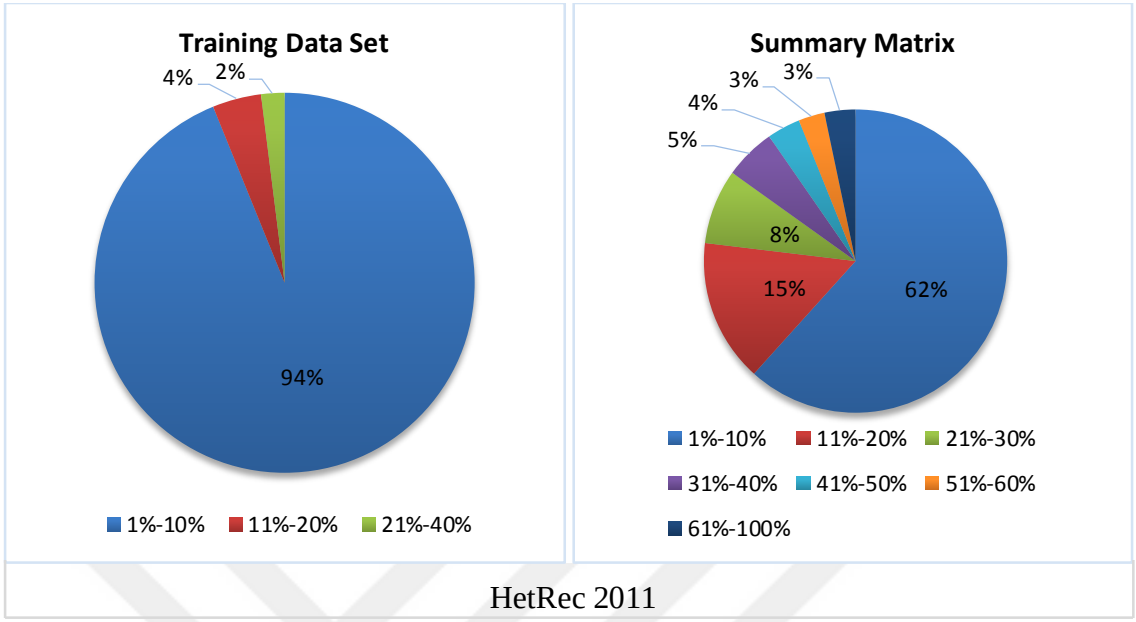
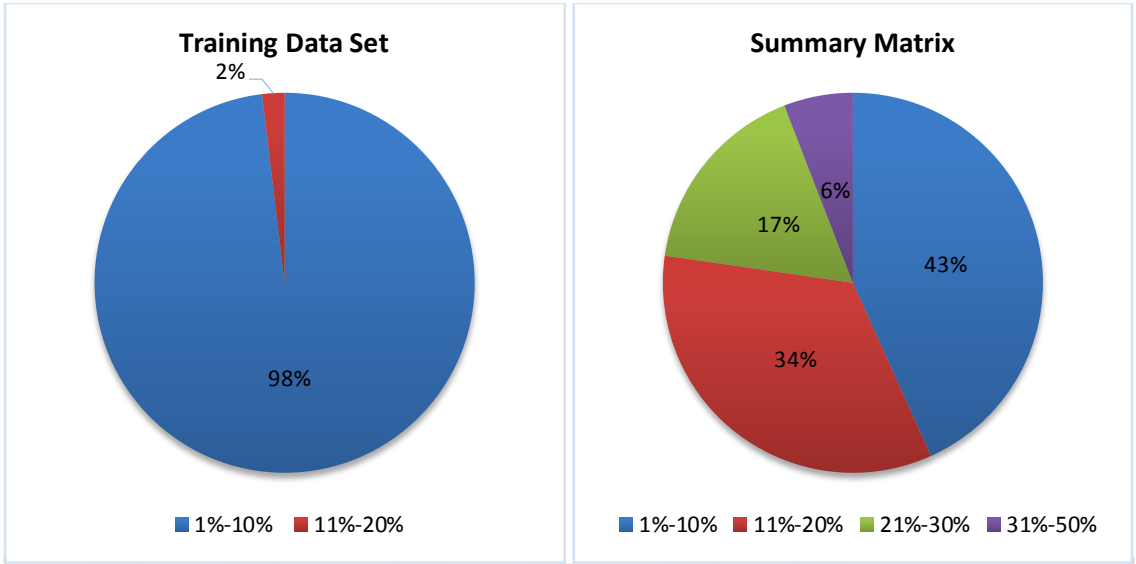


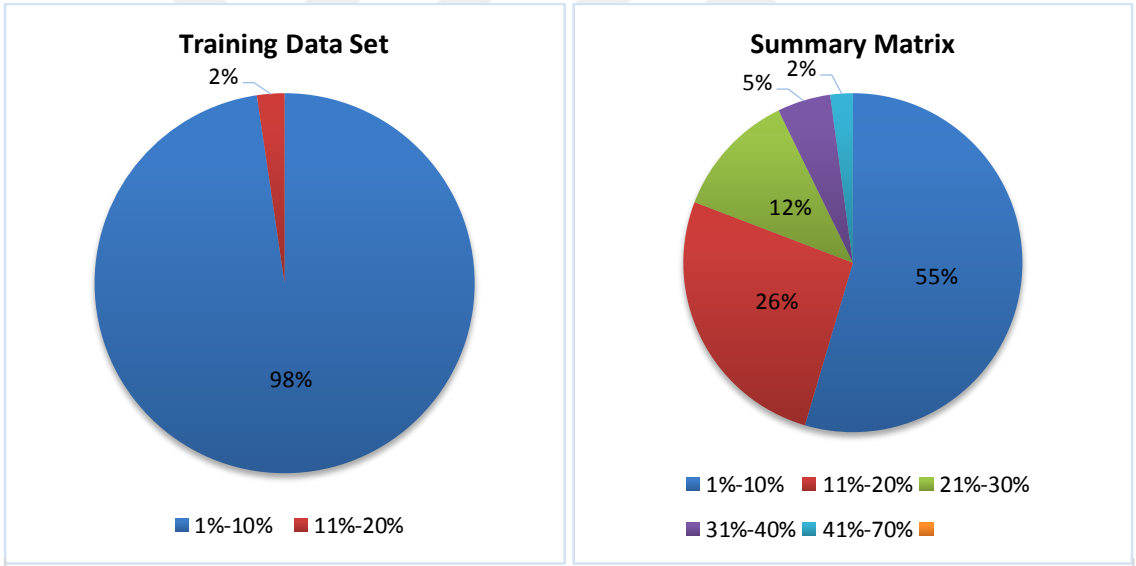
Figure 3.5 Number of users versus number of items.

Note that as seen in Figure 3.5, 94%-96% of users rated less than 10% of all items in the training data sets. Also in Figure 3.6, we notice that 98% of the items in the training data sets are rated by less than 10% of users. This percentage is very low and reduce the opportunities for getting accurate recommendations.

In Figure 3.6, the percentage of the ratings that are given by all users to one item is illustrated.



HetRec 2011



MovieLens 1M

Figure 3.6 Number of items versus number of users.

Note that as seen in Figure 3.5 and Figure 3.6, re-distribution of ratings in the summary matrix for users and items. All percentages of ratings increased over 10%. This means more opportunities for getting accurate recommendations for users.

In Figure 3.7, general schematic of techniques that we used in our approach is illustrated.

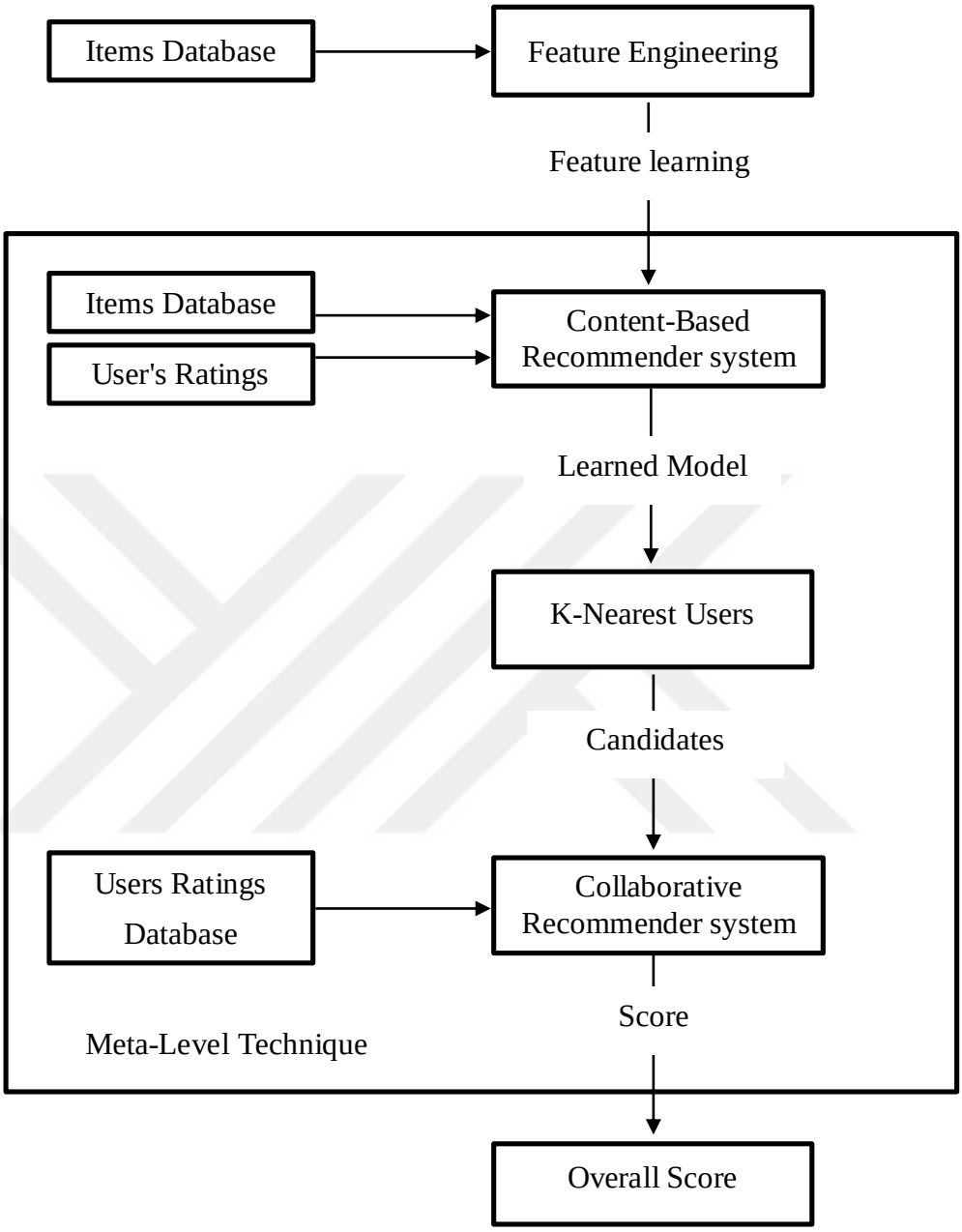


Figure 3.7 General schematic of techniques that we used.

3.4 Summary

In this chapter, the results obtained through creating the summary matrix can be summarized as follows:

- Decreasing the number of items.
- Increasing the rating density.
- Increasing the ratings of users.
- Increasing the ratings of items.

Now, we have two important questions will proof in the next chapter:

- How useful of reducing the items?
- Can the proposed approach improve the recommendation accuracy?

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter, we re-predict the ratings of a testing data set. Following this, we will review the findings of comparing two techniques of hybrid recommender systems based on the summary matrix with the Collaborative Filtering Pearson Correlation approach based on a training data set. Each technique has a different pattern, which makes it vary in the strengths and drawbacks. Therefore, each technique has characteristic results.

4.1 Overview

Recommender systems have been evaluated in many different evaluation metrics over the past several years [1], [25], [43]. Recommender systems evaluation is difficult because the evaluation results are mutable, it is based on algorithms, data sets, and evaluation metrics together. Evaluation metrics are divided into two major categories according to desired recommendations results. The first category is based on numeric value (i.e. error ratio) that represents the difference of original rate and predicted rate, and is called predictive accuracy metrics. The second category is based on relevance (i.e. separating the range of rating into two groups) that represents the relevant or irrelevant relation between original rate and predicted rate, and is called classification accuracy metrics. There is motivation to use both types of evaluation metrics in this thesis because every category follows a certain pattern for evaluation.

4.2 Data Sets and Preprocessing

The summary matrix is created by implementing Algorithm 3.1 on two data sets MovieLens 1M and HetRec 2011, as we mentioned in Chapter 3. The purpose of creating the summary matrix is to improve the performance and get accurate recommendations.

We propose two techniques of hybrid recommender systems according to the summary matrix. Each one has advantages different from the other because the first technique combines two techniques and another consists of three techniques.

HRS-1 denotes combining summary matrix and Collaborative Filtering Pearson Correlation approach.

HRS-2 denotes combining summary matrix, K-Nearest User, and Collaborative Filtering Pearson Correlation approach.

CFP denotes to Collaborative Filtering Pearson Correlation approach.

In Table 4.1, average amount of time used (in second) and average number of similar users for testing each data is listed.

Table 4.1 Average amount of time used and average number of similar users.

Data Sets	HetRec 2011			MovieLens 1M		
Techniques	CFP	HRS-1	HRS-2	CFP	HRS-1	HRS-2
Average amount of time used	0.568	**0.074**	**0.379**	0.83	**0.151**	**0.716**
Average No. of similar users	231	**195**	**52**	310	**300**	**55**

In Figure 4.1, average amount of time used of HRS-1 and HRS-2 comparing with CFP is illustrated.

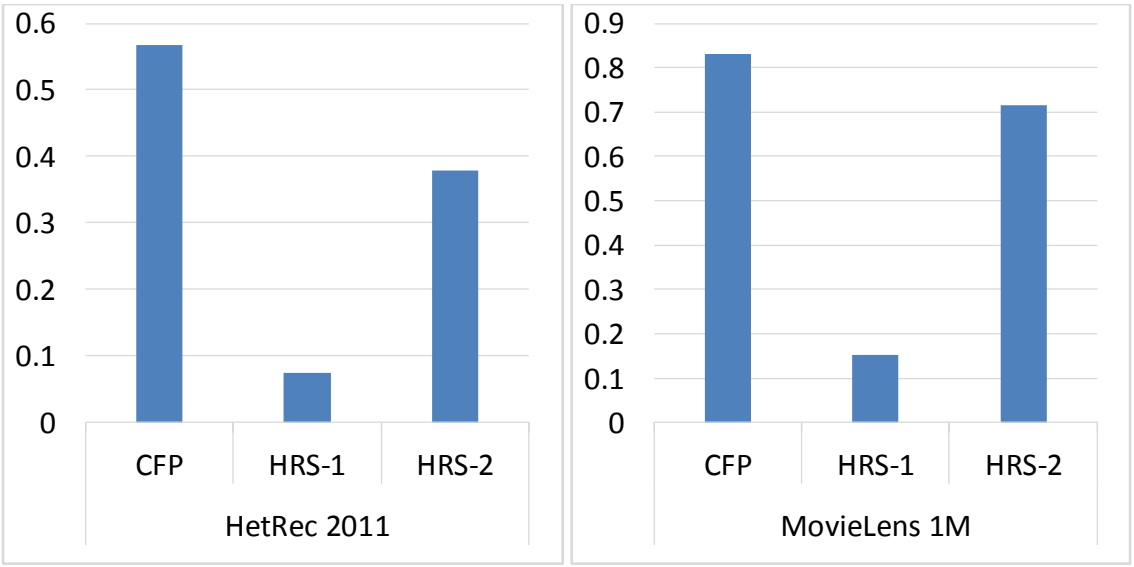


Figure 4.1 Average amount of time used.

The performance superiority of HRS-1 in time used for two reasons: it consists of two techniques are combined and limited items in the summary matrix. So, the HRS-1 technique will be faster than other techniques, can be seen from Figure 4.1.

In Figure 4.2, average number of similar users for each sample of HRS-1 and HRS-2 comparing with CFP is illustrated.

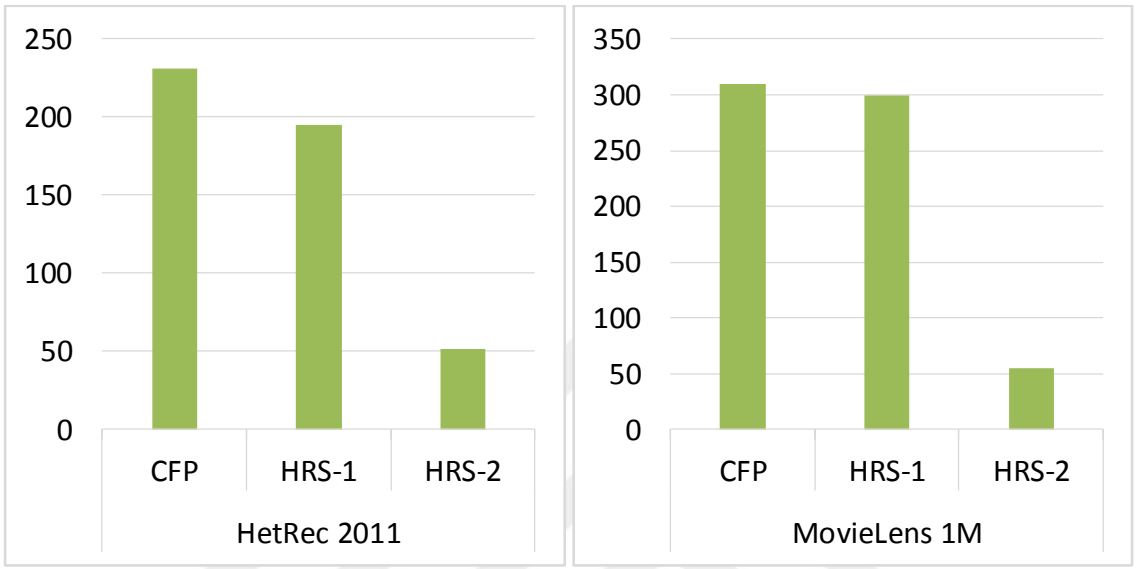


Figure 4.2 Average number of similar users.

One reason for performance superiority of HRS-2 is in average number of similar users. The HRS-2 retrieves only the top eightieths of similar users $K = 80$, which leads to a shortened search process in the entire summary matrix, can be seen from Figure 4.2.

4.3 Evaluation Metrics

We applied five evaluation metrics belonging to two categories. It would be better to choose one or more evaluation metrics to compare the accuracy of different recommender systems [25].

4.3.1 Predictive Accuracy Metrics

Predictive accuracy metrics are based on numerical differences between predicted ratings and true ratings that are given by the user to the movies. The rating is estimated by five-stars in the selected data set: HetRec 2011 and MovieLens 1M.

Recommender systems evaluation relies on how close predicted ratings are to true ratings. The recommender system is considered successful if the difference between the numerical values is small or vice-versa.

There are many evaluation metrics for evaluating the ability of recommender systems to correctly predict a specific item. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are two of the most important evaluation metrics [1]. These predictive accuracy metrics are used for recommender systems evaluation because it is easy to calculate and understand.

MAE Eq. (4.1) measures the average absolute deviation between predicted rating and true rating. RMSE Eq. (4.2) represents the sample standard deviation of the differences between predicted rating and true rating.

$$MAE = \frac{\sum_{i=1}^{T} |p_i - r_i|}{T} \quad (4.1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{T} |p_i - r_i|^2}{T}} \quad (4.2)$$

Where, p_i is the predicted ratings, r_i is the rating values, T is the total number of predictions generated for all active users.

RMSE metric was used as a condition to determine the winner in the competition of Netflix Inc. [26]. The condition was to improve the results of RMSE metric of a proposed algorithm 10% compared with the Netflix Inc. algorithm that is known as Cinematch.

In Table 4.2, the results of MAE and RMSE evaluation metrics of HRS-1 and HRS-2 comparing with CFP are listed.

Table 4.2 MAE and RMSE evaluations.

Data Sets	HetRec 2011			MovieLens 1M		
Techniques	CFP	HRS-1	HRS-2	CFP	HRS-1	HRS-2
MAE	0.67	**0.63**	**0.64**	0.77	**0.733**	**0.743**
RMSE	0.87	**0.823**	**0.825**	0.97	**0.927**	**0.93**

In Figure 4.3, performance evaluations of predictive accuracy metrics: MAE and RMSE are compared on CFP, HRS-1, and HRS-2 are illustrated.

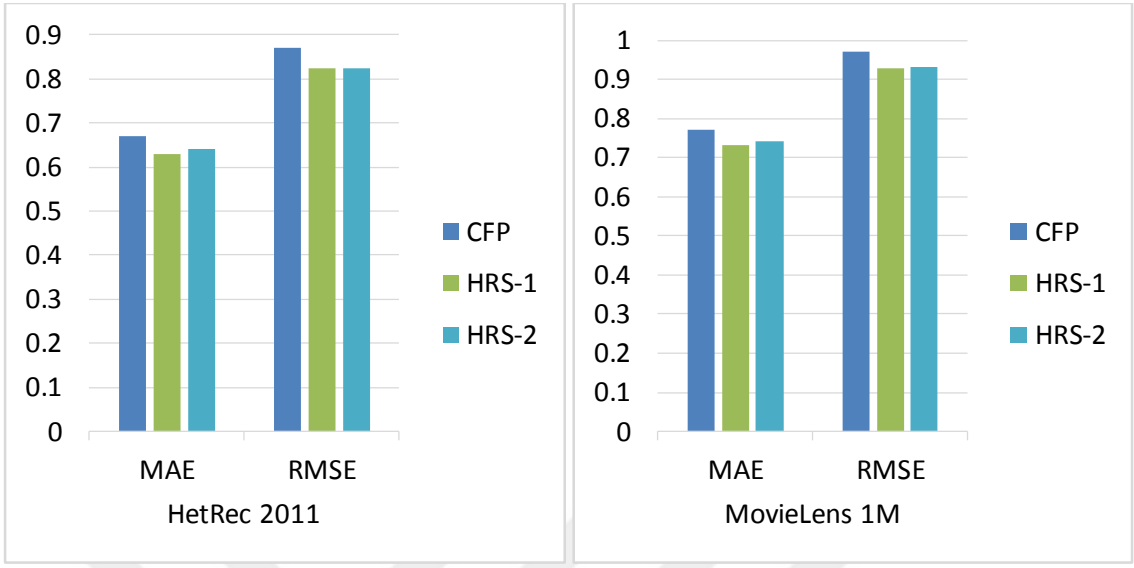


Figure 4.3 Evaluations of predictive accuracy metrics.

The performance superiority of HRS-1, can be seen from Figure 4.3.

In Figure 4.4, MAE for CFP, HRS-1, and HRS-2 with 40%,…, 90% of the training data set is illustrated.

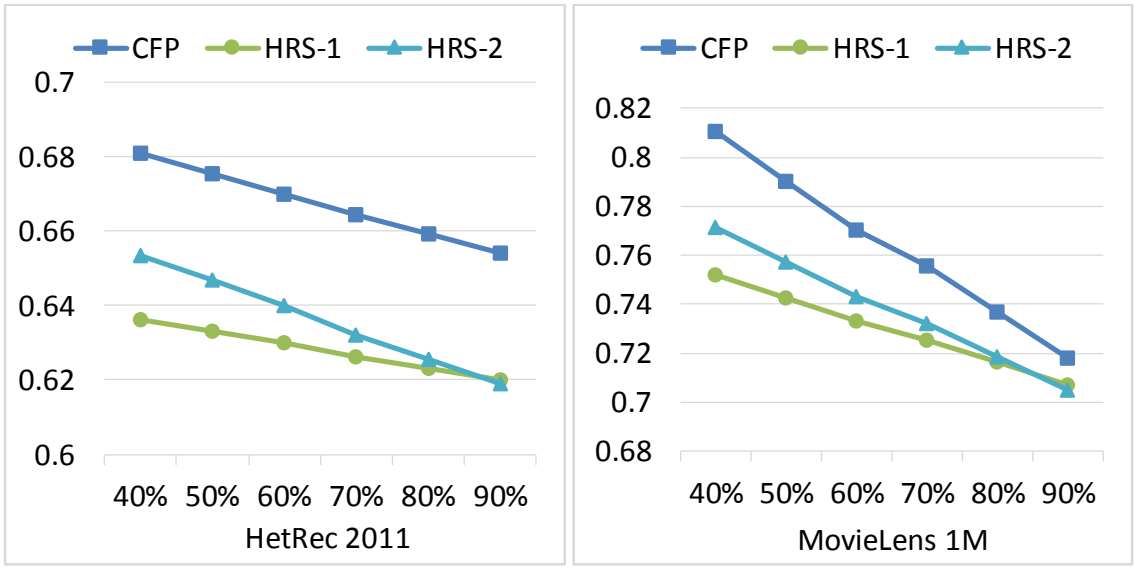


Figure 4.4 MAE for CFP, HRS-1, and HRS-2.

When we have 90% of the available ratings, the performance superiority of HRS-2, can be seen from Figure 4.4.

In Figure 4.5, RMSE for CFP, HRS-1, and HRS-2 with 40%,…, 90% of the training data set is illustrated.

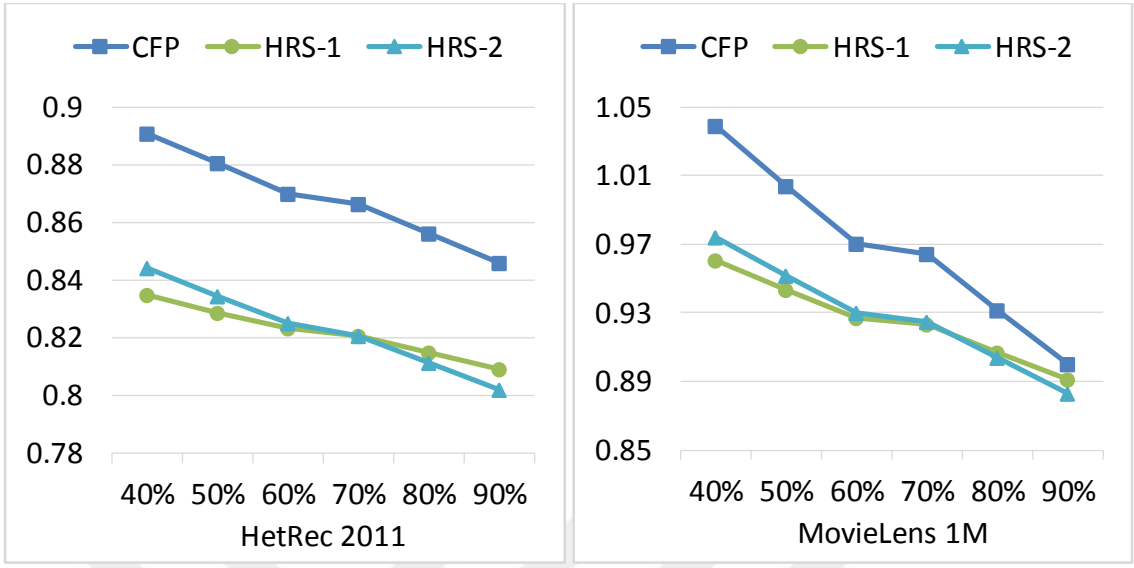


Figure 4.5 RMSE for CFP, HRS-1, and HRS-2.

When we have 80% of the available ratings, the performance superiority of HRS-2, can be seen from Figure 4.5.

4.3.2 Classification Accuracy Metrics

Classification accuracy metrics are based on relevance between predicted ratings and true ratings to determine which items are relevant (i.e. good) and which items are irrelevant (i.e. bad). We will separate the data set into two classes depending on a threshold. All ratings of 0.5 to less than 3 "irrelevant" and 3-5 "relevant".

We can classify each recommendation as [40]:

- True Positive (TP), an acceptable item recommended to user.
- True Negative (TN), an unacceptable item not recommended to user.
- False Positive (FP), an unacceptable item recommended to user.
- False Negative (FN), an acceptable item not recommended to user.

In Table 4.3, confusion matrix that accumulates the numbers of true/false and positive/negative recommendations is listed. Each column of confusion matrix represents the instances in a predicted class and each row represents the instances in an actual class (vice-versa) [21]. In this thesis, each column represents the actual class and each row represents a predicted class.

Table 4.3 Confusion matrix.

	Relevant	Irrelevant	Total
Recommended	TP	FP	TP+FP
Not Recommended	FN	TN	FN+TN
Total	TP+FN	FP+TN	TP+TN+FP+FN

Precision and recall are the most popular metrics in the information retrieval field and depend on separation between relevant "positive" and irrelevant "negative" items. Precision and recall are used in [33], [39]. F-measure allows for combining precision and recall into a single score.

Precision Eq. (4.3) is defined as the ratio of relevant items recommended to a number of items recommended. Precision represents the probability that a recommended item is relevant.

$$Precision = \frac{TP}{TP+FP} \tag{4.3}$$

Recall Eq. (4.4) is defined as the ratio of relevant items recommended to total number of relevant items. Recall represents the probability that a relevant item will recommended.

$$Recall = \frac{TP}{TP+FN} \tag{4.4}$$

F-measure Eq. (4.5) is defined as average number of precision and recall. F-measure represents the balance between precision and recall.

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision+Recall} \tag{4.5}$$

In Table 4.4, the results of precision, recall, and F-measure evaluation metrics of HRS-1 and HRS-2 comparing with CFP are listed.

Table 4.4 Precision, Recall, and F-measure evaluations.

Data Sets	HetRec 2011			MovieLens 1M		
Techniques	CFP	HRS-1	HRS-2	CFP	HRS-1	HRS-2
Precision	0.865	**0.868**	0.839	0.893	**0.91**	0.88
Recall	0.868	**0.88**	**0.89**	0.892	**0.9**	**0.91**
F-Measure	0.866	**0.876**	0.864	0.892	**0.9**	0.89

In Figure 4.6, performance evaluations of classification accuracy metrics: precision, recall, and F-measure are compared on CFP, HRS-1, and HRS-2 are illustrated.

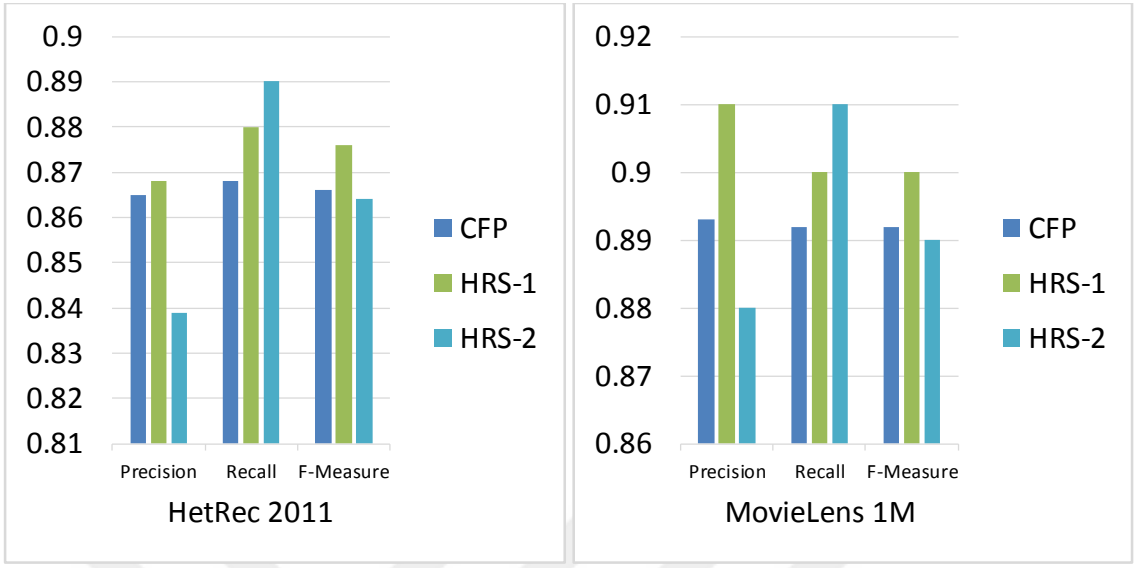


Figure 4.6 Evaluations of classification accuracy metrics.

The performance superiority of HRS-1 in precision, recall, and F-measure and HRS-2 in recall, can be seen from Figure 4.6.

In Figure 4.7, precision for CFP, HRS-1, and HRS-2 with 40%,…, 90% of the training data set is illustrated.

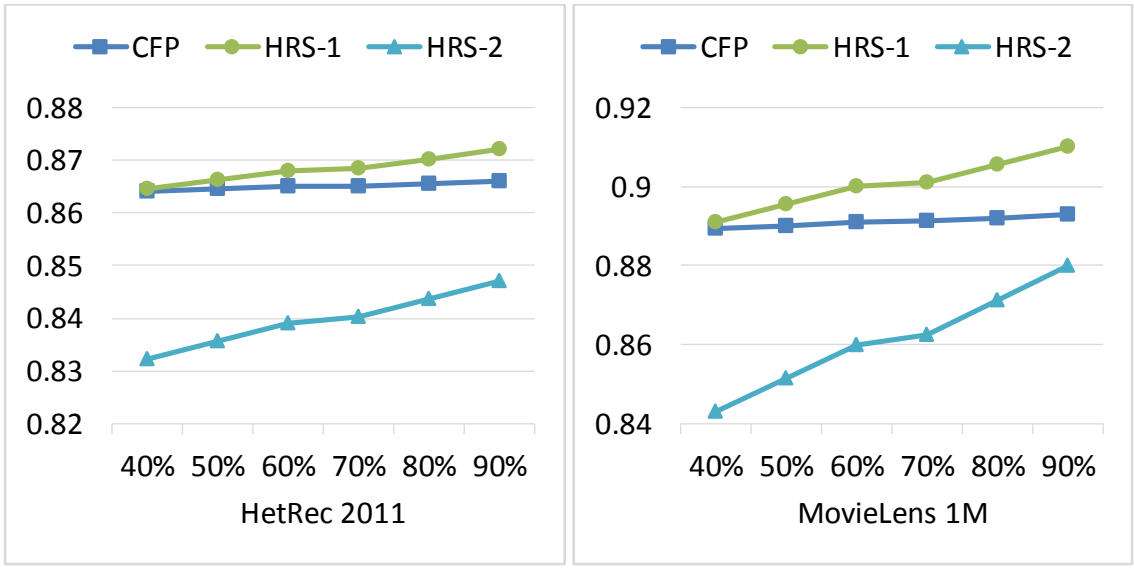


Figure 4.7 Precision for CFP, HRS-1, and HRS-2.

The performance superiority of HRS-1 with 40%,…, 90% of the available ratings, can be seen from Figure 4.7.

In Figure 4.8, recall for CFP, HRS-1, and HRS-2 with 40%,…, 90% of the training data set is illustrated.

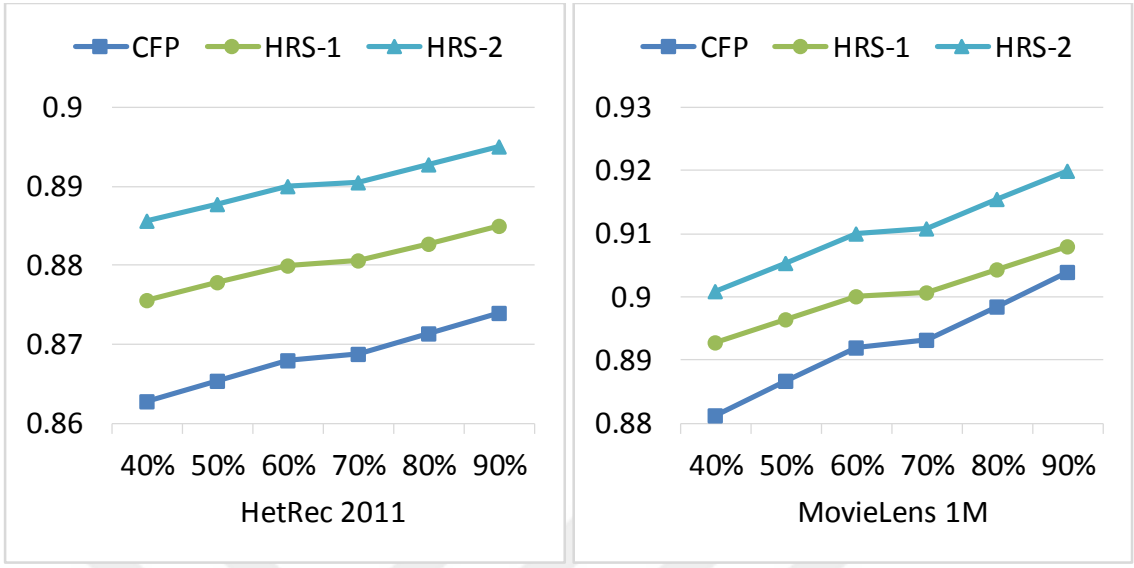


Figure 4.8 Recall for CFP, HRS-1, and HRS-2.

The performance superiority of HRS-1 and HRS-2 with 40%,…, 90% of the available ratings, can be seen from Figure 4.8.

In Figure 4.9, F-measure for CFP, HRS-1, and HRS-2 with 40%,…, 90% of the training data set is illustrated.

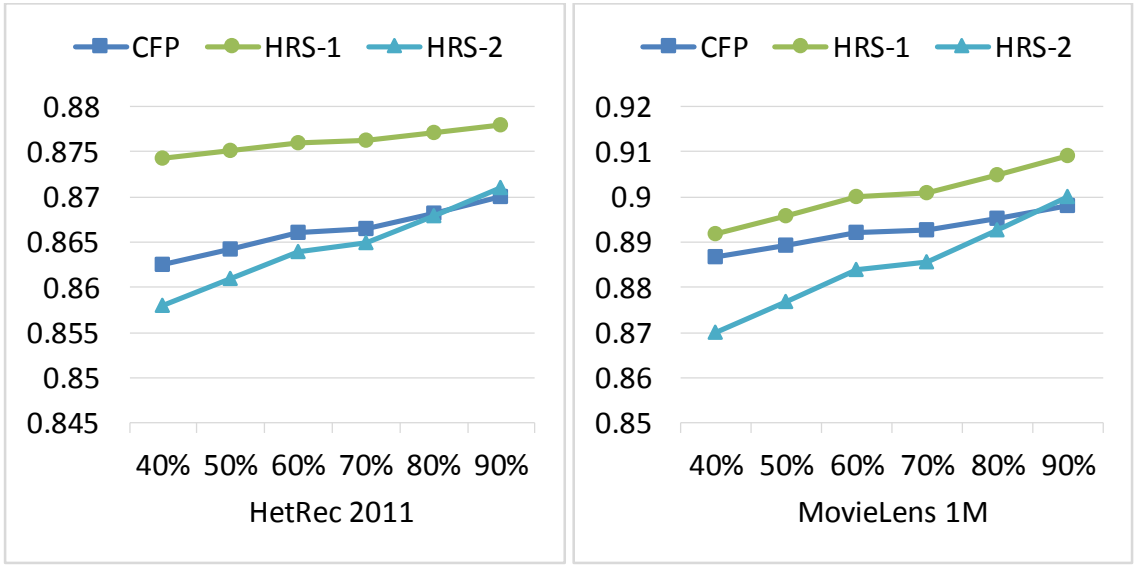


Figure 4.9 F-measure for CFP, HRS-1, and HRS-2.

The performance superiority of HRS-1 with 40%,…, 90% of the available ratings, can be seen from Figure 4.9.

In Table 4.5, explain more details about the advantage of reducing the items through reducing the amount of time used for testing one sample (in second) are listed.

There are three steps to get the recommendation list: preparing the matrix, calculating the similarity scores, and listing the recommendations. Each of these steps calculates based on a different equation as we mentioned in Chapter 2. Table 4.5, explains the comparison between CFP, HRS-1, and HRS-2.

Table 4.5 Time used for testing one sample.

Data Sets	HetRec 2011			MovieLens 1M		
Techniques	CFP	HRS-1	HRS-2	CFP	HRS-1	HRS-2
Preparing the matrix	27.31	**2.034**	**2.034**	31.858	**2.195**	**2.195**
Calculate the similarity scores	1.95	**0.151**	**0.231**	2.333	**0.243**	**0.405**
Listed the recommendations	0.015	**0.014**	**0.012**	0.03	**0.012**	**0.01**

In Table 4.6, a comparison of Confidence Weighted Online Collaborative Filtering (CWOCF) approach [55] depends on the same data set with our approach depends on the summary matrix is listed.

Table 4.6 Comparison according to CWOCF approach.

Data Sets	predictive accuracy metrics	Techniques		
		CWOCF	HRS-1	HRS-2
HetRec 2011	MAE	0.6499	**0.63**	**0.64**
	RMSE	0.8473	**0.823**	**0.825**
MovieLens 1M	MAE	0.7609	**0.733**	**0.743**
	RMSE	0.9580	**0.927**	**0.93**

Note that as seen in Table 4.6 and Figure 4.10, our approach excelled in all evaluations of predictive accuracy metrics.

In Figure 4.10, a comparison of Up and Up-q approaches [77] depends on MovieLens 1M data set with our approach depends on the summary matrix is illustrated.

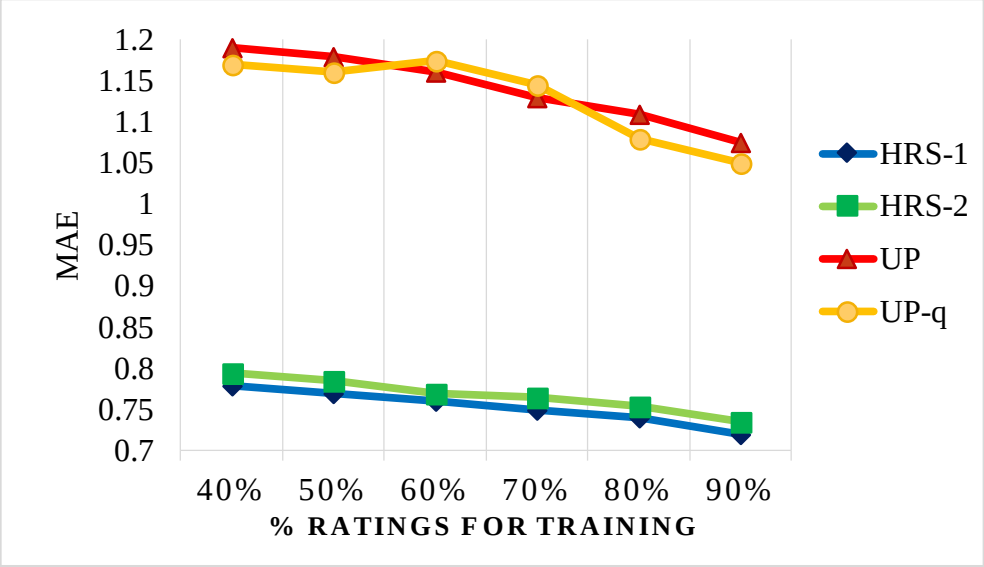


Figure 4.10 Comparison of Up and Up-q approaches with our approach.

4.4 Summary

According to the results obtained in this chapter, we proved that:

- Reducing the items was useful in speed.
- The proposed approach improves the recommendation accuracy.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this thesis, we propose to create a summary matrix that incorporates limited items to alleviate the impact of scalability, sparsity and cold start problems in recommender systems.

The proposed approach increases the rating density, which contributes to solving the aforementioned problems. We use the summary matrix in two hybrid recommender systems and evaluate the results. The results show our summary matrix was helpful in speed, increased the rating density, and got better recommendations.

Table 5.1 Percentage of improvement for all results with respect to CFP.

Data Sets	HetRec 2011		MovieLens 1M	
Techniques	HRS-1	HRS-2	HRS-1	HRS-2
Average amount of time used	87%	33.3%	82%	14%
Average No. of similar users	15.7%	77.5%	3.2%	82.25%
MAE	6.3%	4.9%	4.84%	3.68%
RMSE	5.2%	4.9%	4.1%	3.58%
Precision	0.5%	-	1.3%	-
Recall	1.6%	2.55%	0.8%	1.6%
F-Measure	1.1%	-	1.1%	-

5.2 Future Work

This work suggests several interesting directions for future work. We calculated the likeness between users based on user-user similarity. Item-item similarity may also be tried.

Additionally, we aspire to develop this work to apply it on diverse data sets such as music, books, jokes, and Twitter followers. We would like to conduct a study at a larger scale which would involve feature selection and feature creation.

REFERENCES

[1] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich, (2011). Recommender Systems: an Introduction, Cambridge University Press, New York, NY, USA.

[2] Andrey Feuerverger, Yu He, and Shashi Khatri, (2012). "Statistical Significance of the Netflix Challenge," Institute of Mathematical Statistics, 27(2): 202-231.

[3] Mohammad Amir Sharif and Vijay V. Raghavan, (2014). "A Large-Scale, Hybrid Approach for Recommending Pages Based on Previous User Click Pattern and Content," Proceedings of the 21st International Symposium (ISMIS 2014), 25-27 June 2014, Roskilde, Denmark, 103-112.

[4] Deepak Agarwal, Bee-Chung Chen, Pradheep Elango, and Raghu Ramakrishnan, (2013). "Content Recommendation on Web Portals," Communications of the ACM, 56(6): 92-101.

[5] Robert M. Bell and Yehuda Koren, (2007). "Lessons from the Netflix Prize Challenge," ACM SIGKDD Explorations Newsletter, 9(2): 75-79.

[6] Abhinandan Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram, (2007). "Google News Personalization: Scalable Online Collaborative Filtering," Proceedings of the 16th International Conference on World Wide Web (WWW2007), 8-12 May 2007, Alberta, Canada, 271-280.

[7] Greg Linden, Brent Smith, and Jeremy York, (2003). "Amazon. Com Recommendations: Item-to-Item Collaborative Filtering," Internet Computing, IEEE, 7(1): 76-80.

[8] Alexander Felfernig, Klaus Isak, Kalman Szabo, and Peter Zachar, (2007). "The VITA Financial Services Sales Support Environment," Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence (IAAI2007), 22-26 July 2007, Vancouver, British Columbia, Canada, 1692-1699.

[9] Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Bosagh Zadeh, (2013). "WTF:The Who-to-Follow System at Twitter," Proceedings of the 22nd International Conference on World Wide Web (WWW2013), 13-17 May 2013, Rio de Janeiro, Brazil, 1596.

[10] Francesco Ricci, Lior Rokach, and Bracha Shapira, (2011). Recommender Systems Handbook, Artificial Intelligence, Springer-Verlag New York, NY, USA.

[11] Lev Grossman, “Facebook, Pandora Lead Rise of Recommendation Engines,” http://content.time.com/time/magazine/article/0,9171,1992403,00.html, 27 May 2010.

[12] Zan Huang, Hsinchun Chen, and Daniel Zeng, (2004). “Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering,” ACM Transactions on Information Systems, 22(1): 116-142.

[13] Gediminas Adomavicius and Alexander Tuzhilin, (2005). “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, 11(6): 734-749.

[14] Delgado Joaquin, Ishii Naohiro, and Ura Tomoki, (1998). “Content-based Collaborative Information Filtering: Actively Learning to Classify and Recommend Documents,” Proceedings of the Second International Workshop on Cooperative Information Agents II, Learning, Mobility and Electronic Commerce for Information Discovery on the Internet, 4-7 July 1998, Paris, France, 206-215.

[15] Michael J. Pazzani and Daniel Billsus, (2007). “Content-Based Recommendation Systems,” The Adaptive Web Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 4321: 325-341.

[16] Robin Burke, (2007). “Hybrid Web Recommender Systems,” The Adaptive Web Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 4321: 377-408.

[17] Jason Brownlee, “Discover Feature Engineering, How to Engineer Features and How to Get Good at It,” http://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/, 26 September 2014.

[18] Sunil Ray, “Feature Engineering: How to Transform Variables and Create New Ones?,” https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/, 12 March 2013.

[19] Zdenek Zabokrtsky, “Feature Engineering in Machine Learning,” https://ufal.mff.cuni.cz/~zabokrtsky/courses/npfl104/html/feature_engineering.pdf, 25 March 2015.

[20] Richard D. Lawrence, George S. Almási, Vladimir Kotlyar, Marisa S. Viveros, and Sastry S. Duri, (2001). “Personalization of Supermarket Product Recommendations,” Data Mining and Knowledge Discovery, 5(1): 11-32.

[21] David M. W. Powers, (2011). “Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation,” Journal of Machine Learning Technologies, 2(1): 37-63.

[22] Paula Cristina Vaz, David Martins de Matos, Bruno Martins, and PavelCalado, (2012). “Improving an Hybrid Literary Book Recommendation System through Author Ranking,” Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL2012), 10-14 June 2012,Washington, DC, USA, 387-388.

[23] Yoshua Bengio, Aaron Courville, and Pascal Vincent, (2013). “Representation Learning: A Review and New Perspectives,” IEEE Trans. PAMI, special issue Learning Deep Architectures 35: 1798-1828.

[24] Navid Razmjooy, Bibi Somayeh Mousavi, and Fazlollah Soleymani, (2013). “A Hybrid Neural Network Imperialist Competitive Algorithm for Skin Color Segmentation,” Mathematical and Computer Modelling, 57(3-4): 848-856.

[25] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl, (2004). “Evaluating Collaborative Filtering Recommender Systems,” ACM Transactions on Information Systems (TOIS), 22(1): 5-53.

[26] Netflix Inc., American multinational entertainment company, http://www.netflix.com, 29 August 1997.

[27] Amazon.com, Inc., American electronic commerce and cloud computing company, http://www.amazon.com, 5 July 1994.

[28] Last.fm, music website, the United Kingdom, http://www.Last.fm, 20 March 2002.

[29] linkedin, business-oriented social networking service, the United States, https://www.linkedin.com, 14 December 2002.

[30] GroupLens Research, human–computer interaction research lab, the Department of Computer Science and Engineering at the University of Minnesota, http://www.grouplens.org, 1992.

[31] MovieLens, GroupLens Research, http://www.movielens.org, May 1996.

[32] HetRec workshop, Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011), http://ir.ii.uam.es/hetrec2011/, 23-27 October 2011, Chicago, IL, USA.

[33] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl, (2000). “Application of Dimensionality Reduction in Recommender System -- A Case Study,” Proceedings of Web Mining for E-Commerce -- Challenges and Opportunities (WebKDD2000), 20 August 2000, Boston, MA, USA.

[34] Ryan Baker, “Big Data: Week 3 Video 3 - Feature Engineering,” https://www.youtube.com/watch?v=drUToKxEAUA , 17 March 2014.

[35] Rong Hu and Yansheng Lu, (2006). “A Hybrid User and Item-based Collaborative Filtering with Smoothing on Sparse Data,” Proceedings of the 16th International Conference on Artificial Reality and Telexistence (ICAT2006), 29 November - 1 December 2006, Hangzhou, China,184-189.

[36] SongJie Gong, HongWu Ye, and XiaoYan Shi, (2008). “A Collaborative Recommender Combining Item Rating Similarity and Item Attribute Similarity,” Proceedings of the International Seminar on Business and Information Management (ISBIM2008), 19 December 2008, Wuhan, Hubei, China, 58-60.

[37] Sutheera Puntheeranurak and Thanut Chaiwitooanukool, (2011). “An Item-based Collaborative Filtering Method using Item-based Hybrid Similarity,” Proceedings of the IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS), 15-17 July 2011, Beijing, China, 469-472.

[38] Siavash Ghodsi Moghaddam and Ali Selamat, (2011). "A scalable collaborative recommender algorithm based on user density-based clustering," Proceedings of 3rd International Conference on Data Mining and Intelligent Information Technology Applications (ICMIA), 24-26 October 2011, The Westin Resort Coloane, Macao, 246-249.

[39] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John T. Riedl, (2000). "Analysis of Recommendation Algorithms for E-commerce," Proceedings of the 2nd ACM Conference on Electronic Commerce (EC2000), 17-20 October 2000, Minneapolis, MN, USA, 285-295.

[40] Paolo Cremonesi, Roberto Turrin, Eugenio Lentini, and Matteo Matteucci, (2008). "An Evaluation Methodology for Recommender Systems," Proceedings of the 4th International Conference on Automated Solutions for Cross Media Content and Multi-channel Distribution (AXMEDIS2008), 17-19 November 2008, Florence, Italy, 224-231.

[41] Ben Schafer J., (2008). The Application of Data-Mining to Recommender Systems, Encyclopedia of Data Warehousing and Mining, 2nd edition, 45-50.

[42] Jinhua Sun and Yanqi Xie, (2009). "A Web Data Mining Framework for E-Commerce Recommender Systems," Proceedings of the International Conference on Computational Intelligence and Software Engineering (CISE2009), 11-13 December 2009, Wuhan, China.

[43] Mehmed Kantardzic, (2011). Data Mining Concepts, Models, Methods, and Algorithms, John Wiley & Sons, Inc. New York, NY, USA.

[44] Jiawei Han and Micheline Kamber, (2006). Data Mining: Concepts and Techniques, the Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

[45] Soumen Chakrabarti, Martin Ester, Usama Fayyad, Johannes Gehrke, Jiawei Han, Shinichi Morishita, Gregory Piatetsky-Shapiro, and Wei Wang, (2006). "Data Mining Curriculum: A Proposal," ACM SIGKDD, 1-10.

[46] Michael Goebel and Le Gruenwald, (1999). "A Survey of Data Mining and Knowledge Discovery Software Tools," SIGKDD Explorations, 1(1): 20-33.

[47] David L. Olson and Yong Shi, (2006). Introduction to Business Data Mining, Boston: McGraw-Hill/Irwin, New York, NY, USA.

[48] Chandrika Kamath, (2009). Scientific Data Mining A Practical Perspective, Lawrence Livermore National Laboratory, California, USA.

[49] Ralf Mikut and Markus Reischl, (2011). "Data Mining Tools," Data Mining and Knowledge Discovery, 1(5): 431-443.

[50] Karl Rexer, Heather N. Allen, and Paul Gearan, "Understanding Data Miners," http://analytics-magazine.org/understanding-data-miners/, June 2011.

[51] James Kobielus, "The Forrester Wave ™: Predictive Analytics and Data Mining Solutions," ftp://ftp.software.ibm.com/software/kr/data/pdf/forpred.pdf, 4 February 2010.

[52] Gareth Herschel, "Magic Quadrant for Customer Data-Mining Applications," http://www.spss.com.hk/pdfs/gartner_magic_quadrant.pdf, 1 July 2008.

[53] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan, (2010). "Collaborative Filtering Recommender Systems," Human–Computer Interaction, 4(2): 81-173.

[54] Dominique Haughton, Joel Deichmann, Abdolreza Eshghi, Selin Sayek, Nicholas Teebagy, and Heikki Topi, (2006). "A Review of Software Packages for Data Mining," The American Statistician, 57(4): 290-309.

[55] Jing Lu, Steven Hoi, Jialei Wang and Peilin Zhao, (2013). "Second Order Online Collaborative Filtering," Proceedings of 5th Asian Conference on Machine Learning (ACML2013), 13-15 November 2013, Canberra, Australia, 29: 325-340.

[56] Leon Bottou, "Feature Engineering," http://www.cs.princeton.edu/courses/archive/spring10/cos424/slides/18-feat.pdf, 22 April 2010.

[57] David L. Olson and Dursun Delen, (2008). Advanced Data Mining Techniques, Business Information Systems, Springer-Verlag Berlin Heidelberg, 9-35.

[58] Charu C. Aggarwal, (2015). Data Mining: The Textbook, Database Management & Information Retrieval, Springer International Publishing.

[59] Charu C. Aggarwal, (2016). Recommender Systems: The Textbook, Database Management & Information Retrieval, Springer International Publishing.

[60] Bo Xiao and Izak Benbasat, (2007). "E-commerce product recommendation agents: use, characteristics, and impact," Society for Information Management and The Management Information Systems Research Center Minneapolis, MN, USA, 31(1): 137-209.

[61] Jorge Morais A., Eugénio Oliveira, and Alípio Mário Jorge, (2012). "A Multi-Agent Recommender System," Advances in Intelligent and Soft Computing, Springer Berlin Heidelberg, 151: 281-288.

[62] Mahmood A. Mahmood, Nashwa El-Bendary, Jan Platoš, Aboul Ella Hassanien, and Hesham A. Hefny, (2014). "An Intelligent Multi-agent Recommender System," Advances in Intelligent and Soft Computing, Springer International Publishing, 237: 201-213.

[63] Bruno Veloso, Benedıta Malheıro, and Juan Carlos Burguıllo, (2015). "A Multi–Agent Brokerage Platform for Media Content Recommendation," International Journal of Applied Mathematics and Computer Science, 25(3): 513-527.

[64] David J. Hand, Padhraic Smyth, and Heikki Mannila, (2001). Principles of Data Mining, MIT Press Cambridge, MA, USA.

[65] Xindong Wu, Vipin Kumar, Ross Quinlan J., Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg, (2008). "Top 10 Algorithms in Data Mining," Knowledge and Information Systems, 14(1): 1-37.

[66] Sumathi, S. and Sivanandam, S. N., (2006). Introduction to Data Mining and its Applications, Database Management & Information Retrieval, Springer International Publishing.

[67] Oded Maimon and Lior Rokach, (2010). Data Mining and Knowledge Discovery Handbook, Database Management & Information Retrieval, Springer International Publishing.

[68] Martin Hilbert, and Priscila López, (2011). "The World's Technological Capacity to Store, Communicate, and Compute Information," Science, 332(6025): 60-65.

[69] Cisco, "The Zettabyte Era: Trends and Analysis," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html, 2 June 2016.

[70] José María Cavanillas, Edward Curry, and Wolfgang Wahlster, (2016). New Horizons for a Data-Driven Economy, Springer International Publishing, 143-165.

[71] Viktor Mayer-Schonberger and Kenneth Cukier, (2013). Big Data: A Revolution That Will Transform How We Live, Work, and Think, Houghton Mifflin Harcourt Publishing Company, New York, NY, USA.

[72] Martin Hilbert, "What is Big Data?," https://www.youtube.com/watch?v=XRVIh1h47sA&index=51&list=PLtjBSCvWCU3rNm46D3R85efM0hrzjuAIg, 12 August 2015.

[73] International Telecommunication Union (ITU), "Key ICT Indicators for Developed and Developing Countries and the World (Totals and Penetration Rates)," http://www.itu.int/en/ITU-D/Statistics/Documents/statistics/2016/ITU_Key_2005-2016_ICT_data.xls, June 2016.

[74] Klaus Berberich and Dhruv Gupta, "Advanced Topics in Information Retrieval – Recommender Systems," http://resources.mpi-inf.mpg.de/d5/teaching/ws14_15/atir/slides/2014-atir-ch02-recommender-systems.pdf, 10 November 2014.

[75] Dietmar Jannach and Gerhard Friedrich, (2013). "Tutorial: Recommender Systems," Proceedings of the 23rd. International Joint Conference on Artificial Intelligence (IJCAI-13), 3-9 August 2013, Beijing, China, 1-128.

[76] Yibo Chen, Chanle Wu, Ming Xie, and Xiaojun Guo, (2011). "Solving the Sparsity Problem in Recommender Systems Using Association Retrieval," Journal of Computers, 6(9): 1896-1902.

[77] Iván Cantador, Alejandro Bellogín, and Pablo Castells, (2008). "A Multilayer Ontology-based Hybrid Recommendation Model," AI Communications, 21(2-3): 203-210.

APPENDIX-A

DISTRIBUTION OF RATINGS

In Table A.1, the percentage of rating that are given by one user to all items in the training data sets versus the summary matrix is listed. The percentage column of Table A.1 shows the rating density. As, it is clearly been increase rating density in the summary matrix contribute to improve the recommendation accuracy.

Table A.1 Number of users versus number of items.

	HetRec 2011				MovieLens 1M			
	Training Data Set		Summary Matrix		Training Data Set		Summary Matrix	
Ratings	Items	%	Items	%	Items	%	Items	%
1%-10%	9570	93.85	486	61.67	3385	87.17	176	58.47
11%-20%	423	4.148	120	15.22	125	3.21	72	23.92
21%-30%	132	1.29	63	7.99	16	0.41	23	7.64
31%-40%	61	0.59	43	5.45	357	0	11	3.65
41%-50%	11	0.12	28	3.55			9	2.99
51%-60%			22	2.79			6	1.99
61%-70%			10	1.26			1	0.33
71%-80%			9	1.14			1	0.33
81%-90%			5	0.63			2	0.66
91%-100%			2	0.25				
Total	10197		788		3883		301	

In Table A.2, the percentage of rating that are given by all users to one item is listed.

Table A.2 Number of items versus number of users.

	HetRec 2011				MovieLens 1M			
	Training Data Set		Summary Matrix		Training Data Set		Summary Matrix	
Ratings	Users	%	Users	%	Users	%	Users	%
1%-10%	2073	98.1	914	43.25	5894	97.58	3279	54.29
11%-20%	39	1.84	719	34.03	141	2.34	1577	26.12
21%-30%	1	0.05	356	16.85	4	0.06	719	11.9
31%-40%			111	5.25	1	0.02	305	5.05
41%-50%			13	0.62			129	2.14
51%-60%							26	0.43
61%-70%							5	0.083
71%-80%								
81%-90%								
91%-100%								
Total	2113		2113		6040		6040	

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Ahmed Adeeb Jalal

Date of birth and place : 6th July 1980, Baghdad-Iraq

Foreign Languages : English

E-mail : ahmedadeeb80@gmail.com

EDUCATION

Degree	Department	University	Date of Graduation
Master			
Undergraduate	Software Engineering	Al-Rafidain University College	2001-2002
High School	Scientific Section	Al-Resala for Boys	1997-1998

WORK EXPERIENCE

Year	Corporation/Institute	Enrollment
2008-Current	Al-Iraqia University – Iraq	Software Engineer
2003-2008	Iraqi pipeline company – Iraq	Software Engineer
2002-2003	Al-Mustansiriya University – Iraq	Lecturer at computer labs

PUBLISHMENTS

Papers

1. Ahmed Adeeb Jalal and Oğuz Altun — "Engineering Mining a Large Scale Data Based on Feature Engineering, Metadata, And Ontologies," the International Journal of Digital Information and Wireless Communications (IJDIWC), August 2016, 6(4): 219-229.

Conference Papers

1. Ahmed Adeeb Jalal and Oğuz Altun — "Feature Engineering in Hybrid Recommender Systems," Proceedings of The Third International Conference on Data Mining, Internet Computing, and Big Data (BigData2016), 21-23 July 2016, Konya, Turkey, 1-14.