T. C.

ALTINBAŞ UNIVERSITY

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

Collaborative Filtering Recommendation System: Comparison Study

Inas Amjed Mohammed Al Mani

M. Sc. Thesis

Supervised by Prof. Dr. Osman UÇAN

Istanbul, 2018

Collaborative Filtering Recommendation System: Comparison Study

By

Inas Amjed Mohammed Al Mani

Electrical and Computer Engineering

Submitted to the Graduate School of Science and Engineering

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

[2018]

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Osman UÇAN
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Oğuz BAYAT	Engineering, Altınbaş	
Prof. Dr. Osman UÇAN	Engineering, Altınbaş	
Assoc. Prof. Dr. Ersin GÖZE	Vice Rector, National Defense	

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Çağatay AYDIN

Head of Department

Assoc. Prof. Dr. Oğuz BAYAT

Director

Approval Date of Graduate School of
Science and Engineering: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

INAS AMJED AL MANI

ACKNOWLEDGEMENTS

All praise and thanks are due to Allah S.W.T, and peace and blessings be upon his messenger, Mohammed (peace be upon him).

Firstly, I would like to express my thankfulness and gratitude to my advisor Professor Dr. OSMAN NURI UÇAN, for his support, patience, interest, fervency and guidance. His massive knowledge helped me to complete this thesis.

In addition, I am grateful and tankful to my parents, who always encouraged, motivated and supported me in my life. They try to do everything they could to light up my way to success. I want to dedicate great thanks to my father who taught me the meaning of perseverance and access to all things that I wanted. I am wishing maybe I could make you happy in your grave and be proud of me like you always crave, (may God bless you). I am also thankful to my brothers (Anas, Younis and Yousif), and my lovely sister (Aseel) for their support and their warm love and prayers. My sincere appreciation also extends to my dear and my helpmate, my husband (Ali), for his support, love, prayers and encourage. Thank you for the cheering me up and stood beside me through all the good and bad time without him I did not reach what I wish. Once again, thank you. Additionally, I want to thank my children (Malk and Harith) for their patience for the times that I have spent far away of them. Also, I cannot forget to thank all of my friends to encourage me to continue this study.

Finally, I would like to thank all staff of the Faculty of Electronic and Computing Engineering, for their kindness and cooperation during my study.

INAS AMJAD AL MANI

ABSTRACT

Collaborative Filtering Recommendation System: Comparison Study

AL MANI, Inas Amjad,

M.S, Electrical and Computer Engineering, Altınbaş University,

Supervisor: Prof. Dr. Osman Nuri UÇAN

Date: [May, 2018]

Pages: 86

Recommender systems (RS) have been getting serious attention in solving information overload problems by suggesting to users, items that are potentially of interest to them. Recommendation systems usually produce a number of suggestions in one of the given techniques. The RS divided into three types: Content-based filtering, Collaborative Filtering and Hybrid recommender system. Collaborative Filtering (CF) is the most popular recommendation technique and widely adopted in many commercial domains. However, CF does not consider any additional information, making it difficult to solve the cold-start and data sparsity problems. As found in most knowledge, likewise, Recommender system has some problems such as cold-start, data sparsity and scalability and so on; many researches are done to solve these problems and to increase the accuracy of the prediction. This study will produce comparison of different algorithms in RS, which are KNN, SVD and Naïve Bayesian to check the best performance of them. Two different sizes of dataset is applied (80% with 20%) and (60% with 40%) each size includes training and testing. Moreover, the performance is computed by three metrics (MAE, RMSE and time). The results revealed that Naïve Bayesian has highest accuracy in both metrics MAE and RMSE for both sizes of the datasets for; whereas in term of time the result was vary, depending on size of the dataset with the technique of the algorithm.

Keywords: Recommender system, Content-based filtering, Collaborative filtering, Latent Semantic Analysis and matrix factorization

ÖZET

İşbirliğine Dayalı Filtreleme Tavsiye Sistemi: Karşılaştırma Çalışması

AL MANI, Inas Amjad,

Yüksek Mühendis, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi

Danışman: Prof. Dr. Osman Nuri UÇAN

Tarih: Mayıs 2018

Sayfalar: 86

Tavsiye Sistemleri (TS) kullanıcılara potansiyel olarak ilgilerini çekebilecek seçenekleri önererek bilgi aşırı bilgi yüklenmesi ile ilgili problemlerin çözümü konusunda büyük bir ilgi görmektedir. Tavsiye sistemleri genellikle belirli tekniklerden birinde belli sayıda öneri üretir. Tavsiye Sistemleri üç türe ayrılır: İçerik Temelli Filtreleme, İşbirliğine Dayalı Filtreleme ve Hibrit Tavsiye Sistemi. İşbirliğine Dayalı Filtreleme (İDF) en yaygın kullanılan tavsiye tekniğidir ve birçok ticarî alana yaygın bir biçimde uyarlanmıştır. Bununla birlikte, İDF herhangi bir ek bilgiyi değerlendirmez ve bu yüzden soğuk başlatma ve veri seyrekliği problemlerinin çözümünü zorlaştırır. Çoğu bilgi için de söz konusu olduğu şekilde, Tavsiye Sistemi'nin de soğuk başlatma, veri seyrekliği ve ölçeklenebilirlik gibi bazı sorunları vardır. Bu sorunların çözülmesi ve tahmin doğruluğunun arttırılması için birçok araştırma yapılmaktadır. Bu araştırma, Tavsiye Sistemleri'ndeki farklı algoritmalar olan KEYK (k-En Yakın Komşuluk), TDA (Tekil Değer Ayrışması) ve Naïve Bayes arasında karşılaştırmalar üreterek bunların en iyi performanslarının kontrol edilmesini sağlayacaktır. İki farklı veri seti boyutu (% 80 ve % 20 ile % 60 ve % 40) uygulanmıştır ve her bir boyut eğitim ve test içeriğine sahiptir. Dahası, performans da üç ölçü (OMH (Ortalama Mutlak Hata), KOKH (Kök Ortalama Kare Hatası) ve zaman) ile hesaplanmıştır. Elde edilen sonuçlar, her iki veri seti için de gerek OMH gerekse KOKH’nda en yüksek doğruluk derecesinin Naïve Bayes ile sağlandığını, zaman açısından ise sonuçların veri setinin boyutuna ve algoritma tekniğine bağlı olarak farklılık gösterdiğini ortaya koymuştur.

Anahtar Kelimeler: Tavsiye sistemi, İçerik-temelli filtreleme, İşbirliğine dayalı filtreleme, Gizli Anlam Çözümlemesi ve matris faktörizasyonu

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

RS	Recommendation System
CF	Collaborative Filtering
K-NN	K-Near Neighbors
SVD	Singular Value Decomposition
PCA	Principle Component Analysis
LDA	Latent Dirichlet Allocation
FMM	Flexible Mixture Model
MF	Matrix Factorization
PMF	Probabilistic Matrix Factorization Model
CB	Content-Based Filtering Approach
IR	Information Retrieval
ML	Machine Learning
DT	Decision Trees
SVM	Support Vector Machine
ANN	Artificial Neural Networks
LIBRA	Learning Intelligent Book Recommending Agent

CBCF	Content-boosted collaborative filtering
COS	Cosine Similarity
PCC	Pearson's Correlation Coefficient
ACOS	Adjusted Cosine
CPCC	Constrained Pearson's Correlation Coefficient
SRC	Spearman's Rank Correlation
IP	Inner Product
SM	Singularity Measure
BS	Bayesian Similarity
WPCC	Weighted Pearson Correlation Coefficient
CPCC	Constrained Pearson Correlation Coefficient
EMDP	Effective Missing Data Prediction
RIP	Rated Item Pools
JacUOD	Jaccard Uniform Operator Distance
ECA	Electric Circuit Analysis
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
NDCG	Normalized Discounted Cumulative Gain
MAP	Mean Average Precision
DCG	Discounted Cumulative Gain

1. INTRODUCTION

1.1 OVERVIEW

Nowadays, we exist in the time of technology with the immensity and effortless attractiveness of information has made choices of people is myriad when they doing anything like buying products, reading blogs or e-books and/or playing any audiovisual files. In addition, the users face difficulties in obtaining useful information and in selecting the resources, which interests them. Recommender systems (RSs) becomes the trend of various reported papers in the literature to deals with information excess difficulties by proposing to customers the items that are potential of interests to them.

If anyone wishes for finding, the book may have numerous thousand books stacked on its shelves, libraries in cyberspace offer millions and millions of hardcover books, e-books as well as articles and newspaper whether for sale or rent. The inhabitants possess a huge information resources number. Conversely, the cost, time and complexity of people to find the suitable information are increased [1].This is because of unfeasibility of tracking the store to each individual customer. Merely the overall numbers, then, of what is made available governs the option. A bookstore, for instance, will exhibit only the books that are bestselling and popular [2].

Also, we can observe this hard in any audiovisual files, because it is tricky to prolong the applicable music. Thus, recommender systems have an imperative function in customizing and filtering the preferred information. Association rules mining method can be used to sort the related users, and subsequently, the associated songs are suggested to them. For finding the similar users, Frequent Pattern Growth Algorithm is being used [3].

The Ricci [4] describe the recommendation system, as an exacting category of the information system that gives three major tasks for the users. The 1st thing is RSs help them to create options without the adequate individual knowledge of the choices. The 2nd thing, it recommends merchandises such as movies to users, at the last; it offers users with information to assist them to choose which products to pay for. The user and item are the most important components that rely on the composition of recommendation system [5].

Recommender Systems (RSs) stand for "technique can recommend items to the active users that might be interested of his\her" [6].

Such suggestions are linked to a variety of decision-making processes. This is not limited, but its extent to which product to purchase, which music file to pay attention, which movie to look at, or which online information to read [4].

RS is addressed in numerous research disciplines, such as cognitive science, approximation theory, human-computer interaction, machine learning, information retrieval…etc. [7].

In addition, the RSs work on several technologies basis as in classification learning, information filtering, adaptive hypermedia and user modeling [8]. There are dual classes of duty to clarify the goal of the recommender system, rating prediction, and item recommendation. The 1st one, rating prediction which is indicating to the system predicts the lost magnitudes from the user-item rating matrix. In the item recommendation (also called a top-N recommendation), that is a recommender system returns the requesting user a sorted list of items as the final result. The recommender systems are functional in several areas to adopt recommendation for the users via rating prediction or item recommendation; the most domains that are generally used include (movies, books, restaurants, songs, journals, etc.).

After knowing the intention of recommendation, the recommender systems approach divided into three foremost divisions, collaborative filtering, content-based filtering, and hybrid approaches [9].

Collaborative Filtering (CF) approach is the mainly trendy recommender systems methods; depend chiefly concentrate on exploiting users' chronological favorites on the bought products to expect users' importance [10], [11].

In the other word, CF utilizes the related neighbors to produce recommendations for the users. CF is extremely practical in the real world and effortless to realize. There are two key classes of CF method: memory-based and model-based methods [12].

The memory-based approaches concentrate principally on finding the resemblance among items and users. Memory-based approaches can be further categorized as ‘user-based approaches’ [13], [14], [15] and ‘item-based approaches’ [16], [17] to forecast the suggestion of dynamic customers. The model-based approaches can be applied to exercise a distinct model and subsequently employ a model to envisage the suggestion. The aim of model-based algorithms is to discover the behaved outlines among users through dependence on machine learning or data mining methods. A lot of diverse models were projected to be applied in model-based approaches, such as the dimensionality reduction models together with latent semantic models [18], probabilistic matrix factorization model [19] and latent factor model [20].

Up until now, a variety of methods related to the notion of CF model-based method are proposed and used to carry out the recommendation. The 1st method in this field is the Probabilistic Matrix Factorization method (PMF) that works to recognize any unknown thing under the data [21]. Another technique is Singular Value Decomposition (SVD), which produces high-quality recommendations [22].

In a content-based approach, the system finds out items that are comparable to the ones that the user wished for formerly. The items resemblance can be determined derived from the characteristics connected with the compared items [23]. Content-based filtering employs the data items content to forecast its application based on the profile of the user. Research on content-based recommender systems is intertwined with many computer science subjects, particularly Information Retrieval and Artificial Intelligence.

The hybrid approaches continually employ the combination of content-based filtering and collaborative filtering to generate the recommendation [10].

Collaborative filtering is a well-known recommendation algorithm where the forecast and suggestion are done by the activities of other users in the system. The hybrid approaches for all time employs the integrating of content-based filtering and collaborative filtering to produce the recommendation [10].

1.2 PROBLEM BACKGROUND

As noted previously, the developments of information on the Internet, and the customers have been looking for systems that provide a high-quality recommendation, which made the need for recommendation services important. Since the development of the 1st research article on collaborative filtering in the mid-1990s, the intellectual study on recommender systems was grown considerably over the past 10 years. In general, Recommendation system provides propositions on content like which items to purchase, what music to pay attention or which online news to comprehend. Based on the users wishing for, the items are recommended [24].

The prime confronts in recommender systems facing the recommendation approach are the recommendation accuracy. For instance, this challenge always occurs as a result of the most two serious problems. The 1st problem, when the user performance data is excessively meager for the recommender system to supply reasonable suggestions. This issue, in general, referred to as the data sparsity problem it means if recommendations systems are employed for the big number of merchandises. In that case, the user-item matrix size is sparse and huge thus it is difficult to create recommendations and to uphold the recommendations performance [25]. The other problem is the cold start. It appears as one situation of the sparsity data challenge, which relates to the case when a new item or user accesses the system; it is tricky to locate the same ones for the reason that there is meager information. Both of these problems cause negative results on the recommendation quality of the recommender systems applications. This issue in the collaborative filtering approach could be solved by using hybrid approach items.

Occasionally, the rates and reviews provided by the user who not often utilizes his profile are unrelated comparing to the profile of a huge record. This case is known as trust, it is possibly fixed by providing the main concerns to the user by observing and then estimating the things they usually surf for and purchase [24].

In such approaches, the system works in the two dimensional space, i.e., the RSs utilize rating in user-item rating matrix as a knowledge source to make the recommendation such as [13], [14], [15]. That scenario makes recommendations in terms of the rating information and does not consider further information that can be extremely essential for enhancing the recommendation [10]. Otherwise, it was found the performance comparisons of two accomplishment approaches of collaborative filtering, which are memory-based and model-based, using data sample of PT X e-commerce such as [26].

Many studies were conducted in the field of Recommender Systems. But, the developed methods are relatively uncomplicated. Collaborative Filtering algorithms are an influential optional formulation appear when Netflix.com suggested in 2006 one million dollar award to any person or research collection that can develop "Cinematch" by ten percent or better [27], [28]. This has taken three years for the award to be lastly given, that depicts the massive difficulty to enhance these systems. The accomplishment, actually, has been performed when the peak candidates linked forces in a shared effort, near to the end line, as a final point to seize the award that remained off target year after year.

These Recommender Algorithms are termed Model-Based [29], [4]. They use matrix factorization methods to estimate the latent factors, that loosely stated could be reflected as fundamental domain descriptors. For instance, in a film recommender system, underlying factors could explain drama against comedy, age suitability, violence amount, and are determined by factoring the items ratings and users sparse matrix.

As matrix factorization presently has a superior hand on Recommender Systems, its advantages are not free. For appetizers, the algorithm is rooted in factoring an essentially sparse matrix that can simply be confirmed not to comprise a distinctive solution.

The matrix factorization methods are two; the 1st one is Probabilistic Matrix Factorization method (PMF). The 2nd method is Singular Value Decomposition (SVD). [30], indicate that the resulting reduced orthogonal dimensions from SVD are less noisy than the original data and capture the latent associations between the terms and documents. Earlier work [31] that had the benefit of this semantic characteristic to decrease the feature space dimensionality.

The foremost scheme of latent factors use is productively adopted for Information Retrieval in 1980. [32], exploited SVD to discover latent factors in documents [33]. [22]

And [34] convey the exploitation of latent semantic relations, PCA 1 or SVD idea to recommender systems [22], [34]. In Netflix Prize contest, a matrix factorization model has been better than typical ones. BellKor used ALS2 method and concentrated on extra information, for instance, temporal dynamics or demographic data [35]. Paterek effectively used a variety of matrix factorization methods. He inserted partialities to regularized matrix factorization [36]. Additional matrix factorization methods that are professionally applied are: the latent semantic analysis based on Probability [37], probabilistic matrix factorization [38].

One important algorithm that researcher used in model-based of collaborative filtering is Bayesian algorithm to build forecasts for the tasks of CF. Let the features are separately specified for the class, the likelihood of a definitely given class for the entire features can be determined and subsequently the class with the uppermost probability will be categorized as the forecasted class [39].

Along with Collaborative Filtering Recommender System, another method to the classic approach is memory-based. The neighborhood-based CF algorithm, a prevalent memory-based CF algorithm, has the subsequent procedures: compute the matching or weight, $w_{i,j}$, that mirrors distance, weight , or correlation along with two items or two users, i and j; construct a prediction for the user activities by obtaining the weighted average of the entire ratings of the user or item on a definite user or item, or by means of a uncomplicated weighted average [17]. If the task is to produce a top-N recommendation, we require locating k the majority of analogous items or users (nearest neighbors) after the similarities computing, after that combine the neighbors to acquire the top-N most frequent items as the recommendation. This technique called K- nearest neighbors (KNN).

1.3 SIGNIFICANT OF THE STUDY

The needing of accuracy system to recommend items to the users has much value. This thesis studied three techniques of recommendation system namely KNN, SVD and Naïve

Bayesian to compare the performance. In addition, the study helped us to discover the accuracy of recommendation system for these techniques in three terms MAE, RMSE and time.

1.4 SCOPE OF THE STUDY

- The datasets are selected from standard datasets from different sources which are specialized for recommendation system (Movie Leans, Jester, and YOW).
- Evaluation: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and time.
- Computer specification: Windows 7 (32 bit), RAM (2G) and Matlab 2017.

1.5 THESIS OUTLINE

This thesis arranged in chapters, chapter one includes introduction and definition of the recommendation system, explains the problem background, establishes scope of the study and shows significant of the thesis. Follow by; chapter two contains basic knowledge, the literature review of the modern studies. Moreover, the types of recommendation system are explained, also the computation methods and types of similarity are illustrated. Likewise, the challenges are demonstrated. Whereas, chapter three explains the methodology that achieved the objective of the study.

Furthermore, chapter four describes and discusses the results of the study. Finally, chapter five shows the summery and future work for this study.

2. LITERATUREVIEW

2.1 INTRODUCTION

In this chapter, a comprehensive review, and study of recommender systems studies are shown. This chapter includes the elementary concepts of a recommender system, the feedback of recommendation, the typical approaches using in the recommender systems, the challenges in the recommender systems, and the areas in which the recommender systems have been adopted. Furthermore, this chapter also reviews the noteworthy efforts by researchers, which have been put into the new tracks of recommendation. Besides, gives the categorization of the used methods and gives details the basic concepts related to it. The literature reviews on other concepts related to the current study; such as link prediction techniques, gradient descent method, and standard quantum-based similarity method are as well presented in this chapter.

2.2 RECOMMENDATION SYSTEMS

Recommendation systems have been exceedingly universal. It facilitates the customer to realize information and choose alternatives where they do not have the necessary learning to judge a specific item [40]. Rising information on the Internet has made it tricky to handle data and programming challenges [41]. Recommendation works on a routine user proposition that suits his requirements. In addition, it helps users to get the information they want Faster and reduce non-useful data [42].

Recommendation system depends straightforwardly on the feedback information and the manner in which such information is obtained and used relies on the particular recommendation method. User preferences can be obtained by following their activities, additionally, they may be asked by recommender system [41]. Recommendation system assists the users to determine what they require and do not ask them to determine what they want to do with the information. Many researchers presented several definitions for recommender systems. For instance, [4] defined recommender systems as technique can recommend items to the active users that might be interested of his\her" [6]. The [43], describes recommender systems as "systems that get the opinions concerning items from a community of users and then use those outlooks to direct other users within that community to those items that are interesting for them". Figure 2.1 shows the environmental factors of recommendation, including (users, data, recommender system and application).

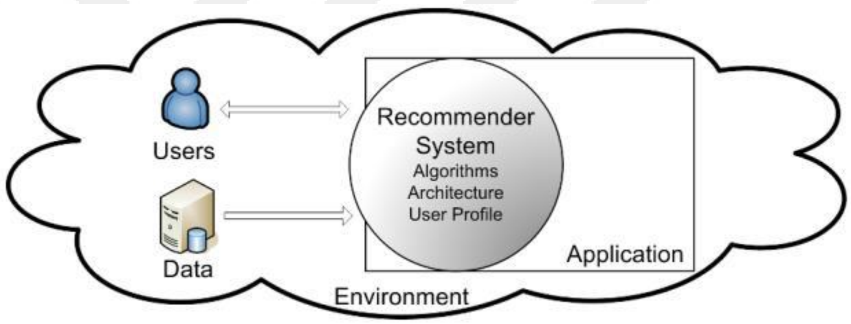

Figure 2.1: Recommender system environment [44]

2.3 OVERVIEW OF RECOMMENDER SYSTEMS

The predicting theories, information retrieval, approximating theory, the cognitive science, and to the consumer option modeling in marketing, and works in numerous kinds of the research area all of this led to the emergence the recommendation system [45]. In the mid-1990, the researchers begin to find out an innovative research area dependent on the recommendation problems that clearly mention the on rating structure [10].

Collecting all available information about customer requirements (e.g., books, movies, music, apps, websites, flight destinations…etc.). There are dual methods to obtain this information. The 1st one explicitly (by collecting user's evaluations of products). The 2nd one is implicitly (Monitor the user websites visit and the links download). Additionally, RS may also make use of demographic properties of users, together with age, race, and gender.

The straight (explicit) feedback is based on degree (1- 5) or by binary (like, dislike), it has not essentially to be suitable to magnetize the user's interest to evaluate an item. The indirect (implicitly) feedback information is characterized by the relationship score of the definite user options like saving, watching, printing, etc. Namely, the process is achieved using a distanced-monitoring procedure devoid of user's direct participation [4].

2.3.1 User-Item Rating Matrix

Rating matrix is a very imperative matrix for the reason that without it the recommended items to users are unattainable [2]. The aggregations of the users who have given an assessment of a set of items in the information domain for the system of CF. There are lots of kinds of ratings someone by integer value score like 5-star scale, while other use (like/dislike) measures or unary evaluations, as in "has purchased" [4], [29].

In general, the value of (the user, item) should return the user's preference for a particular element and if it would be without the evaluation it is going to be unidentified. Table 2.1 explains the evaluation matrix for 5 users and 7 movies, where U = $\{u_1, u_2, \dots, u_n\}$ to signify the users, I = $\{i_1, i_n, \dots, i_m\}$ for the products, and R as $n \times m$ rerecorded rating matrix in the system, with $u \in U$, $i \in I$.

Table 2.1: A rating matrix of movies on a 1–5 star

	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	5		3	2			1
u_2	4			3			2
u_3	1	4	2		5		
u_4		5					3
u_5		4	4			5	5

All users rate an item as to illustrate his/her attention in that particular item. For instance, in the five-star rating system, 1 and 2 stars characterize undesirable feedbacks, whereas 3, 4 and 5 stars stand for the optimistic feedback. if the rating unexploited (no rating location) in the matrix, RS algorithm can predict this no rating location; also proposes an item to a user if his/her predicted rating for this item says, 3, 4 or 5 stars [7].

2.4 CLASSICAL RECOMMENDER SYSTEMS APPROACHES

The investigation of recommendation system has been grown since the information and data are augmented which led to the appeared problem of overload information, so it was applied to the entire academic and industrial area. Recently, studies began to identify which RS based on, whatever recommendation is personalized or not, the setting of data and the type of input data. However, the modern taxonomy of the RSs is classified reliant on the made recommendations. In view of that, RSs is possibly sorted into three broad classes: content-based filtering, collaborative filtering, and hybrid approaches. Figure 2.2 below presents the taxonomy of RSs.

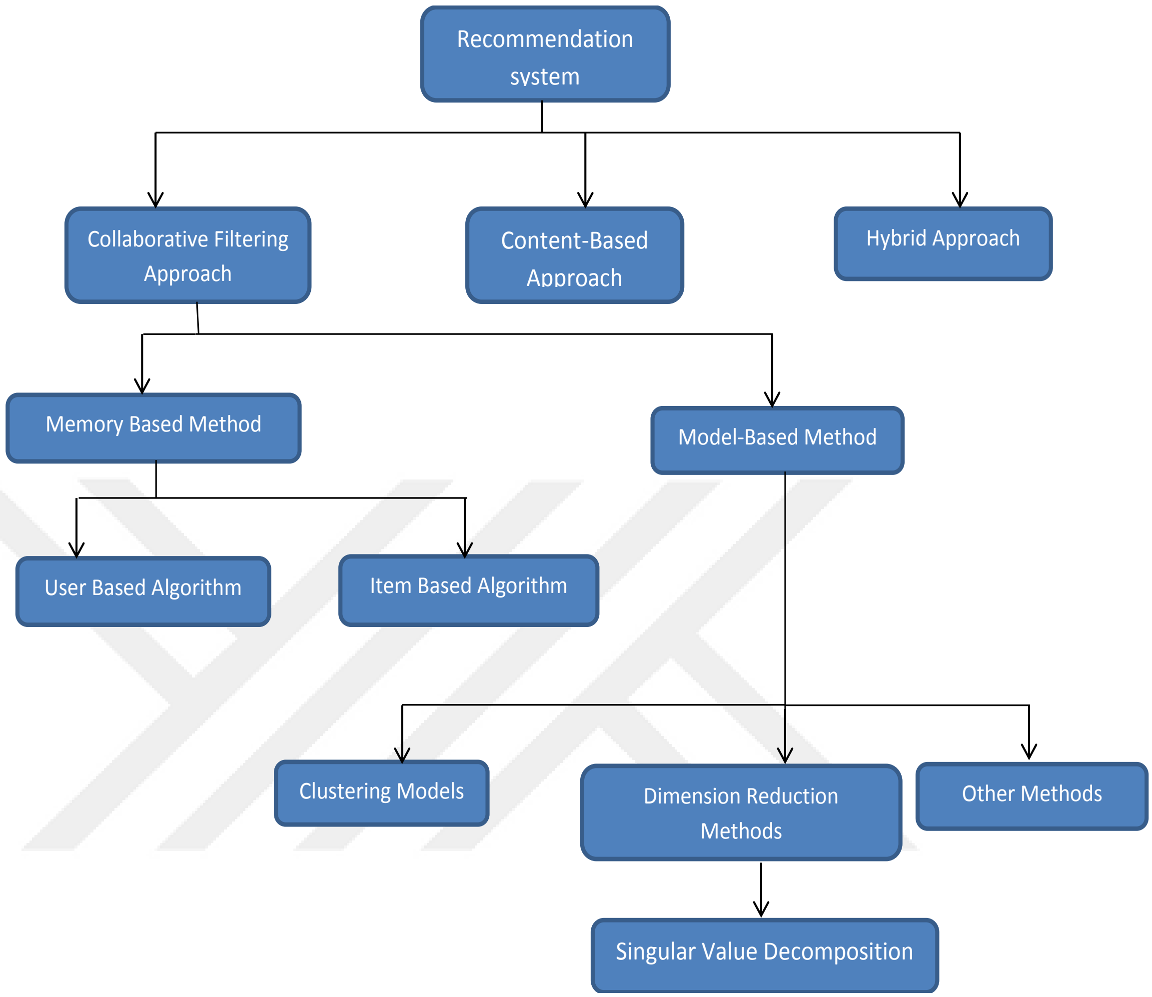

Figure 2.2: Descriptions of classical recommender systems approaches

2.4.1 Collaborative Filtering Approach

The CF approach is highly well liked recommender systems methods; CF steps principally concentrate on employing users historical favorites on the bought items to estimate user's interests [10], [11]. CF proposes items to the active user from comparable users in the system. CF approaches presume similar neighbors to generate recommendations for the users [46]. In addition, the CF approach is considered as active in the real world, simple to realize and apply [10].

It is broadly used in numerous E-commerce tracks with businesses such as Amazon [47], TiVo and Yahoo! [48], several CF systems were proposed to cover E-resource domain as in Pocket Lens [49], Fly-casting [50] and Smart Radio [51]. According to [52], the CF approach compared to other traditional recommender systems approaches still take part in a dominant role in almost all kinds of application fields such as E-learning, E-resource, E-commerce, and E-tourism.

The CF subfield of recommender systems is trendy owing to numerous causes; most consider the circumstance that the data have a much-uncomplicated structure: user-item matrix. All the same, all regarding the items that must be suggested throughout the application of the CF methods is unidentified excluding the user rating. Furthermore, the accessibility of data arrangements for CF leads to the expansion and the construction of lots of CF methods [41].

There are two central kinds of CF approaches: memory-based [53], [16] and [15] and model-based [66], [18], [65] and [12] methods. The huge dissimilarity with the memory-based and model-based algorithms is their processing data mode. The model-based guesses the ratings by using statistics and machine learning techniques to be trained about a model from the central part of data, whereas the memory-based algorithm has several heuristic regulations to forecast the ratings [54]. Model-based methods are more precise then memory-based methods and were studied to fix the shortcomings of memory-based CF methods [55], [13].

2.4.1.1 Memory-Based Methods

Memory-based CF approaches are dealing straightforwardly with ratings of users with the intention of predicting and endorse unvalued items. As a rule, the likeness metrics amid items or users have been employed in this technique, for each of their ratings [42], [56] and [57]. The memory-based methods (neighborhood-based approaches) concentrate essentially on similarity finding among items or users. Memory-based methods are feasibly extra categorized as ‘user-based algorithms’ [13], [14] and [15] and ‘item-based algorithms’ [16], [17] to foresee the recommendation of active users. This algorithm operates based on the entire user-item rating matrix and it works based on three steps as shown in Figure 2.3

i. Determine the resemblance among effective users and other users.

ii. Choose the neighbors (comparable users).

iii. Create commendations.

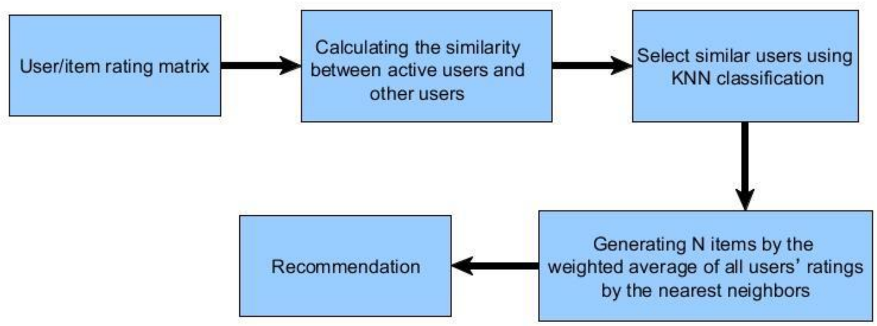

Figure 2.3: Memory-based recommendation process

a. Compute the Similarity:

We can see from the Figure 2.3, the similarity evaluations are one of the central elements for collaborative filtering approach, as the recommender systems usually face a problem of how they can compute the similarity among items or users. The further identical two users are, the more possible it is that the other will like a latest liked item by one of these users. Dissimilar measures of similarity (distances) are possibly applied to a problem [58] (more details and discussions on similarity methods in Section 2.4).

The most common similarity measures are Pearson's correlation coefficient and cosine distance. The 1st one can be adopted to compute the similarity among items in recommender systems and they are specified by their correlation, that calculates the linear relationship between objects [59]. Based on the predefined covariance of data points x and y and their standard deviation σ using equation (2.1) we can compute the Pearson correlation [59]:

$$Pearson(x, y) = \frac{\sum(x,y)}{\sigma x * \sigma y} \tag{2.1}$$

Cosine similarity is another approach to take into account the items as document vectors of n-dimensional and calculates their likeness as the cosine of the angle that they make equation (2.2):

$$cos(x,y) = \frac{(x \bullet y)}{\|x\| \|y\|} \tag{2.2}$$

Where x is the norm of vector x and • indicates a vector dot product.

b. K-Near Neighbors (K-NN) Classification:

The k-NN is one of the more complicated and more contemporary classification algorithms in data mining field, and it has numerous benefits [46], [60]. Such as, make numerical predictions in complex functions and remain easy to interpret, tell us which variables are important in making prediction ‘any variable get scaled down too can be thrown out' and k-NN is an online method we can adjoin new data at whatever time. For this reasons, the k-NN algorithm becomes used in some applications such as recommend. User-based filtering and item-based filtering methods of k-NN that user-based algorithm produces the forecasts for users by performances or scores of analogous users whereas item-based algorithm creates the forecasts based on likenesses among items.

In the user-based version, k-NN commends reliable items for an effective user by performing the following steps [61]:

i. Define k user's neighborhood for the effective user by means of the similarity measure.

ii. K neighbors of effective users are then found, and the forecast value of item i for a is calculated.

iii. The topmost-n items are designated and suggested to the effective user.

Even though user-based CF is efficacious approaches, it undergoes from memory and time necessities when using this technique on a large-scale dataset [17]. The item-based procedures use the likeness amid items to compute the forecasts.

c. Generate The Recommendations:

For the identified neighbors set, the last recommendation process causes a recommendation for an active user, this step generating N items by the weighted average of all users' ratings by the nearest neighbors. By far the most common approach to aggregate ratings as anticipated by [62], for an item i for active user u.

$$p_{i,k} = \bar{r}_i + \frac{\sum j \in N(i)\ sim_{i,j} * (r_{j,k} - \bar{r}_j)}{\sum j \in N(i) \left| sim_{i,j} \right|} \qquad (2.3)$$

$P_{i,}$ k, is the predicted rating for user i for item k, i r is the mean score for user i, $sim_{i,}$ j, is the likeness amid users i and j, and N (i) is the neighborhood of user i.

As indicated previously, the memory-based methods are the trendy prediction techniques and are extensively used in a lot of marketable recommender systems. Many researchers have used various memory-based methods in their work. For example, [47] used memory-based in amazon.com and used three methods: classical collaborative filtering, search-based methods and cluster models to resolve the problems of recommendation. In [62], the Group Lens system used memory-based for Netnews, to assist users to get articles they will require.

Memory-based approaches can be additionally categorized as 'user-based algorithms [13], [14] and [15] and 'item-based algorithms' [16], [17] to envisage the recommendation of user activities. User-based algorithms used the scores of their related users to predict the recommendation; [63] presented a framework for collaborative filtering. Likewise, [14] present an optimization technique to regulate the weights of ratings routinely from learning users. Item-based algorithms forecast the ratings of involved users from the evaluated information of items close to those selected by the active user [64]. [16] Developed concerns model-based recommendation to solve the complications of scalability. In the work presented by [17], they explored an item-based approach to solve the increased number of participants in the system by the relationship identification among items in the system.

For determining similarity among users or items, user-based and item-based algorithms frequently employ the vector space similarity [13] and the Pearson correlation coefficient [62] algorithms. In CF approach, the Pearson correlation coefficient method normally can accomplish superior results over the other popular method cosine similarity algorithm, as it takes the differences of user rating style into consideration [64].

2.4.1.2 Model-Based methods

In the model-based methods, learning datasets can be adopted to instruct a distinct model and after that employ a model to forecast the commendation. In the other word, Model-based CF methodologies exploit users' response data to construct a forecast model and at that point endorse items to the effective users. Namely, those model-based CF procedures do most of the tricky work in the training phase, where they build an investigative model of the commendation problem" [10].

Numerous models are suggested to be applied in model-based methods. For example, clustering models [65], [67], Bayesian models [7], the dimensionality reduction models including (latent semantic models) [18], probabilistic matrix factorization [19] and latent factor model [20].The following two sections deal with clustering and dimensionality.

i. Clustering Models:

A clustering method stands for a process of collecting analogous objects in spaces into sets, each member (object) in this one set is considered similar to other objects and non-similar to other groups. Clustering algorithms are used widely in many different applications. In computational biology, a clustering is used to group genes holding same proteins functional. In computer science, the clustering method used in software evolution, data mining search result grouping, crime analysis, etc. The clustering methods become very useful for many tasks and especially for CF have been extensively studied by several studies. For example, [68] cluster users and items into two groups independently by using a hierarchical clustering algorithm especially when few data were available and attempts to make equilibrium for accuracy and robustness of predictions. They experimentally show the method is constructive for dealing with sparsity problem and develop the recommendation accuracy.

There is also another significant contribution on clustering techniques for collaborative filtering. For example, in work presented by [67], they build method by combining between model-based and memory-based collaborative filtering and used clusters to supply smoothing procedures to resolve the lost-value difficulties [67]. [69] Recommended items by combining user clustering with item clustering, and the algorithm uses item clusters to solve the problem of cold-start for presenting new items. Likewise, [70] proposed cluster algorithm to partition the user profile into clusters based on similar items to deal with new items problems; and show the use of clustering can enhance recommendation performance on new items.

- **Naïve Bayes Classifier**

It represents widespread learning method for machine learning and data mining [71], [72]. The mainly general technique to carry out CF with a probabilistic method is to outlook the favorites forecast problem as a sorting problem, where it is characteristically described as the process of allocating an object to one of a number of predefined groupings [73]. Naïve Bayes is employed as model-based probabilistic RS technique; it has the capability to calculate the likelihood of user's likely interests directly, where neither description of similarity nor distance is looked-for. Nevertheless, this model applies the obtainable likings to forecast an objective favorite, where these favorites are conditionally autonomous given the target favorite.

The difficulty can be illustrated by given instance with n attributes (X_1, X_2, X_3,…., X_n), and the aim is to forecast the class C_k so must exploit $P(C_k|X_1, X_2, X_3,…, X_n)$ or alternatively by Bayes' theorem [74]:

$$P(C_k|X_1, X_2, X_3, \ldots , X_n) =\frac{P(C_k)\prod_{i=0}^{n}P(X_i|C_k)}{P(X_1,X_2,X_3,\ldots X_n)} \tag{2.4}$$

To calculate the $P(X_1, X_2, X_3, \ldots , X_n|C_k)$ the Naïve Bayes gives:

$$P(X_1, X_2, X_3, \ldots , X_n|C_k) =P (X_1|C_k) P(X_2|C_k) \ldots P(X_n|C_k) \tag{2.5}$$

Assuming the characteristics are autonomous in a specified class, the likelihood of entirely the structures is considered, subsequently, the extreme likelihood is categorized as the foreseen class [39].

A Bayesian technique has gotten worthy consequences, where the likelihood computations are not influenced pointedly by noise data and inappropriate attributes. Henceforth, this method performance is still great [73]. Table 2.2 shows the steps of Naïve Bayesian prediction model.

Naïve Bayes Prediction Algorithm

Input: Aij the training set of user-item matrix, where i = 1, 2, 3... U and j=1, 2, 3… M, U: number of users and M: number of items

Output: N recommended items for each user.

Begin

- Create the co-occurrence matrix.

- For each user i

- For each new item j (new item means that Aij is empty)

- Calculate the conditional probability between item j and rated items of user i.

End for j

- Recommend the N highest probability items to user i.

End for i

End

ii. Dimensionality Reduction Methods

Dimensionality decrease means of trimming the quantity of applicable data while conserving the most important information content. It is frequently practiced in areas such as data mining, machine learning, and cluster analysis. The majority of methods of dimensionality decrease include feature extraction that utilizes out of sight variables, or supposed latent variables, to define the fundamental reasons of co-occurrence data" [75]. According to [75], “dimensionality lessening is well applicable in model-based collaborative filtering, as for the majority of applications. Only a small fraction of user-item pairs are studied such that the number of applicable variables can be notably reduced” [75].

Many dimensionality reduction methods were carried out in recommender systems, including Singular Value Decomposition (SVD) [76], Principle Component Analysis (PCA), and Latent Dirichlet Allocation (LDA) [77]. An underlying semantic CF method depends on a statistical modeling method that presents underlying class variables in a combination model adjusting to find out user communities. [18] Depicted new model-based approaches and offered an algorithm based on an overview of probabilistic latent semantic analysis to continuous-valued response variables.

Furthermore, [37] put forward a statistical model; that is better than typical memory-based methods, higher accuracy, constant time prediction, and an explicit and compact model representation; [78] presented for collaborative filtering a flexible mixture model (FMM). They extend the clustering algorithms into collaborative filtering method via clustering both users and items with each other simultaneously. The investigational result illustrated that proposed model (FMM) is capable of outperforming other approaches for collaborative filtering assignment virtually.

A lot of researchers were exploiting the Matrix Factorization (MF) methods to lessen the data sparsity and cold start problems, such as [79], [80], by removing misleading or inconsequential users or items to decrease the dimensionality of the user-item matrix ratings. For instance, in [81], the authors present straightforward and efficient algorithms for solving weighted low-rank approximation problems and apply the method for collaborative filtering. [82] For maximum margin matrix factorization examined a direct gradient-based optimization method. There were also efforts by researchers to use the probabilistic into dimensionality reduction. [19] Presented the Probabilistic matrix factorization model (PMF), where the MF and probabilistic models are combined in a systematic way, that scales linearly with the number of explanations and, more significantly, perform satisfactorily on the large and sparse dataset.

The matrix factorization methods allocate the same idea with Singular Value Decomposition (SVD) [76]; this approach lets anyone determine the latent features of the interaction amid users and items. Matrix factorization procedure was demonstrated to be successful to solve most of CF constraints.

Singular Value Decomposition (SVD) - It is frequently used in the solution of unrestrained linear least squares problems, matrix rank estimation, and canonical correlation analysis [83].The foremost scheme of underlying elements is productively used for Information Recovery in 1980. [32], used SVD to explore latent factors in documents [33]. [22] And [34] reassign the notion of exploitation of underlying semantic relations, SVD or PCA 1 to recommender systems [22], [34].

SVD is a popular matrix factorization method that factorizes an m ´ n matrix R into three matrices as:

$$A= U \times S \times V^T \tag{2.6}$$

Where U and V are two orthogonal matrices. Matrix U is m×r, the eigenvectors of AA^T form the columns of U. Matrix V is p×r, the eigenvectors of A^TA compose the columns of V. as well, the singular magnitudes are diagonals elements of matrix S, that are only positive and prearranged in the downward sort. Matrix S is r×r. The matrices gotten by doing SVD are mainly functional for our relevance since SVD has the finest lower rank rough calculations of A matrix based on Frobenius norm. It is feasible to decrease the r×r matrix S to include just k biggest diagonal magnitudes to find a matrix S_K, $k < r$. If matrices U and V are decreased in view of that, the reassembled matrix $Ak = U_K.S_K.V_K^T$ is the neighboring rank-k matrix to A. Namely, A_K lessens the Frobenius norm $\|A - A_K\|$ overall rank-k matrices. This technique is termed as simple SVD, which is applied to the original matrix [84].

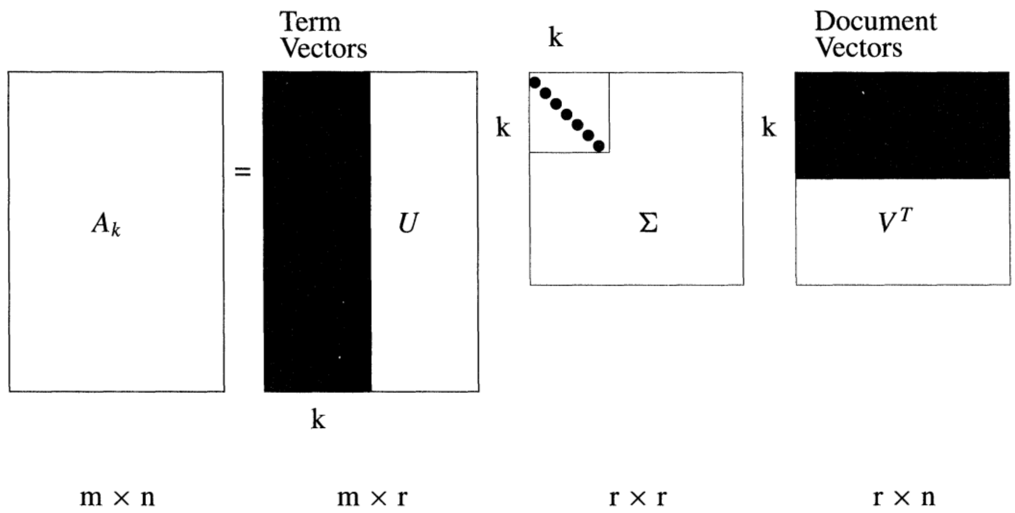

Figure 2.4: Mathematical representation of the matr& Ak.

In general, SVD can be adopted in Recommender Systems to decrease the preference matrix dimensionality since it can offer the finest low-rank estimate of the original matrix [85]. The procedures are:

1) Assume A is the rating matrix and decomposes it as follows (5) into 3 matrices $U_{m\times r}$, $S_{r\times r}$ and $V_{n\times r}$
2) Decrease the preference matrix dimensionality to k dimensional and pick merely k biggest diagonal magnitudes from matrix S, k < r. As a result, it acquires a matrix $U_{m\times k}$, $S_{k\times k}$ and $V_{n\times k}$.
3) Determine U and S matrixes that stand for m users in the k dimensional feature space.
4) Compute S and VT matrixes that stand for n items in the k dimensional feature space.
5) The envisaged evaluation for user u_i on item j is specified by:

$$v_{i,j} = \bar{v}_i + U\sqrt{s(i)} \times \sqrt{s}V^T(j) \tag{2.7}$$

Where, $v_{i,j}$ is a foreseen rating that is given by user u_i for item j and $\bar{v}_i$is the typical rating of user u_i to the items.

2.4.2 Content-Based filtering approach

A review of some methods proposed based on Content-Based filtering approach (CB) in this section will be discussed. A number of researchers, such as [86], [87] turned to make the recommendation for the users utilizing the specifications of CB by exploiting the essence of data items to expect its weight in accordance with the user. Wherefore in most of CB recommendations, item depictions are textual characteristics extracted from emails, Web pages, product descriptions or news articles [5]. Figure 2.7 depicted the architecture of a CB approach. As we can see, the recommendation sequence achieved as follows:

i. Content analyzer - The first element is the content analyzer. It’s the information source that used in the process of recommendation. The major aim of the content analyzer is working to map the non- structural information to represent it in an outline appropriate for the subsequent processing steps.
ii. Profile learner - This module utilized the content analyzer output as input assembles data courier of the user favorites and attempts to simplify this data, to build the user profile. The profile learner component always used machine-learning techniques to collect the data.
iii. Filtering component - Filtering component utilizes user profile to propose applicable items as the last step in the content-based recommendation process.

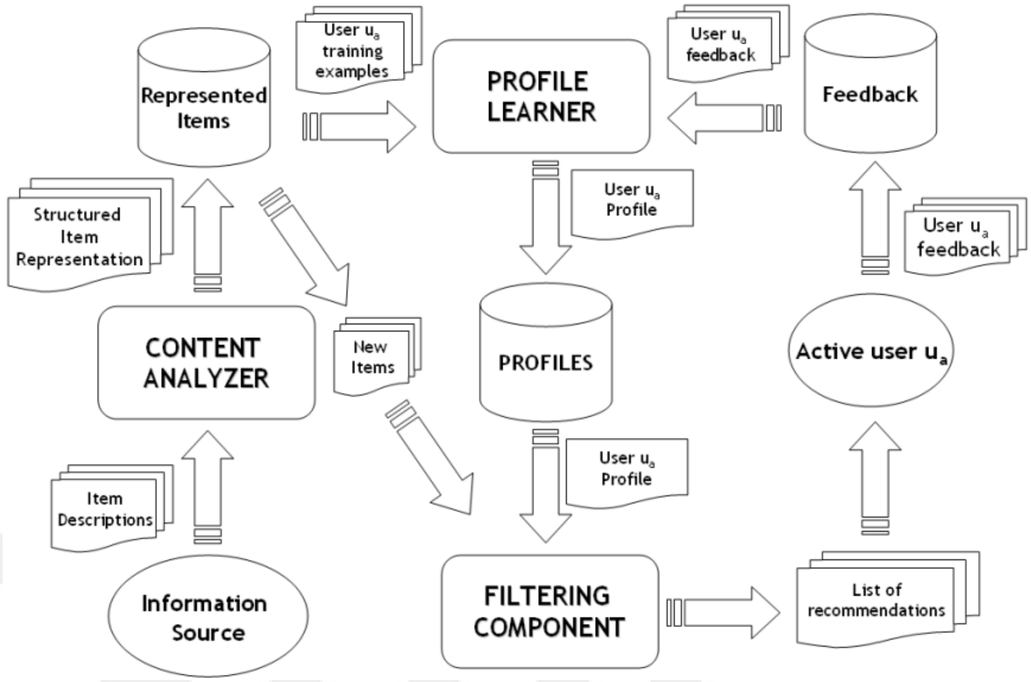

Figure 2.5: Content-Based approach architecture [5]

In CB approach, the recommender systems are trained to propose items that are related to the previous user's preferences. The resemblance among items can be computed by the characteristics connected with the compared [4].CB approaches can further classify as two techniques to generate recommendations:

One technique generates recommendations using the Information Retrieval (IR) methods [88], [5]. As examples, similarity methods "cosine similarity and Pearson correlation coefficient similarity" based on heuristically way. The associated content with the preferences of users considered as a query, and then the non-rated documents are achieved with application related to this query. The other technique an alternative to IR methods to generates recommendations; it is Machine Learning (ML) techniques [89], [90]. As an example, K-Nearest Neighbor (KNN) [91], Decision Trees (DT) [92], Support Vector Machine (SVM) [93]. Moreover, utilizing the Artificial Neural Networks (ANN) [94], through treat recommending as a classification task [95]. Several CB systems have been proposed to cover many recommendation domains, such as TV domain like PTV [97] and AVATAR [96], and music domain such as Foaling the Music system [98] and Music FX [99]. As examples of the CB recommendation for the domain of web, are Web Watcher [100] and Web sail [101]. For instance of CB approach for web recommendation, [102] exploit the implicit feedback from users and viewing frequency for web pages to build the ACR News Vectors. Then clustering model utilized to train, and pages are recommended via a CB approach on related clusters. Figure 2.8 describes a general architecture for the personalized web system divided into two components.

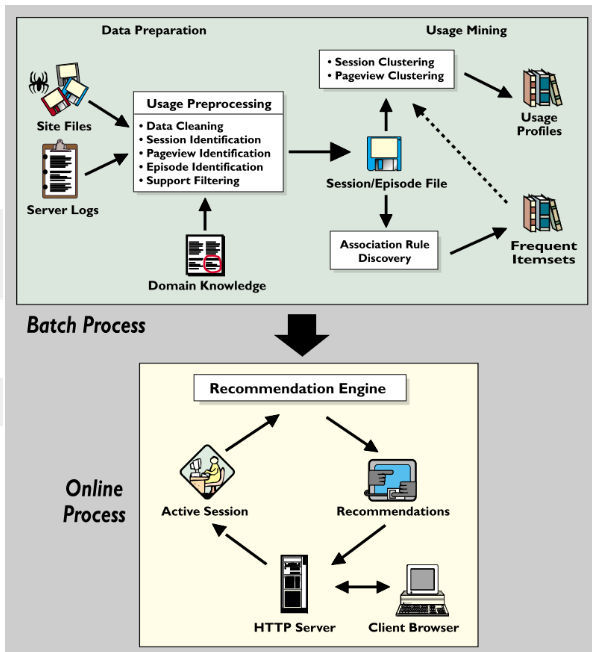

Figure 2.6: A widespread structural design for usage-based Web personalization [102]

Much research on CB approaches has focused on the domain of book recommendation utilizing ML methods. In work presented by [103], they proposed LIBRA system (Learning Intelligent Book Recommending Agent), exploits text from fields as in the subject terms, reviews, synopses, and title, to guide ML technique as in the naive Bayes classifier. Based on 1 to k, rating scale can be straightforwardly charted to k classes, or instead, the numerical rating can be employed to evaluate the training case in a probabilistic binary classification situation. In addition, the [1] is current book recommendation classification for the digital library. The researcher investigated user profiles that have been the record of using book category and interrelated data.

Additionally, the developed model of book recommendation system to assist users for book searching showed that the users pleased with the book recommendation system.

2.4.3 Hybrid approaches

In general, these approaches mix content-based filtering and collaborative filtering. Terms this combine provides to take the advantages that existing each method to obtain more accurate predictions such as [55], [104] and [105]. According to [10], there are three aspects can be described hybrid approaches:

i. First aspect CB and CF each one works alone and then combine their predictions to generate the final recommendations [106], [86].
ii. The second aspect adds some of CF characteristics inside of CB and then CB generate the recommendation [107].
iii. The last one adds some of CB characteristics inside of CF and then CF generates the recommendation [108], [109] and [110].

In presented work by [109], they suggested a general framework for content-boosted collaborative filtering (CBCF), it makes a better-personalized recommendation by knowing more about an item, such as the director and genre of a movie for the movies recommendation. Figure 2.9 shows the architecture of their recommendation framework.

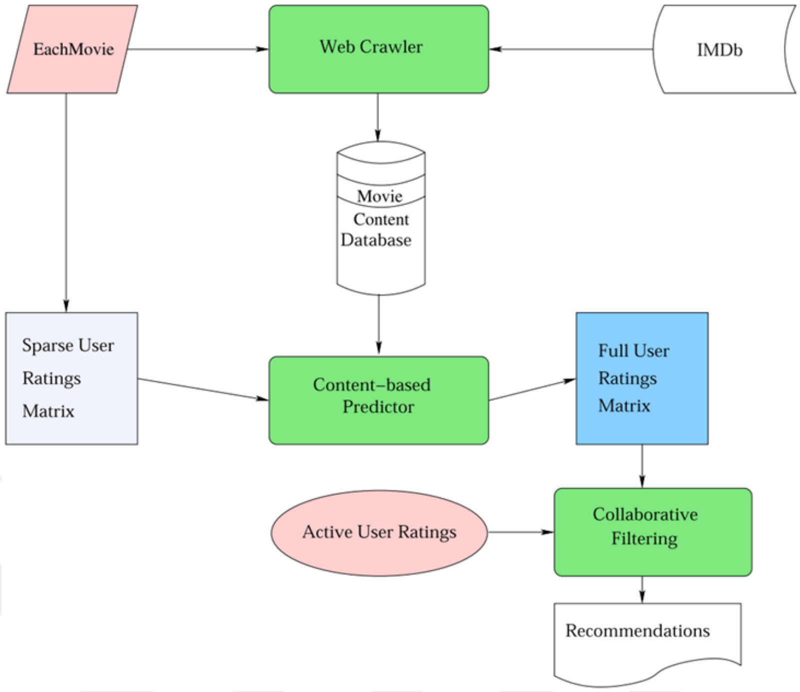

Figure 2.7: Content-Boosted Collaborative Filtering (CBCF) approach design [109]

As examples of the hybrid, approach for recommendation domain of movies is the Cinema Screen [110]. The collaborative filtering process supplies a subjective record of the movies for the existing user into the CB filtering process. The weights are a sign of ratings by as a minimum one statistically noteworthy peer of the present user. Figure 2.10 is shown this a process in details.

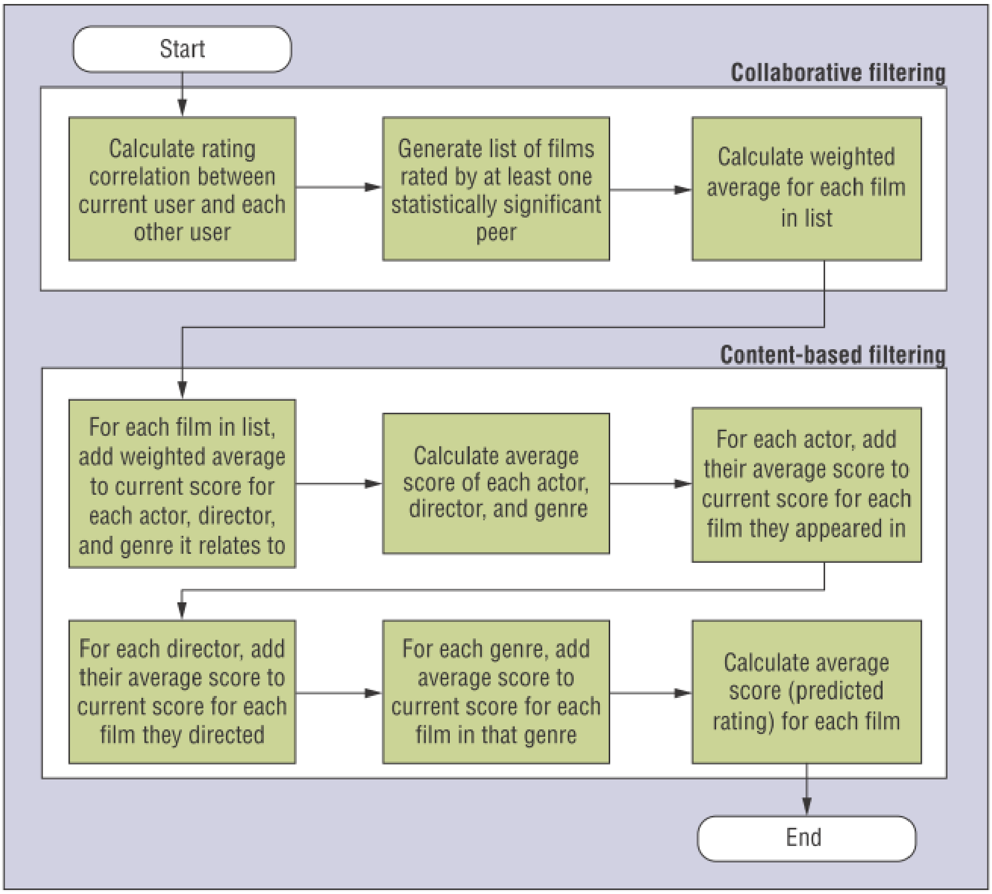

Figure 2.8: Cinema Screen recommendation process [110]

Hybrid recommender system joins CF and CB to overcome the constraints of any recommender system and in that way enhance recommendation behavior. Later, given two basic RSs approaches (CF and CB), seven basic mechanisms of the combination are suggested for integrating them to structure an innovative hybrid approach [111].

As such, many researchers exploited this combination ways to proposed different hybrid methods. There have also been efforts made by [112], they proposed two hybrid CF methods, sequential mixture CF, and joint mixture CF, it is similar to content-boosted CF algorithm [109]. These approaches perform well in the sparse data environment. In Labor system [113] employs instance-based learning to produce content-based user profiles that are subsequently evaluated in a shared manner. Through both types of recommender systems approaches, [114] depicted the development of a Web 2.0 TV program recommendation system “queveo.tv.” Through use methods, CF and CB complement are in such a manner, which the content-based method suggests the typical programs and CF provide the finding of recent shows. In later work, [52] proposed hybrid approach (CCF) by bringing both content-based filtering and collaborative filtering approaches together for the news topic recommendation.

2.5 SIMILARITY MEASUREMENT APPROACHES

This section introduces the conformist similarity-based recommendation that is often memory-based collaborative filtering method. Moreover, it reviews the related work in the track of similarity measure for recommender systems. Similarity computation is the backbone of collaborative filtering as neighborhood formation is done based on these values. To estimate the resemblance between any two users or items, the majority collaborative filtering methods adopted Cosine Similarity (COS) [13] and Pearson’s Correlation Coefficient (PCC) [62]. The main concept for both of them is to opt for the ratings of the users that rated both items. Although utilized the COS and PCC notably in memory-based CF, there have been many other methods. Table 2.2 presents the similarity measures frequently used for CF.

Table 2.2: Similarity measures frequently utilized for CF [118]

Measures	Definition
Pearson's Correlation Coefficient (PCC)	$$sim(u_x,u_y)=\frac{\sum_{h=1}^{n`}\left(r_{u_x,i_h}-\bar{r}_{u_x}\right)\left(r_{u_y,i_h}-\bar{r}_{u_y}\right)}{\sqrt{\sum_{h=1}^{n`}\left(r_{u_x,i_h}-\bar{r}_{u_x}\right)^2}\sqrt{\sum_{h=1}^{n`}\left(r_{u_y,i_h}-\bar{r}_{u_y}\right)^2}}$$ where $r_{u,i}$ are the ratings of item i by user u, $\bar{r}_u$ is the average rating of user u for all co-rated items, and n ' is the number of items co-rated by both users.
Cosine (COS)	$$sim(u_x,u_y)=\frac{\sum_{h=1}^{n`}r_{u_x,i_h}r_{u_y i_h}}{\sqrt{\sum_{h=1}^{n`}{r^2}_{u_x,i_h}}\sqrt{\sum_{h=1}^{n`}{r^2}_{u_y i_h}}}$$ where $r_{u,i}$ is the rating of item i by user u and n ' is the number of items co-rated by both users
Adjusted Cosine (ACOS) for Similarity between items	$$sim(i_x,i_y)=\frac{\sum_{j=1}^{m`}\left(r_{u_j,i_x}-\bar{r}_{u_j}\right)\left(r_{u_j,i_y}-\bar{r}_{u_j}\right)}{\sqrt{\sum_{j=1}^{m`}\left(r_{u_j,i_x}-\bar{r}_{u_j}\right)^2}\sqrt{\sum_{j=1}^{m`}\left(r_{u_j,i_y}-\bar{r}_{u_j}\right)^2}}$$ Where $r_{u,i}$ is the rating of item i by user u, $\bar{r}_u$ is the average rating of user u for all the items rated by the user, and m' is the number of users who rated both of the items
Constrained Pearson's Correlation Coefficient (CPCC)	$$sim(u_x,u_y)=\frac{\sum_{h=1}^{n`}\left(r_{u_x,i_h}-r_{med}\right)\left(r_{u_y,i_h}-r_{med}\right)}{\sqrt{\sum_{h=1}^{n`}\left(r_{u_x,i_h}-r_{med}\right)^2}\sqrt{\sum_{h=1}^{n`}\left(r_{u_y,i_h}-r_{med}\right)^2}}$$ where $r_{u,i}$ is the rating of item i by user u, r_{med} is the median value in the rating scale (e.g. 3 in the rating scale of 5 and, it is 4 in a scale from 1 to 7.), and n ' is the number of items co-rated by both users
Spearman's Rank Correlation (SRC)	$$sim(u_x,u_y)=1-\frac{6\sum_{h=0}^{n`}d_h^2}{n`(n`^2-1)}$$ Where d_h is the difference in the ranks of item h by the two users and n ' is the number of items co-rated by both users

Diverse enhancements for the conventional similarity measure had been presented in the literature, given the incompetence of the COS and PCC methods [115]. They can be further classified into three kinds: a novel, new similarity measures; second modification, that made some upgrading for the traditional similarity measures COS and PCC by overcoming their disadvantages and, thirdly adaptation, which utilized a measurement idea from dissimilar domains to recommender systems domain. Table 2.3 shows the classification of similarity methods that have been applied to collaborative filtering approach.

Table 2.3: Improvement categories of the similarity measure

Enhancement method	The proposed Similarity methods
Novel	Inner Product (IP) [116], [117], Proximity, Impact, and Popularity (PIP) [118], Singularity Measure (SM) [119], and Bayesian Similarity (BS) [115]
Modification	Weighted Pearson Correlation Coefficient (WPCC) [63], Constrained Pearson correlation coefficient (CPCC) [120], Effective Missing Data Prediction (EMDP) [121], Rated Item Pools (RIP) [122], and Jaccard Uniform Operator Distance (JacUOD) [123]
Adaptation	Electric Circuit Analysis (ECA) [15]

According to [116], COS and PCC work quite well for explicit ratings; but it is suffered to deal with implicit ratings. For instance, [116], they proposed new similarity measures Inner Product (IP) in place of the traditional measures to solve the difficulties of unconstructive favorites and normalization in contained ratings. Another extensive investigation by [118], recommends the new similarity measure PIP according to three semantic heuristics: Proximity, Impact, and Popularity. PIP tries to increase the divergence of similarity among users with semantic accords and those with semantic variances in ratings. In later works, [15] proposed ECA similarity measuring for collaborative filtering; they adopted the electrical circuit investigation to determine the potential variations among nodes on an electric circuit. The tentative consequences demonstrate that ECA technique carries out greatly superior to the state-of-the-art approaches. There is also another significant contribution on using the similarity algorithms in different recommender models. For example, the [64], [124] and [125] solve the sparsity problem, by exploiting the social relation between users and incorporated into matrix factorization model. The objective of similarity methods it is to compute the similarity between the user and their friends and used the values in the process of recommendation models.

In this study, our notion of similarity is similar to the previous works. The research adopted the similarity algorithms to analyze the similarity value among the users to find the nearest *n* users to the certain user.

2.5.1 Binary Similarity and Distance Measures

It is a standard term for the patterns. It was an essential subject in the pattern investigation challenges like the classification and clustering [126]. There are a number of categories of binary matching measures. Altogether concentrate on the measurement of response that locates a similarity concerning any two comments. The logic fundamental of these techniques is the degree that the two users have a common pattern of attributes [127].

Jaccard and Dice's measures were widespread binary similarity measures. Table (2.3) gives you an idea about the 2×2 matching for the two objects (X and Y) if X= $\{x_1,x_2,x_3,\ldots, x_n\}$ and Y=$\{y_1,y_2,y_3,\ldots,y_n\}$. The symbol ***a,*** in the Table 2.4, is the features number where the magnitude of x_i and y_i are both 1, that means a positive matches, whereas ***b*** is the features number where the value of x_i is 0 and value of y_i is 1, ***c*** is the number of features where the value of x_i is 1 and value of y_i is 0, and ***d*** is the features number where xi and yi have 0, indicates ‘negative matches' [127].

Table 2.4: 2×2 matching list

		Y	
		1 (Presence)	0 (Absence)
X	1 (Presence)	A	B
	0 (Absence)	C	D

From Table 2.4, similarity and distance measures for Jaccard and Dice are computed as by [126]:

$$Jaccard_{sim} = \frac{a}{a+b+c} \tag{2.8}$$

$$Dice_{sim} = \frac{2a}{2a+b+c} \tag{2.9}$$

$$Jaccard_{dis} = 1 - Jaccard_{sim} \tag{2.10}$$

$$Dice_{dis} = 1 - Dice_{sim} \tag{2.11}$$

Jaccard and Dice measures stand for the undesirable match exclusive measures, where the negative matches are not at all times measured as a criterion of the similarity amid dual objects [128]. For this reason, the positive matches are frequently more noteworthy over the negative matches [126]. In equations (2.8), (2.9), Dice and Jaccard measures are the same that offers high weight for positive matches in case of Dice similarity.

For instance, suppose that Table 2.5 has dual binary data vectors, the Jaccard and Dice measures are analyzed by:

Table 2.5: Two binary data vectors

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9
X	1	0	1	0	1	1	0	0	1
Y	1	1	0	0	1	1	0	0	0

Along with the matching table the values of a = 3, b = 2 and c = 1, then the values of similarity and distance measures is

$$Jaccard_{sim} = \frac{3}{3+2+1} = \frac{3}{6} = 0.5$$

$$Dice_{sim} = \frac{2*3}{2*3+2+1} = \frac{6}{9} = 0.66$$

$Jaccard_{Dis} = 1 - Jaccard_{sim} = 1 - 0.5 = 0.5$

$Dice_{Dis} = 1 - Dice_{sim} = 1 - 0.66 = 0.34$

2.6 CHALLENGES OF RECOMMENDER SYSTEMS

Even if CF systems are employed in many domains such as industry or academy, a number of concerns are a serious problem for the researchers in the recommender systems area. Here, the dissimilar restrictions face by the main approaches of traditional recommender systems will be reviewed, which were discussed in the literature of the preceding sections and focusing chiefly on collaborative filtering approach.

i. Data Sparsity

The sparsity problem happened in collaborative filtering when the user rates extremely a small number of items from the whole number of items in the user-item matrix. This issue happens since it cannot be determined whether the user did not evaluate an item because he did not like it or simply because he has not experienced it [129].

Therefore, with the quickly rising number of users and items, the difficulty of data sparsity is increasingly intractable. Accordingly, the used user-item matrix for collaborative filtering was massive and sparse, that has confronted in the recommendation performances. In the literature, the sparsity of ratings has been discovered as a major problem for CF approach. Several proposals have been made to solve this issue ([130], [131] - [132] - [133] and [134]. The researchers have suggested a compensation system to reduce the sparsity problem by which users are rewarded for providing ratings to items. Others have offered to capture the ratings by implicitly look at the user's behavior [135].

i. Cold Start

This issue is a regular trouble in recommendation systems. It appears as one situation of the sparsity data challenge. The cold start happens in collaborative filtering because of the absence of information about users or items.

The problem of cold start is the condition at what time a recent item or user goes into the system. In this situation the recommender systems do not have any information about new users and items based on that, there are two categories of cold start problems: new item and new user problems. In these cases, it is tough to supply a suggestion such as a new user case [136], became deficient in information about the user that is existing. Moreover, for a new item problem, no ratings are typically presented and consequently collaborative filtering cannot create functional recommendations in case of a new item in addition to the new user [137].

Nevertheless, several studies begin to conquer this problem, such as [138 and [105]. Moreover, this issue in the collaborative filtering approach could be solved by using hybrid approach items because the hybrid approach tries to evade the restriction in both methods collaborative filtering and content-based filtering.

ii. Scalability

The scalability is the capability of the recommended methods to deal with large real-world data. As the numbers of users and items grow, the classical collaborative filtering approach will suffer from scalability issue. Unlike sparsity problem, the scalability problem may present a more resilient challenge, because the number of ratings will continue increasing over time. Moreover, scalability problem one of the big challenge for memory-based CF, since all computations are performed at the prediction time. The performance effectiveness of collaborative filtering is essentially between O (M+N) and O (M^2 N), where M is users number and N is items number [139]. Item-based method less sensitive to scalability problem compared to the user-based method in collaborative filtering approach [16] and [140]. Because the computing the likeness amid users in the user-based method is online; unlike in the item-based method that calculates the similarity of the items offline.

Improvement of recommender systems is of paramount importance and more research work to address this challenge. The scalability issue can be considered as a common problem among all the recommender systems approaches. The problem could be solved by several methods such as in [141], [142] - [143] and [67]. Some of the primary solutions to the scalability problem have been based on model-based techniques such as dimensionality reduction techniques.

iii. Privacy-preserving

Privacy stands for one of the challenges existing in the recommender system applications. When we necessitate constructing an ideal recommender system, we must keep in minds to defy user privacy and make them feel insecure. However, recommender systems operate by collecting user data, creating, and storing user profiles to match them and find similar users.

It has been discussed that the richness or features of the user profile enormously influences the quality of recommendations received [129]. However, a possible and valid concern that a user might have while parting with personal, sensitive data is that of misuse of the data for malicious purposes. As a result, there is a necessity to propose solutions that will economically and wisely use user data [144]. Some methods are proposed to conserve the privacy of users and their data [145], [146] - [147] and [148].

iv. Gray Sheep

Gray sheep represents the users whose attitudes are constantly consented or not consent with at all people clusters and accordingly do not make use of collaborative filtering [150]. The black sheep concept referred to the conflicting cluster whose personal tastes formulate recommendations almost unfeasibly. Although this is the recommender system failure, non-electronic recommenders also include huge troubles in these cases; as a result, black sheep is a tolerable failure [149]. A hybrid approach is considered as a good solution for Gray sheep problem. For example, [150] offered a hybrid approach mixing CF and CB to address the gray sheep problem [150]. Some methods used the clustering technique for gray sheep users' problem. A clustering solution is proposed to detect these users [151].

v. Shilling Attack

The other confront in recommender systems is "Shilling Attack." In RSs, where all persons can offer the ratings, natives are possibly providing many optimistic ratings for their items so that they will be recommended to other users and negative ratings for their competitors. These attacks can be tough to detect, as it is hard to tell which users are fake and which are real in most cases.

Several methods are proposed to deal with this situation such as [152], [153] - [154] and [155]. Most of these studies are based on the Probabilistic model. Recently, the shilling attacks models for collaborative filtering system have been identified, and their effectiveness has been studied such as [153].

2.7 EVALUATION OF RECOMMENDER SYSTEMS

The major objective of the research in the recommender systems is to improve the value of suggestions, which are produced. Performance assessment was measured as one of the main concerns in recommendation systems' development. Providing that the recommendation is practically a method to grasp about other objectives (customer contentment, positive marketing, better sales, etc.), emerging RS algorithms require considering this and measuring the intentional effects. On the other hand, execution of the different algorithm on actual e-commerce and then measuring properties are can be pricey [29].

Offline valuation is measured as a fundamental part of the RS. It is supportive to regulate the finest processes before concerning real e-commerce users. It is correspondingly useful for directing comparisons among algorithms [156].

As a general rule, the rating matrix R is separated into a training set R-train to learn f, and a test set R-test adopted to calculate the forecast accurateness. A forecast assignment can be used to calculate a lost rating in the user/item matrix. So, the difficulty of discovering the most excellent items is typically transferred into recommending a list of N items to a dynamic user u probable to be of interest to him/her, that is as well stand for as Top-N Recommendation [4],[7].

This entire process is frequent on a recurrent base as in N-fold cross-validation by allocating the rating matrix R into N of sets with identical size, at that point, every set is used as the test set as well as wholly other training sets. Accordingly, the system performance is feasibly evaluated by in view of the results of each run. That method can regulate the special effects of test set variation else) [10].

2.7.1 Rating Prediction Accuracy

Rating prediction accuracy determines the dissimilarity along with the scoring the method predicts and the real rating; in other words, the accuracy measurement frequently evaluates the predictions by dividing the rating data into a training and test set. In the rating data, a training set is applied to develop a method, while a test set is to validate the method built. The proposed methods are then asked to predict ratings of items in the test set and the accuracy of the algorithm are proportionate to the correctness of predictions on the ratings in the test set.

There are numerous metrics to determine a variety of features of recommendation accuracy. Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) has been extensively employed in many studies [157], [158] - [159] and [15]. MAE and RMSE can be employed to assess the nearness of expected ratings to the factual ratings [75]:

The MAE metric is defined as:

$$MAE = \frac{1}{T}\sum_{i,j}\left|R_{i,j} - \hat{R}_{i,j}\right| \tag{2.12}$$

The RMSE metric is defined as:

$$RMSE = \sqrt{\frac{1}{T}\sum_{i,j}\left(R_{i,j} - \hat{R}_{i,j}\right)^2} \tag{2.13}$$

$R_{i,j}$ is the rating user *i* gave to item *j*, $\hat{R}_{i,j}$ is the rating user *i* gave to item *j* as foreseen by a technique, and *T* is the total number of verified ratings.

2.7.2 Item Recommendation Accuracy

It is also called classification accuracy, measures how well the recommender system method differentiates good items from bad ones. The evaluation measures commonly used in classification tasks Precision and Recall were initially suggested by [160]. Precision and Recall are most well liked that are eminent in the RSs literature [161], [22] - [15] and [162].

The Precision metric is defined as:

$$Precision = \frac{Number\ of\ correctly\ recommended\ items}{Number\ of\ recommended\ items} \tag{2.14}$$

Where the number of properly recommended content items is the set of applicable made recommendations for the user, a number of recommended items is the completely made recommendations number for the user.

The Recall metric is defined as:

$$Recall = \frac{Number\ of\ correctly\ recommended\ items}{Number\ of\ actually\ preferred\ items} \tag{2.15}$$

Where the number of properly suggested content items is the set of applicable recommendations made to the user, and a number of actually preferred items be the set of items that are attractive along with the user's preferences.

2.7.3 Item Ranking Accuracy

A recommender system reflects the demanded user a sorted record of items as the ending result. Thus, we can compare the ranking order of the recommended items with the real ranking order to test set as an assessment of the recommendation performance. The metrics Normalized Discounted Cumulative Gain (NDCG) [163] and Mean Average Precision (MAP) [156]; are an extensively adopted measure of ranking quality [164], [165] - [166] and [167]. By other words, the metrics measure the exactness of returned results and their ranking over the user's acknowledged likings [166]. For NDCG metric first, the Discounted Cumulative Gain (DCG) accumulated at a particular rank

Position p is defined as:

$$DCG@p = \sum_{i=1}^{p} \frac{2^{rel_i - 1}}{\log_2(i+1)} \tag{2.16}$$

Where rel_i denotes the ranking score at position i.

The NDCG is then calculated by:

$$NDCG@p = \frac{DCG@p}{IDCG@p} \tag{2.17}$$

Where IDCG is the DCG of the "ideal ranking order", i.e., the ranking order based on the actual ratings in the test set.

The other metric, the MAP is defined as:

$$MAP = \frac{1}{N} \sum_{i=1}^{N} \frac{i}{Pi} \tag{2.18}$$

N refers to the applicable recommendations number; $\frac{i}{Pi}$ refers to the value of precision at a specified cut-off rank Pi, and i refer to the applicable recommendations number of rank Pi or less.

Our proposed methods used the MAE and RMSE; this is due to the study focusing on predicting the missing rating in the user-item rating matrix. For a prediction based recommender system, it is necessary to evaluate whether the prediction ratings are accurate.

2.8 SUMMARY

In this chapter, RS is explained in details. In the beginning, an overview of RS is produced with recent studies of this field. Then, types of RS is illustrated which has many algorithms that applied to each type of RS, wherein each type there are certain algorithms that implemented by the previous studies. Followed by the similarities of RS is demonstrated, where the similarity contains many methods that proposed to find the similarity in the matrix. Moreover, one of the critical points of RS is stated which called challenges of RS. Finally, the evaluation of RS also explained with the methods and formulas that used in this topic.

3. RESERCH METHODOLOGY

3.1 INTRODUCTION

Based on the literature review discussed in Chapter 2, the research identifies the limitations in the methods nowadays to address the recommender system problems. The main goal of this thesis is to compare the performance of the three will-knows methods namely KNN, SVD, and Naïve Bayesian, where the evaluation is achieved by different category of datasets which include high density and high sparsity.

This chapter presents the methodology used in this research. It was necessary to prepare a suitable methodology before implementing this research to compare the accuracy of the recommender system. A methodology is a guideline for solving a research problem. It contains the general framework of the research and the steps required to carry out the research systematically. These steps include discussion on the research components such as the phases, techniques, and tools involved. This chapter encompasses the Sections that covered the methodology. Firstly, 3.1 explained in the introduction. The proposed method of this study is explained and presents an operational flowchart in Section 3.2. Section 3.3 is about literature review and data preparation and includes four subsections: problem formulation, literature review, data gathering and data preprocessing. The evaluation matrices discussed in 3.4. Then in 3.5, 3.6 and 3.7 explain the proposed methods of this study with the experimental setting of the significance test, and the state-of-the-art methods. Finally, the chapter ended with a summary in Section 3.8.

3.2 PROPOSED METHOD

In this study, three models are executed to perform the recommender system; the three models are applied by using three well-known methods namely KNN, SVD, and Naïve Bayesian, these algorithms computed the rating of certain items to the users those did not rate these items before. Then, the comparison process is done by computing the performance of these methods. Three comparison models are applied by using (KNN, SVD and Naïve Bayesian); Figure 1 shows the flowchart of this study.

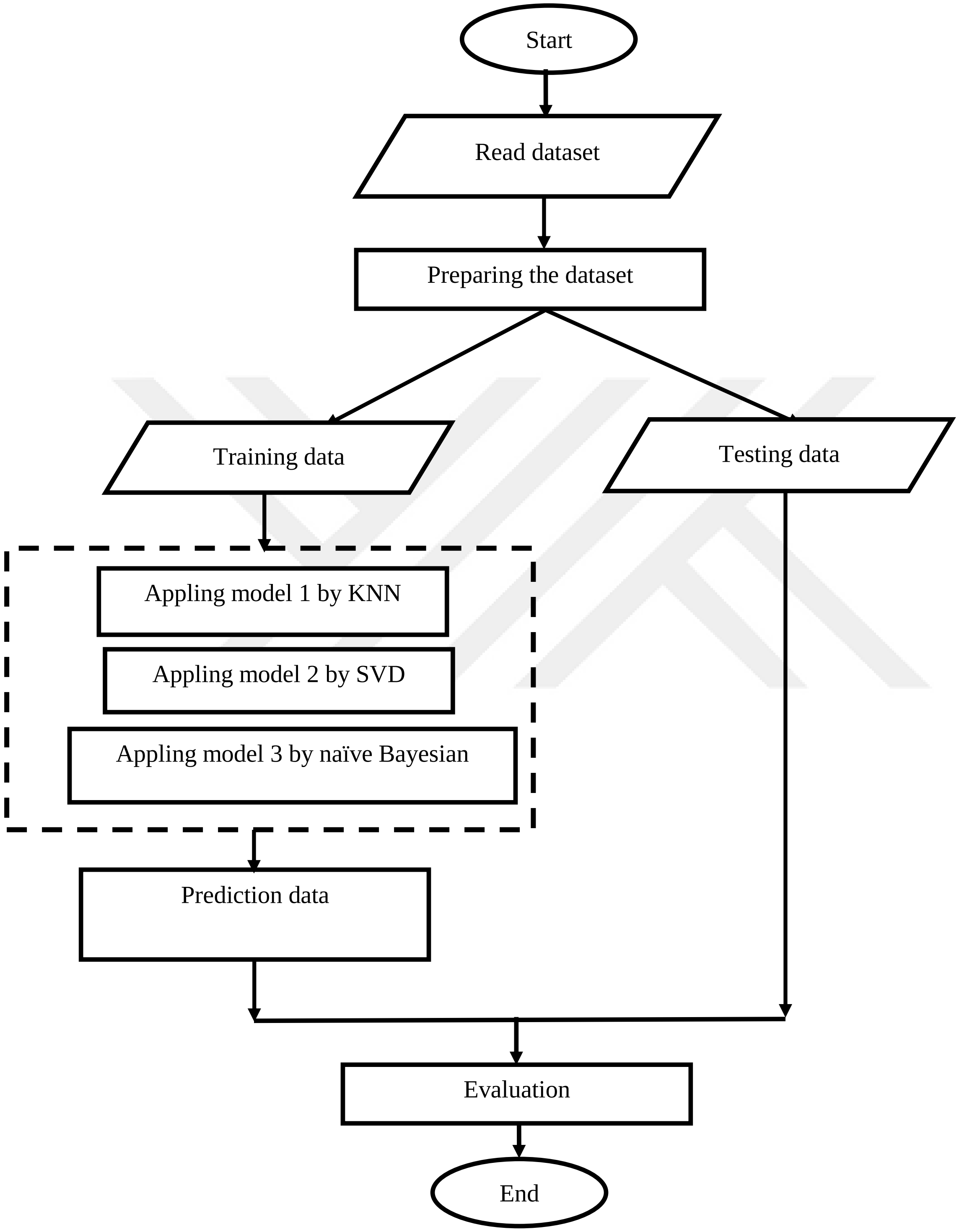

Figure 3.1: Flowchart of the proposed method

3.3 LITERATURE REVIEW AND DATA PREPARATION

This section focuses on describing the premier study regarding to this study. It covers the four main elements: problem formulation, literature review and identifying existing technique, selection of data set and preparation of data.

3.3.1 PROBLEM FORMULATION

Addressing the research problem is the main point that needs to implement any research. In this study, we address the problems in the different techniques of recommendation system. Through problem formulation, we can determine quite particularly the questions of this study that need to answer.

3.3.2 LITERATURE REVIEW

The literature review examined works related to the current recommender system methods to analyze the recommendation techniques used in these methods.

We also collect any useful information related to the study that will help us to define the research problems more clearly. Through reviewing previous works, the best techniques from the model-based method, link prediction technique, machine learning technique and similarity method were used for the proposed solution method.

3.3.3 DATA GATHERING

Another important element is data gathering. Data gathering involves selection of datasets to be used for the purpose of research evaluation. Several datasets have been widely used to evaluate the performance of recommendation methods, such as Movie leans, Jester and Yow. We need to use a dataset that has different categories (high density and high sparsity) to differentiate which algorithms could predict more accuracy items than others could.

For example, Movie leans dataset has accepted density, thus this dataset used in most studies of the recommendation system, whereas, Jester dataset has high-density data, therefore the researchers used this dataset with certain algorithms that deal with high-density data. In contrast, Yow considers high sparsity, for that to get more accurate results we could not apply the some algorithms those applied with Jester, but it needs another type of algorithms such as dimension reduction algorithms and probability.

Movie leans dataset is movie recommender system led by the Group Lens Research has collected and made available rating data sets from the Movie Lens website (http://movielens.org). The datasets were collected over various periods of time, depending on the size of the set. The first released the dataset in 4/1998 with 100,000 ratings from 1000 users on 1700 movies. Recently, the dataset has become 11 million computed tag-movie relevance scores from a pool of 1,100 tags applied to 10,000 movies, Released 3/2014. The rated value of Movie leans is discrete rating by how not like the items (1/5) to how like it (5/5).

Jester dataset is an online joke recommender system by Ken Goldberg from UC Berkeley. This dataset contains 4.1 million continuous of 100 jokes from 73,496 users collected between April 1999 and May 2003. The second version contains over 1.7 million continuous ratings of 150 jokes from 59, 132 users and it is collected between November 2006 and May 2009. In addition, there is an updated version of the second dataset with over 500, 000 new ratings from 79,681 total users (http://www.ieor.berkeley.edu). The Ratings in Jester dataset are real values ranging from (-10.00 to +10.00).

YOW dataset collected at the Carnegie Mellon University for the Yow-now news filtering system, This dataset encompasses around 25 people and 7000 feedback entries from all users, which contains 5921 articles rated by each user. The rating of this dataset is collected from (1 to 5) (http://users.soe.ucsc.edu/yiz/papers/data/YOWStudy).

3.3.4 DATA PREPARING

The preparing processing step is one of the necessary steps required for most experiments in computing the prediction. Since the original datasets are large and difficult to implement by personal computers, therefore in Jester dataset which is consist of 25000 users, we construct a new subset by randomly choosing some users and the new subset has become 16000 users.

Usually, the dataset is divided into two parts a training set and testing set, is a technique used to learn the model and validate the effectiveness. In the dataset, to build up model a training set should implemented, whereas the purpose of the test model is to prove the model built. In this study, we utilized two amounts of ratings as training data (80%, and 60%), and the rest size of dataset employed for testing data (20% and 40%). For example, 80% of the training data means we randomly selected 80% of the whole dataset ratings from the user-item rating matrix as the training data (storing the selected rating in matrix1) and the remaining 20% of the ratings as a testing phase(also storing in matrix2). In both evaluations, we employed the prediction accuracy metrics namely MAE and RMSE to measure the accuracy of the recommendation model.

3.4 EVALUATION METRICS

In this thesis, we employ two prediction accuracy metrics namely Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) to measure the quality of the recommendation.

MAE measures the average absolute deviation of the predicted rating of an item from the actual rating for the item. A smaller value of MAE signifies better prediction quality and the mean absolute error is given by:

$$MAE=\frac{1}{T}\sum_{i,j}\left|R_{ij}-\hat{R}_{ij}\right| \qquad (3.1)$$

A related accuracy metric is the RMSE is another widely used evaluation metric on recommender systems, which squares the error before summing them, the smaller value of RMSE, the more precise a recommendation and is defined as follows:

$$\text{RMSE}=\sqrt{\frac{1}{T}\sum_{i,j}\left(R_{ij}-\hat{R}_{ij}\right)^2} \tag{3.2}$$

Where R_{ij} denotes the rating user i gave to item j, $\hat{R}_{ij}$ denotes the rating user i gave to item j as predicted by a method, and T denotes the total number of tested ratings.

3.5 MODEL ONE APPLYING K-NN ALGORITHM

KNN is one of most common algorithms that used in data mining and it has high publicity in Recommender systems. Figure 2 shows the steps of implementation of the KNN algorithm.

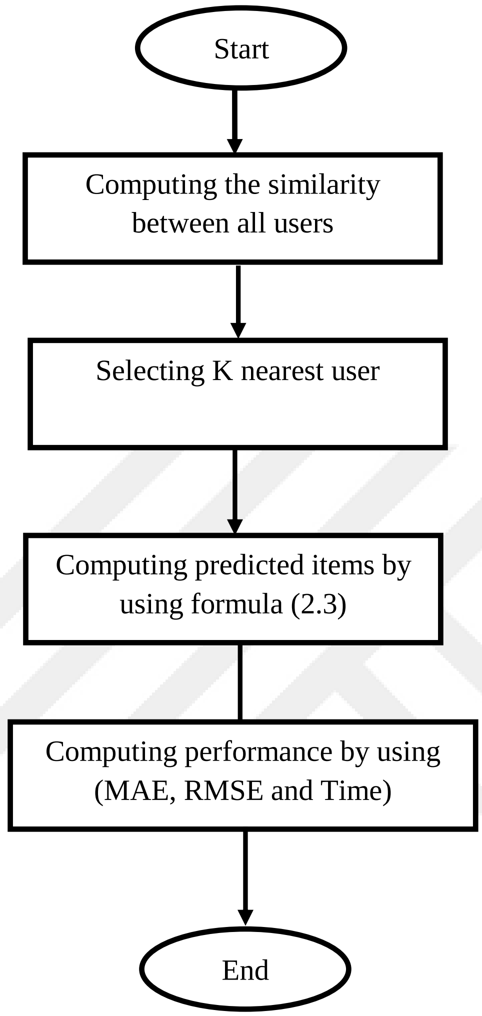

Figure 3.2: Model 1 applying KNN

Firstly, the dataset is read, and then it is prepared and stored in a matrix. The rows refer to the users, columns denote the items and each cell is user rating from (1 to 5) or (-10 to +10) depending on the dataset. However, some cells could be zero, which refer to the unrated items. The dataset divided into two parts training and testing 80% and 20% respectively, each of them stored in different matrices.

After preparing the dataset, KNN starts to work. At the beginning, the similarity between all users in the dataset is computed, using Pearson Correlation method, which is the successful method in matrix similarities [15]. The results of the similarity stored in another two-dimensional matrix. Additionally, the nearest *k* neighborhoods of all users computed depending on the results that computed by using similarity process. Scan the matrix of the dataset to find the items that do not rate by the user and rated by his/her *k* neighborhoods, then computed the rated for those items by calculating the average of the *k* neighborhoods of the user and store the results in a new matrix. Finally, Main Square Error and Root Main Sequence Error that implanted between the final rated matrix and testing matrix (second part of the original dataset) count the performance. The steps are summarized the KNN algorithm in the following table.

Step1: read dataset
Step2: prepare dataset
Step 3: compute the similarity by applying Pearson Correlation
Step 4: selected *K* nearest user
Step 5: compute prediction item of the user by calculating the average of nearest neighborhoods
Step 6: compute performance by using MAE, RMSE and Time

3.6 MODEL TWO APPLYING SVD ALGORITHM

Since the development of the technology, the propagation of recommender systems has become more popular and used by many companies. In addition, the size of the data is huge. Nowadays, the need has become urgent to develop a method has an ability to deal with these extreme conditions. SVD method can accommodate with the huge dataset and the sparsity that founded in the most recent dataset. Figure 3 depicts the SVD method.

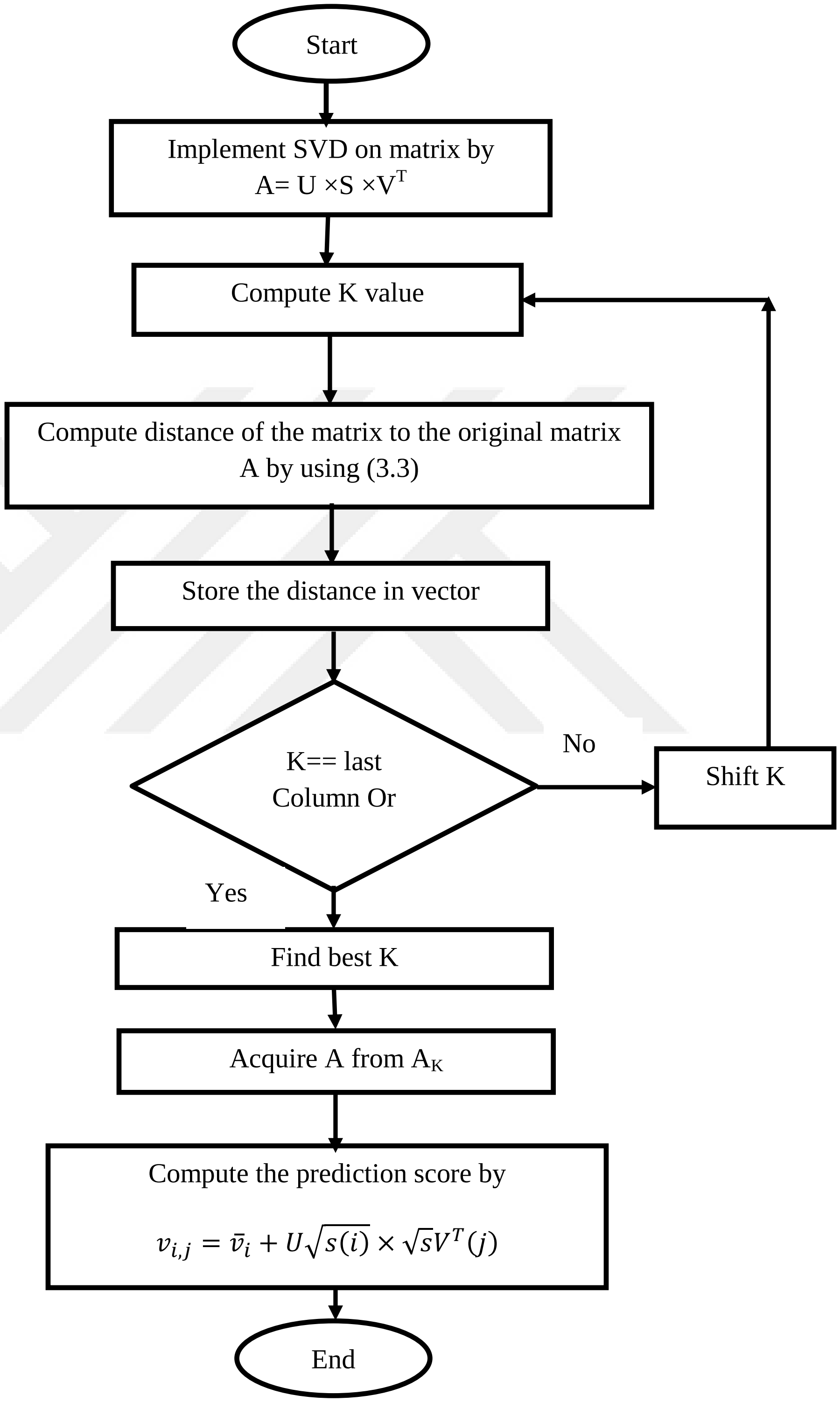

Figure 3.3: Model 2 applying SVD

The algorithm, started by applying formula (2.6) to compute three matrices U, S, and V, the details of these identifiers explained in deep in chapter two. Then, the value of *k* is computed (in this study two values are considered ask which are 10 and 30 for movie lenses and Jester datasets, whereas 5 and 10 in YOW dataset). The searching for the best *k* value is started from the beginning until the end of the matrix *S* with many loops by shifting the S_k (square matrix that generated in each time of the loop with size *k***k*) matrix by one row and column. Then, we compute the distance between the new matrix and original one, the distance is stored in a vector at the end of the procedure. The best value of *k* is selected by using the closer distance to the original matrix, which denotes to the new size of the matrix *S* (latent semantic relation). Then the new matrix is calculated by the formula (3.1) with a proposed value of *k*. Here, the SVD algorithm ignored the unimportant data in the original matrix by computing *k* (latent semantic relation) and generated a new matrix with the important data that calculated by *k*.

$$A_K = U_K\, S_K\, V_K^{\ T} \tag{3.3}$$

Finally, with formula (2.7) computed the prediction score, which is the prediction of the items that calculated by formula 3.3. We summarized the previous explaining by the following steps.

Step 1: Compute the matrix *A* give the dimension of matrix (lets Y m×n) by using formula 2.6, to determine U, S, and V

Step 2: Calculate new matrix depending on *k* by formula 3.3

Step 3: Find the best *k* by computing the distance between the original matrix and new matrix, and store the distance in a vector.

Step 4: Shift *k* by one column m and row until n.a

Step 5: Check whether *k* =n end of metrics, if **yes** go to step 6, otherwise go to step 2

Step 6: Select the minimum distance, to choose best *k*

Step 7: Compute prediction score for items by formula (2.7)

3.7 MODEL THREE APPLY NAÏVE BAYESIAN ALGORITHM

It is one of the most commonly used methods in the classification; method finds the appropriate probability for the user to facilitate the selection of user requests. Figure 4 shows how that method works.

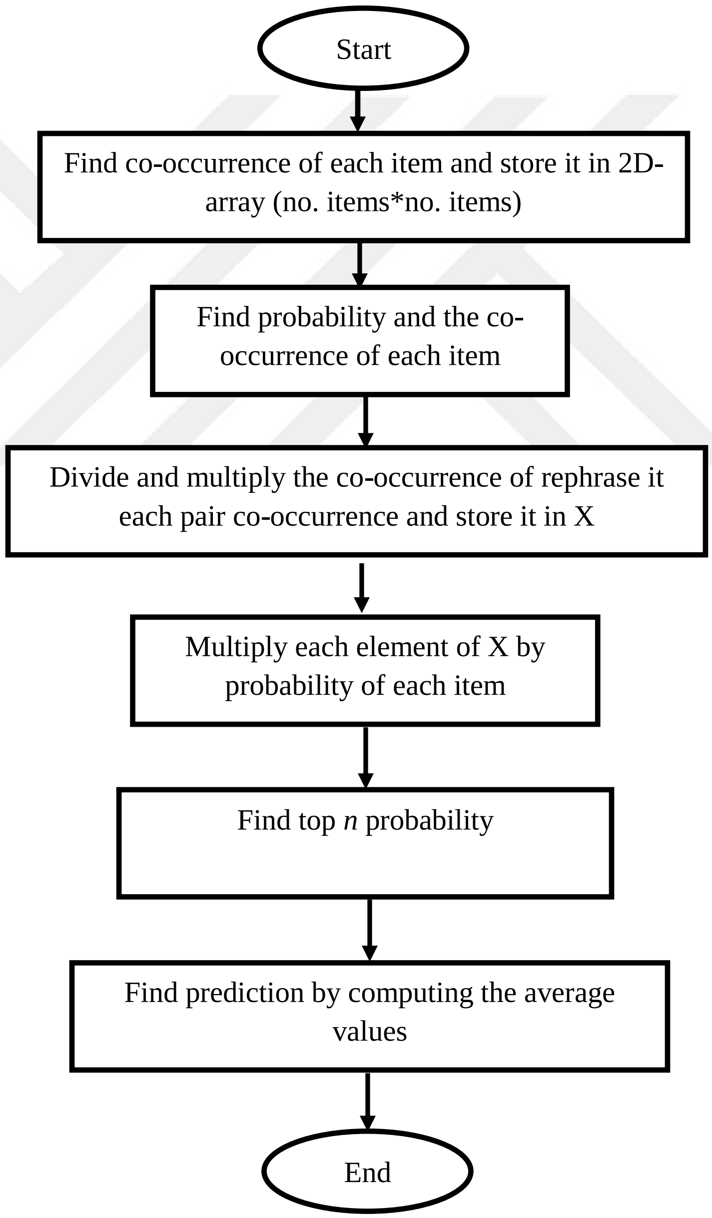

Figure 3.4: Model 3 apply naïve Bayesian algorithm

Step 1: Compute the co-occurrence of each pair for all items, store it in matrix (no. items * no. item)

Step 2: Calculate the co-occurrence of each item by finding the summation of each row from the previous matrix, and then find the probability of each item by dividing the co-occurrence result by the total summation of the matrix of step 1.

Step 3: For all non-rated items divide the co-occurrence of these items by co-occurrence of the not rated item, and then multiply the result with all non-rated items

Step 4: Find the top *n* probability items are computed

Step 5: The average of rated users for those rated the item is computed

As we have seen in the Figure above, the co-occurrence of items is computed by counting the number of repeated each pair for all items. Then, the summation of this array is calculated. The probability of each item is computed by dividing the summation of each row in matrix co-occurrence on the total summation of whole co-occurrence matrix, and then the co-occurrence of each item is counted by finding the summation of each row of the co-occurrence matrix. Additionally, for the items that do not rate previously, this algorithm computed the co-occurrence of these items with the items those rated before. For example, the user that did not rate item 4, but he rated item 1 and 2, thus the algorithm found the co-occurrence for the pairs (1, 4) and (2, 4) and multiply the values of these pairs after dividing it by summation of each row of co-occurrence matrix. For instance, divide the co-occurrence of pair (1,4) by co-occurrence of item 4 (since item 4 did not rate by the user), then repeat this process for all pairs with item 4. Finally yet importantly, the algorithm computes the probability of the items by multiplying the previous result with a probability of items 4, which computed in previous steps.

The previous processes are done to find the probability for all the items those did not rated by the certain users. Now the algorithm starts to find the prediction rated depending on the computed probability. Therefore, the top *n* probability items are computed. Finally, the average of rated users for those rated the item is computed to find the predicted item.

3.8 SUMMARY

In this chapter, we discussed the process of this study, which includes three models. The first model is K-NN algorithm is explained, this algorithm works by finding the similar users then predicts items of the nearest users to the active user. Secondly, model two is applied by using SVD algorithm to exploit the latent relation. Finally, model three is performed using Naïve Bayesian technique to find the probability of the occurrence for all items, and then computed the prediction for non-rated items.

4. RESULTS AND DISSCUSION

4.1 INTRODUCTIONS

The aim of this study is to measure the performance of accuracy and to compare the performance of three different methods (KNN, Naïve Bayesian, and SVD). In this chapter, the experimental results, which depend on three matrices MAE, RMSE and time, will be discussed. The first tow matrices evaluated the accuracy of recommendation items when the value closes to zero, that means high accuracy recommendation and vice versa. The third metric computes the execution time of implementing the models. The recommendation models have been performed in three standard datasets for different categories (Movie Leans, YOW, and Jester).

4.2 DATASET

The dataset used in this study is explained briefly. These data included three different standard datasets with different sizes. These data include Movie Leans, YOW, and Jester. In addition, these datasets contain different categories of items and help to obtain precise results for evaluating the recommendation systems. Usually, these datasets are used in most studies of recommendation systems.

The evaluation process for accuracy is applied by computing two metrics MAE and RMSE; also, the time factor is computed to check the delay time of each algorithm. As it is known in the machine learning techniques, the dataset divided into two parts training and testing. The testing sample is part of the original dataset that selected randomly, which used to compare the result of the certain algorithm to evaluate the performance of an algorithm. To get accurate results, two sizes of training and testing are used 80% and

60% for training against 20% and 40% for testing. The following subsections explain the properties and results for each dataset.

4.2.1 Movie Leans

Movie leans dataset is movie recommender system led by the Group Lens Research. There are many versions of this dataset, where it collected over various periods. The dataset that used in this study consists of 100,000 rating, which rated by 943 users and 1682 items. This dataset has medium density since it has accepted a number of ratings with a number of users and items that mentioned before. The rating value starts from one to five, one means the user did not interest of the movie and five refers to the interesting.

Table 4.1: Result of Movie Leans Dataset in 20% testing and 80% training

Algorithms	SVD		KNN	Naïve Bayesian
Metrics	**K=10**	**K=30**		
MAE	**0.90**	**0.89**	**1.99**	**0.57**
RMSE	**1.03**	**1.02**	**1.41**	**0.79**
Time in seconds	**1910.78**	**1720.87**	**64.87**	**91.4**

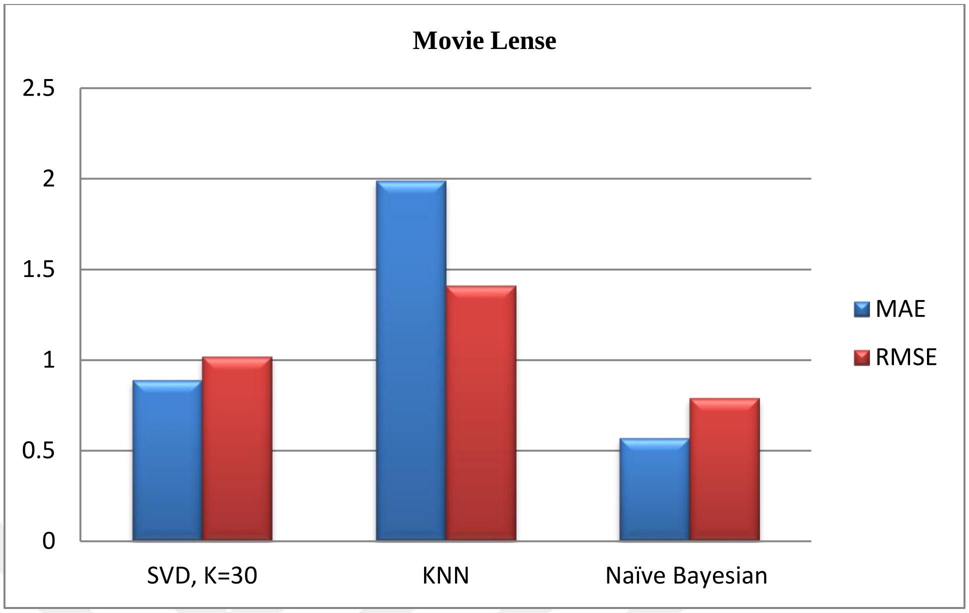

Figure 4.1: Result of Movie Leans dataset (20% testing 80% training)

Table 4.2: Result of Movie Leans Dataset in 40% testing and 60% training

Algorithms	SVD		KNN	Naïve Bayesian
Metrics	**K=10**	**K=30**		
MAE	**0.77**	**0.75**	**2.29**	**0.60**
RMSE	**0.96**	**0.95**	**1.51**	**0.81**
Time in seconds	**2131.93**	**1717.60**	**70.66**	**106.74**

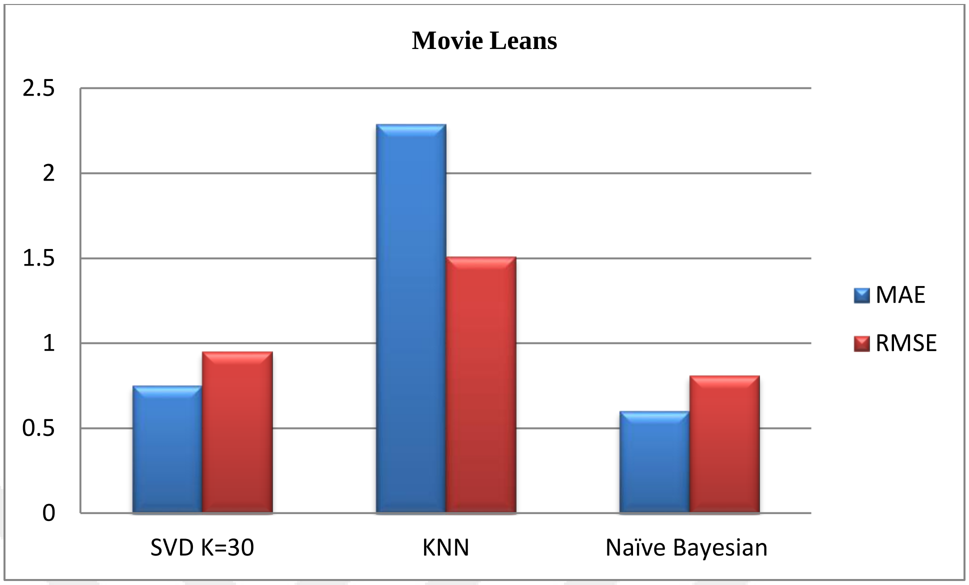

Figure 4.2: Result of Movie Leans dataset (40% testing 60% training)

As shown in Table (4.1) and Figure (4.1) which is applied with 80% of training and 20% of testing. Likewise, in Table (4.2) and Figure (4.2) which are employed with 60% of training and 40% of testing. In both sizes of training, three different algorithms (SVD, KNN and Naïve Bayesian) are evaluated by three metrics (MAE, RMSE and time). The results have revealed that Naïve Bayesian beat on both SVD and KNN for both 80% and 60% of training, where the MAE is 0.57 and the closest value is 0.89 for the SVD algorithm when k=30 when the training 80%. Similarly when the training 60%, the MAE equals 0.60 for Naïve Bayesian and 0.75 for SVD when k=30. In addition, KNN has achieved the worst result, where the result is 1.99 in term of MAE. Likewise, the results do not give us new addition when RMSE is applied, where Naïve Bayesian also give the best result which is 0.79 versus 1.02 for SVD k=30 and 1.41 for KNN when the training 80%, and when the training is 60%, the results are 0.81, 0.95 and 1.51 for Naïve Bayesian, SVD (k=30) and KNN respectively. However, in term of time Naïve Bayesian does not obtain the best result where it took a long time compared with KNN method. The result for Naïve Bayesian is 91.4 seconds, whereas KNN achieved around 65 seconds, (since Naïve Bayesian computes the probability of occurrence of items, which is done by construct two-dimensional size of matrix (items*items), and because the number of items is rather high (1682), so it needs long time to complete the whole process). SVD method for both values of *k* was the worst since this method required a long time to find the closest value of the matrix that is more similar to the original matrix and contain the lattice matrix analysis.

In spite of the results for both training (80% and 60%) assert that the sequence of the best results of used algorithms is Naïve Bayesian, SVD, and KNN, there is a contrast of the result of each algorithm with its own when the training 80% and 60%. For example, Naïve Bayesian with 80% training, the MAE value is 0.57, whereas it is 0.60 with 60% training. This variance takes place since in probability methods have high accuracy results when the training increased. Furthermore, KNN got the worst result when the training became 60% since less value of data computed the similarities of users. On the other hand, the results of SVD improved for both values of *k* when the training 60%. The reason for this development that when the sparsity increased (since when the training 60%, that means 40% of the whole dataset transferred to the testing part), SVD works more efficiently.

4.2.2 Jester

Jester dataset is an online joke recommender system by Ken Goldberg from UC Berkeley. There are many versions of this dataset, in this study, 16,000 different users that selected randomly of 32,500 users, and 100 jokes. The rating value is between +10.0 to -10.0, when the rating value equals +10.0 that means the user interested in the joke, and vice versa when the value -10.0. This dataset has high-density data since most users rated most jokes.

Table 4.3: Result of Jester Dataset in 20% testing 80% training

Algorithms	SVD		KNN	Naïve Bayesian
Metrics	**K=10**	**K=30**		
MAE	**5.0**	**3.79**	**1.24**	**0.60**
RMSE	**2.3**	**2.08**	**1.12**	**0.80**
Time in seconds	**2830**	**2248**	**3065**	**102.04**

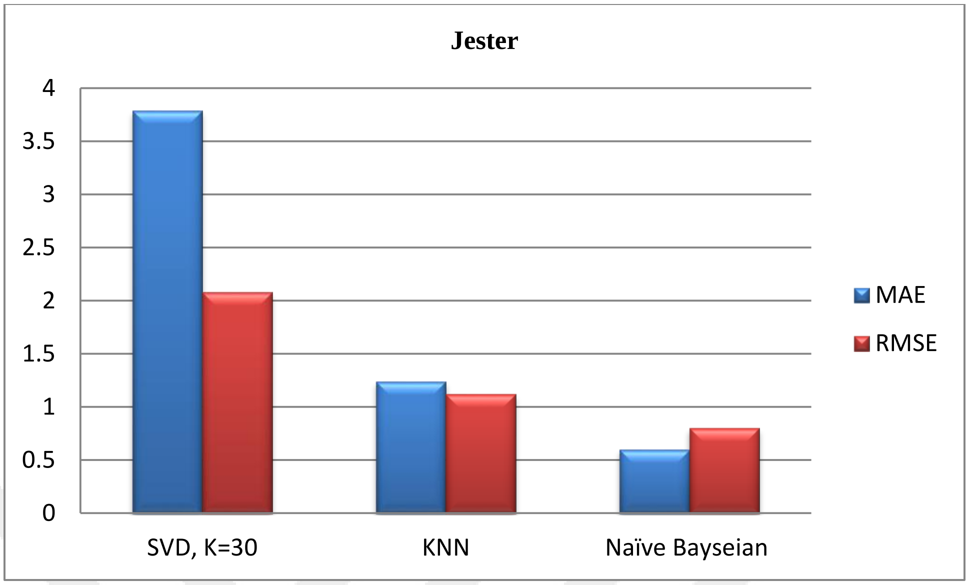

Figure 4.3: Result of Jester dataset (20% testing 80% training)

Table 4.4: Result of Jester Dataset in 40% testing 60% training

Algorithms	SVD		KNN	Naïve Bayesian
Metrics	**K=10**	**K=30**		
MAE	**3.15**	**1.01**	**2.23**	**0.61**
RMSE	**1.86**	**1.04**	**1.94**	**0.80**
Time in seconds	**2592**	**2074**	**3046**	**105.4**

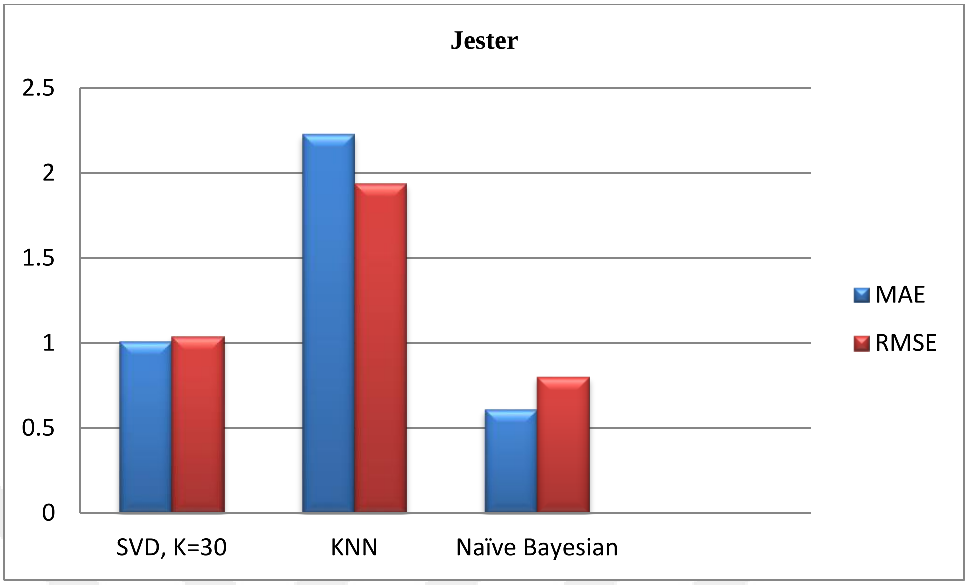

Figure 4.4: Result of Jester dataset (40% testing 60% training)

Table 4.3 with Figure 4.3 and Table 4.4 with Figure 4.4, the training applied with 80% and 60% respectively. Three algorithms used in this study namely, Naïve Bayesian, KNN and SVD, with three metrics MAE, RMSE and time. The results depict the Naïve Bayesian is the ideal method since the results for the three metrics are the best, where MAE is less than one in both pieces of training. Whereas, the closest value of MAE to Naïve Bayesian is KNN (1.24) when the training 80% and SVD (1.01) when the training 60%. The same issue repeated for metric RMSE. In term of time, the difference between Naïve Bayesian and others is large, which is less than 106 seconds for both of training. On the other hand, the time is more than 2200 seconds of SVD and KNN for both pieces of training. This huge difference of the execution time between Naïve Bayesian and both SVD with KNN takes place because the technique of computed the probability of Naïve Bayesian method depends on computation the occurrence of items and these items are just 100, therefore the execution time of this method is short. On the contrary, SVD and KNN have taken a long time to complete their procedures, which the process finds the similarity of 16,000 users for KNN and seek the best value of *k* of matrix 16,000 * 100 for SVD.

Another issue can be found in Jester results, which are the varying of the results for SVD and KNN when the training 80% and 60%. As we have seen in Tables 4.3 and 4.4, the results of KNN for MAE increased (became less accuracy) from 1.24 when the training 80% to 2.23 when the training 60%. This difference since when the training increased the KNN computes the similarity of the users will be more accurate. Conversely, the results of SVD method became high accuracy when the training 60%. The reason for this result is that the SVD achieves high precision when the dataset has sparsity rather than density. As mention before, Jester dataset has high density thus the results of SVD was not good, but when the implementation is applied to 60% training, the results are somewhat improved. In fact, the 60% of training means 40% of the whole dataset converted to the testing part, so the sparsity will increase.

4.2.3 YOW

The last dataset that used in this study is YOW, which collected at the Carnegie Mellon University for the Yow-now news filtering system. This dataset has low density since it has few ratings reaches to 7000 of 5921 articles and 25 users. That means the dataset has high sparsity (most items did not rate). The rating value collected from (1 to 5), one means the user did not interest and five indicates to the interesting for the article.

Table 4.5: Result of YOW Dataset in 20% testing 80% training

Algorithms	SVD		KNN	Naïve Bayesian
Metrics	**K=5**	**K=10**		
MAE	**1.69**	**1.45**	**1.75**	**0.37**
RMSE	**1.25**	**1.21**	**1.34**	**0.61**
Time in seconds	**22.88**	**22.20**	**20.23**	**41.09**

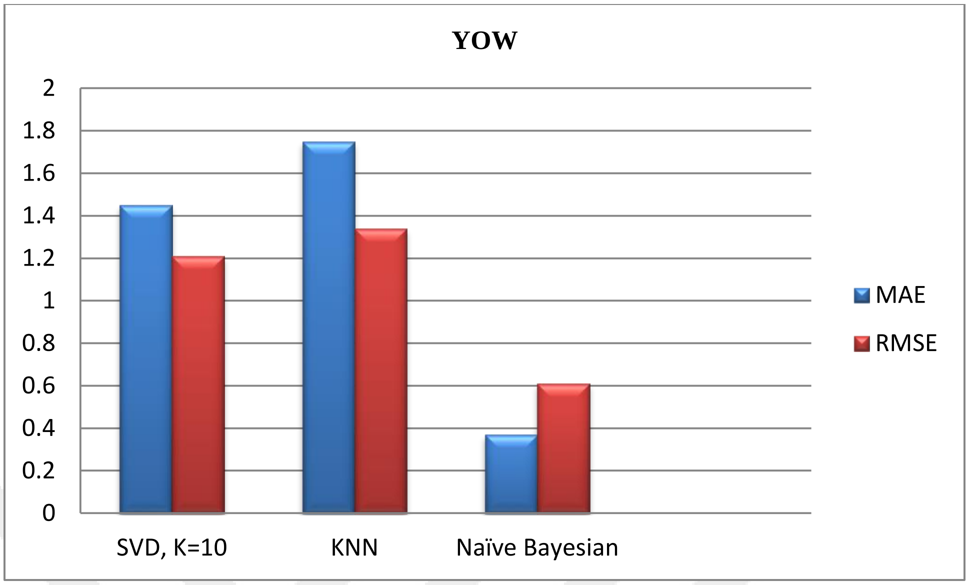

Figure 4.5: Result of YOW dataset (20% testing and 80% training)

Table 4.6: Result of YOW Dataset in 40% testing 60% training

Algorithms	SVD		KNN	Naïve Bayesian
Metrics	**K=5**	**K=10**		
MAE	**2.59**	**2.69**	**2.70**	**0.70**
RMSE	**1.49**	**1.57**	**1.78**	**0.83**
Time in seconds	**20.56**	**20.5**	**20.24**	**40.75**

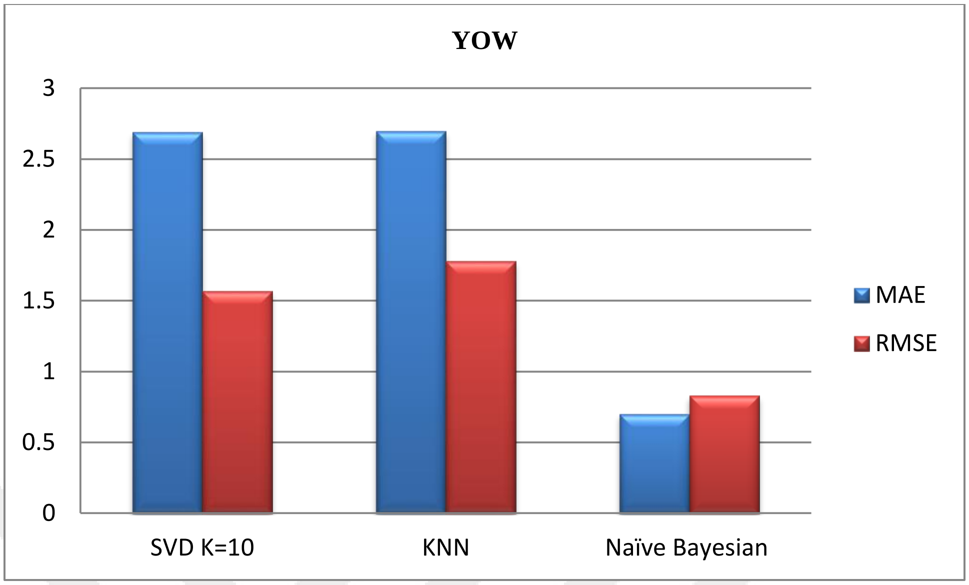

Figure 4.6: Result of YOW dataset (40% testing and 60% training)

Table (4.5) with Figure (4.5) and Table (4.6) with Figure 4.6 depict the result of the three methods (SVD, KNN, and Naïve Bayesian) with training and testing are 80%, 20% and 60%, 40% respectively. The performance evaluated by using MAE and RMSE metrics. In this dataset, the SVD method applied with two values of *K* equals 5 and 10 (since a number of users are 25, so cannot increase the value of *k* more than 20). In fact, when the training is 80%, SVD method with two values the *k* is better than KNN method where the worse value of MAE equals 1.69 when *K* equals five versus 1.75 with KNN, and the result of RMSE is 1.25 with the same condition for SVD method against 1.34 for KNN. This preference of SVD came because the dataset contains high sparsity and that led to the superiority of SVD on KNN method. Nevertheless, again Naïve Bayesian has achieved the best results in both MAE and RSME for both training types, where the value of MAE and RMSE are 0.37 and 0.61 respectively in 80% training and 0.70 of MAE with 0.83 of RMSE in 60% training.

Additionally, in term of time, KNN method and SVD obtained the best result compared to Naïve Bayesian for both type of training, where the value of KNN and Naïve Bayesian is 20.23 and 41.09 seconds respectively. This is fact due to that the number of users in this dataset is 25 and KNN method computes the similarity of the users by constructing two-dimensional size of the matrix (user*user), thus the execution of this method is short.

Whereas Naïve Bayesian calculates the probability of items occurrence also by two-dimensional size matrix, but here (item*item) and the number of items in this dataset is 5961. Moreover, SVD achieved a close result to KNN, which equals to 22.20 seconds, because the number of users is 25, thus the algorithm did not need a long time to find latent semantic analysis.

4.3 SUMMARY

In this chapter, the results of the implementation showed. At the beginning, the datasets (Movie leans, Jester and YOW) that used in this study is explained and the results are implemented by applying two types of training and testing. The common metrics in recommendation system are used which are MAE and RMSE. In addition, the time factor added in this study as an additional metric. All these metrics are used to evaluate the performance of three different algorithms Naïve Bayesian, SVD and KNN. Finally, the two types of implementations gave us a clear view of the performance of the algorithms and the best results of all implementations for both training types was Naïve Bayesian.

5. CONCLUSION AND FUTURE WORK

5.1 INTRODUCTION

Recommender Systems (RSs) stand for as technique can recommend items to the active users that might be interested of his\her (Semeraro et al. 2009) [6]. The common challenges of recommendation system are cold start and sparsity. This work is carried out to compare the performance of retrieval accuracy on different datasets by applying three vary algorithms.

5.2 SUMMARY

This research has executed by utilizing the methodology that explained in chapter three, and discussed in chapter four. The following steps summarized the implementation process of this work.

- Gathering and preparing the datasets to be fit in the used models
- Partitioning the dataset into two groups of training and testing, the first group is 80% for training and 20% for testing; the other group is 60% and 40% or training and testing respectively.
- Applying the proposed method that involves three algorithms, which are KNN, SVD and Naïve Bayesian
- Applying MAE, RMSE and time, achieves evaluation process.

5.3 FUTURE WORK

Knowledge does not stop always in progress. The recommendation system is flexible topic, many works can add to enhance the performance, and thus the following affairs are proposed as future work.

1- Applying different datasets to check the performance of the algorithms
2- Apply different techniques and combine between techniques
3- Try to stratify other measures as well as MAE and RMSE

REFERENCES

[1] J. Pijitra, Book Recommendation System for Digital Library Based on User Profiles by Using Association Rule, Suan Sunandha Rajabhat University Bangkok, Thailand IEEE, 2014.

[2] J. L. Anand Rajaraman, & Ullman, and J.D., Mining of Massive Datasets, Cambridge University Press, 2014.

[3] S. Vinjamuri, M. Sunitha Reddy, Music Recommendation System Using Association Rules, international journal of technology enhancements and emerging engineering research, vol 2, issue 7- 33, 2014.

[4] Introduction to recommender systems handbook, F. Ricci, L. Rokach & B. Shapira. , 2011 (pp. 1-35): Springer US.

[5] Content-based recommender systems: State of the art and trends Recommender systems handbook, P. Lops, M. De Gemmis, & G. Semeraro. (2011). (pp. 73-105): Springer.

[6] G. Semeraro, et al., Knowledge Infusion into Content-based Recommender Systems. Proceedings of the ACM Conference on Recommender Systems, 2009 pp.301 304.

[7] X. Yang, Y. Guo, & Y. Liu, Bayesian-Inference-Based Recommendation in Online Social Networks. Parallel and Distributed Systems, IEEE Transactions on, 24(4), (2013), 642-651.

[8] L. H. Son, Dealing with the new user cold-start problem in recommender systems: A comparative review. Information Systems, (2016) 58, pp. 87-104.

[9] H. Wang, N. Wang & D. Yeung. Collaborative deep learning for recommender systems. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 10-13 August. Sydney, NSW, Australia, (2015), pp. 1235-1244.

[10] G. Adomavicius, & A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. Knowledge and Data Engineering, IEEE Transactions on, (2005), 17(6), 734-749.

[11] R. Bell, Y. Koren, & C. Volinsky,. Modeling relationships at multiple scales to improve accuracy of large recommender systems. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 12 - 15 August. San Jose, California, USA, (2007) (pp. 95-104).

[12] E. Viennet. Collaborative filtering in social networks: A community-based approach. Proceedings of the International Conference on Computing, Management and Telecommunications (ComManTel).21-24 January. Ho Chi Minh City, Vietnam, (2013), (pp. 128-133).

[13] J. S. Breese, D. Heckerman, & C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering. Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence. (1998), (pp. 43-52).

[14] R. Jin, J. Y. Chai, & L. Si, An automatic weighting scheme for collaborative filtering. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. 25- 29 July. Sheffield, United Kingdom, (2004) (pp. 337-344).

[15] J. Yang, J. Kim, W. Kim, & Y. H. Kim, Measuring user similarity using electric circuit analysis: Application to collaborative filtering. (2012), PLoS ONE, 7(11), e49126.

[16] M. Deshpande & G. Karypis, Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS), (2004), 22(1), 143-177.

[17] B. Sarwar, G. Karypis, J. Konstan, & J. Riedl. Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web. 01-05 May. Hong Kong (2001), (pp. 285-295).

[18] T. Hofmann, Collaborative filtering via gaussian probabilistic latent semantic analysis. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. 28 July -01 August. Toronto, Canada, (2003). (pp. 259-266).

[19] R. Salakhutdinov, & A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. Proceedings of the 25th international conference on Machine learning. 05-09 July. Helsinki, Finland, (2008), (pp. 880- 887).

[20] J. Canny, Collaborative filtering with privacy. Proceedings 2002 IEEE Symposium on Security and Privacy. 12-15 May. Berkeley, CA, USA, (2002) (pp. 45-57).

[21] R. Salakhutdinov, & A. Mnih, Probabilistic Matrix Factorization. Proceedings of the 20th International Conference on Neural Information Processing Systems. 03-06 December. Vancouver, British Columbia, Canada, (2007), (pp. 1257-1264).

[22] B. Sarwar, G.KaKarypis, J. Konstan and J. Reidl, "Application of Dimmensionality Reduction in Recommender Systems: A Case Study," in proceeding of the WebKDD workshop at the ACM SigKDD,2000.

[23] Recommender systems, Ricci, F., Kantor, P. B., Rokach, L., and Shapira, B., 2011, handbook, Springer.

[24] K. S. Roshni, N. D. Sachin. Comparative Study on Approaches of Recommendation System. International Journal of Computer Applications, (2, May), 2015, (0975 – 8887) Volume 118 – No.

[25] A. H. Mohd, A. Omar, S. Ramachandram. Collaborative Filtering Based Recommendation System: A survey. International Journal on Computer Science and Engineering (IJCSE). 05 May, (2012), ISSN: 0975-3397 Vol. 4 No..

[26] P. H. Aditya, I. Budi, Q. Munajat. A Comparative Analysis of Memory-based and Model-based Collaborative Filtering on the Implementation of Recommender System for Ecommerce in Indonesi : A Case Study PT X. IEEE, (2016).

[27] R. M. Bell, Y. Koren, C. Volinsky. All Together Now: A Perspective on the Netflix Prize. CHANCE, Volume 23, Issue 1, (2010).

[28] P. D. Kumari, N. S. Reddy. Survey on Item Based Collaborative Filtering Algorithm for Data Processing and Mining of Patterns in Mobile Commerce. International Journal of Computer Science and Management Research, Volume 2, Issue 7, (2013).

[29] M. D. Ekstrand, J. T. Riedl, & J. A. Konstan, Collaborative Filtering Recommender Systems. Foundations and Trends® in Human– Computer Interaction, 4(2), (2010), pp.81–173.

[30] M. W. Berry, S. T. Dumais, and G. W. O'Brian. "Using Linear Algebra for Intelligent Information Retrieval". SIAM Review, 37(4), (1995), pp. 573-595.

[31] D. Billsus, and M. J. Pazzani. “Learning Collaborative Information Filters”. In Proceedings of Recommender Systems Workshop. Tech. Report WS-98-08, AAAI Press 1998.

[32] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. “Indexing by Latent Semantic Analysis”. Journal of the American Society for Information Science, 41(6), (1990), pp. 391-407.

[33] L. Terveen, W. Hill, B. Ammento, D. McDonald, and 1.Creter, "Phoaks: A systems for sharing recommendations," Communication of the ACM, Vol. 40, No. 3, (1997), pp. 59-62.

[34] K. Goldberg, T. Roeder, D. Gupta and C. Perkins, "Eigenstate: A Constant Time Collaborative Filtering Algorithm," Information Retrieval, Vol. 4, No.2, (2001) pp. 133-151.

[35] Y. Koren, R. Bell and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," Computer in Computer, Vol. 42, No. 8, (2009), pp. 30-37.

[36] A. Paterek, "Improving regularized Singular Value Decomposition for Collaborative Filtering," in Proceeding of KDD cup Workshop at SIGKDD'07, 13th ACM International Conference on Knowledge Discovery and Dataminig, (2007), pp. 39-42.

[37] T. Hofmann, "Latent Semantic Models for Collaborative Filtering," ACM Transaction on Information Systems (TOIS), (2004), Vol. 22, No. I, pp. 89-115.

[38] R. Salakhutdinov and A. Mnith, "Probabilistic Matrix Factorization," in J. C. Platt, D. Koller, Y. Singer and S. Roweis, editors, Advances in Neural Infonnation Processing Systems20, Combridge, MA, MIT Press, (2008).

[39] K. Miyahara and M. J. Pazzani, “Improvement of collaborative filtering with the simple Bayesian classifier,” Information Processing Society of Japan, vol. 43, no. 11, (2002).

[40] D. Debashis, S. Laxman and D. Sujoy. “A Survey on Recommendation System”, International Journal of Computer Applications (0975 – 8887) Volume 160 – No 7, February (2017).

[41] D. Jannach, M. Zanker, A. Felfernig, and G. Friedrich. Recommender systems: an introduction. Cambridge University Press, (2011).

[42] X. Yang, Collaborative Filtering Based Social Recommender Systems. PhD. NEW YORK UNIVERSITY, 2014.

[43] P. Resnick & H. R. Varian. Recommender Systems Communications of the ACM, 40(3), 56-58, (1997).

[44] How to get the Recommender out of the Lab? Recommender Systems Handbook J. Picault, M. Ribiere, D. Bonnefoy & K. Mercer, (2011) (pp. 333-365): Springer.

[45] N. R. Paul, G. Karpagam, R. Sindhuja, & S. Yamini, A Ranking Based Combined Approach for Recommending Travel Packages. International Journal of Scientific Research in Science, Engineering and Technology, (2015), (1)2, 460-466.

[46] J. Schafer, et al.., Collaborative Filtering Recommender Systems. The Adaptive Web, (2007), pp.291–324.

[47] G. Linden, B. Smith, & J. York, Amazon. com recommendations: Item-to item collaborative filtering. Internet Computing, IEEE, (2003), 7(1), 76-80.

[48] Y. Koren,. Factorization meets the neighborhood: a multifaceted collaborative filtering model. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. Las Vegas, Nevada, USA, 24-27 August (2008), (pp. 426-434).

[49] B. N. Miller, J. A. Konstan, & J. Riedl. PocketLens: Toward a personal recommender system. ACM Transactions on Information Systems (TOIS), (2004) 22(3), 437-476.

[50] D. B. Hauver, & J. C. French,. Flycasting: using collaborative filtering to generate a playlist for online radio. Proceedings of the First International Conference on WEB Delivering of Music. 23-24 November. Florence, Italy, (2001), (pp. 123-130).

[51] C. Hayes & P. Cunningham. Smart radio: community based music radio. Knowledge-Based Systems, (2001), 14(3), 197-201.

[52] J. Lu, D. Wu, M. Mao, W. Wang & G. Zhang. Recommender system application developments: a survey. Decision support systems, (2015), 74, 12-32.

[53] A. Bellogín, P. Castells, & I. Cantador. Improving memory-based collaborative filtering by neighbor selection based on user preference overlap. Proceedings of the 10th Conference on Open Research Areas in Information Retrieval. 15 - 17 May. Lisbon, Portugal, (2013), (pp. 145-148).

[54] K. Al Falahi, N. Mavridis, & Y. Atif. Social networks and recommender systems: a world of current and future synergies. Computational Social Networks (pp. 445-465): Springer (2012).

[55] C. Basu, H. Hirsh, & W. Cohen. Recommendation as classification: Using social and content-based information in recommendation. Proceedings of the Fifteenth National Conference on Artificial Intelligence. (pp. 714-720) (1998).

[56] S. Xiaohua, K. Fansheng & Y. Song. A comparison of several algorithms for collaborative filtering in startup stage. In Proceedings. IEEE Networking, Sensing and Control, 2005. pp. 25–28.

[57] P. Symeonidis, A. Nanopoulos & Y. Manolopoulos. MoviExplain: A Recommender System with Explanations. Proceedings of the third ACM conference on Recommender systems - RecSys ’09, (2009) pp.317–320.

[58] C. A. Lumbreras. Towards trust-aware recommendations in social networks. Master Thesis, Polytechnic University of Catalonia, Spain (2012).

[59] Data mining methods for recommender systems Recommender Systems Handbook. X. Amatriain, A. Jaimes, N. Oliver & J. M. Pujol, (pp. 39-71): Springer (2011).

[60] J. Bobadilla, et al., A framework for collaborative filtering recommender systems. Expert Systems with Applications, 38(12), (2011), pp.14609–14623.

[61] J. Bobadilla, et al., Recommender systems survey. Knowledge-Based Systems, 46, (2013), pp.109–132.

[62] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom & J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. Proceedings of the ACM conference on Computer supported cooperative work. 22-26 October. Chapel Hill, North Carolina, USA, (1994), (pp. 175-186).

[63] J. L. Herlocker, J. A. Konstan, A. Borchers, & J. Riedl. An algorithmic framework for performing collaborative filtering. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. 15-19 August. Berkeley, California, USA, (1999), (pp. 230-237). ACM.

[64] H. Ma, C. Liu, I. King & M. R. Lyu. Probabilistic factor models for website recommendation. Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. 24-28 July. Beijing, China, (2011), (pp. 265 274).

[65] L. H. Ungar & D. P. Foster. Clustering methods for collaborative filtering. Proceedings of the Workshop on Recommendation Systems. (Vol. 1, pp. 114-129), (1998).

[66] Y. Bergner, S. Droschler, G. Kortemeyer, S. Rayyan, D. Seaton & D. E. Pritchard. Model-Based Collaborative Filtering Analysis of Student Response Data: Machine-Learning Item Response Theory. Proceedings of the International Conference on Educational Data Mining (EDM). 6-9 July. Memphis, TN., USA, (2012), (pp. 95-102).

[67] G. R. Xue, C. Lin, Q. Yang, W. Xi, H. J. Zeng, Y. Yu & Z. Chen. Scalable collaborative filtering using cluster-based smoothing. Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. 15-19 August. Salvador, Brazil, (2005), (pp. 114-121).

[68] A. Kohrs & B. Merialdo. Clustering for collaborative filtering applications. In M. Mohammadian (Ed.) Computational Intelligence for Modelling, Control & Automation (pp. 199-205). Amsterdam: IOS press, (1999).

[69] T. Nathanson, E. Bitton, & K. Goldberg. Eigentaste 5.0: Constant-Time Adaptability in a Recommender System Using Item Clustering. Proceedings of the 2007 ACM conference on Recommender systems. 19-20 October. Minneapolis, MN, USA, (2007), (pp. 149-152).

[70] M. Zhang & N. Hurley. Novel item recommendation by user profile partitioning. Proceedings of the 2009 IEEE/WIC/ACM International Joint

Conference on Web Intelligence and Intelligent Agent Technology-Volume 01. 15-18 September. Washington, DC, USA, (2009), (pp. 508-515).

[71] H. Zhang. The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference FLAIRS 2004, 1(2), (2004) pp.1–6.

[72] D. Lowd & P. Domingos. Naive Bayes models for probability estimation. 22nd international conference on Machine learning, (2005) pp.529–536.

[73] P. N. Tan, M. Steinbach & Kumar. Introduction to data mining. New York: Pearson Education (2006).

[74] N. D. FREITAS. Less Naive Bayes for Recommender Systems. University of Oxford (2013).

[75] L. Lü, M. Medo, C. H. Yeung, Y. C. Zhang, Z. K. Zhang & T. Zhou. Recommender systems. Physics Reports, 519(1), 1-49, (2012).

[76] G. H. Golub & C. Reinsch. Singular value decomposition and least squares solutions. Numerische mathematik, (1970), 14(5), 403-420.

[77] D. M. Blei, A. Y. Ng & M. I. Jordan. Latent dirichlet allocation. Journal of machine Learning research, 3(Jan) (2003), 993-1022.

[78] L. Si & R. Jin. Flexible mixture model for collaborative filtering. Proceedings of the Twentieth International Conference on Machine Learning. Washington, USA, (2003) (pp. 704-711).

[79] M. Jamali & M. A. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. Proceedings of the fourth ACM conference on Recommender systems. 26-30 September. Barcelona, Spain, (2010) (pp. 135-142).

[80] Y. Yu, C. Wang & Y. Gao. Attributes Coupling based Item Enhanced Matrix Factorization Technique for Recommender Systems. arXiv preprint arXiv:1405.0770, (2014).

[81] N. Srebro & T. Jaakkola. Weighted low-rank approximations. Proceedings of the Twentieth International Conference on Machine Learning. 21 August. Washington DC, USA, (Vol. 3, No. 2003, pp. 720-727) (2003).

[82] J. D. Rennie & N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. Proceedings of the 22nd international conference on Machine learning. 07-11 August. Bonn, Germany, (2005) (pp. 713-719).

[83] M.W. BERRY, Large scale singular value computations, Internat. J. Supercomputer Appl., 6 (1992), pp. 13-49.

[84] S. M. Zeinab. New Algorithm for Recommender Systems based on Singular Value Decomposition Method. (2013) (978-1-4799-2093-8/13) IEEE.

[85] P. Sutheera, S. Supitchaya. "Hybrid Naive Bayes Classifier Weighting and Singular Value Decomposition Technique for Recommender System", (IEEE 978-1-4244-9698-3/11), (2011).

[86] M. J. Pazzani & D. Billsus. Content-based recommendation systems. The adaptive web (pp. 325-341): Springer (2007).

[87] M. Szomszor, C. Cattuto, H. O. Alani, K. Hara, A. Baldassarri, V. Loreto & V. D. Servedio. Folksonomies, the semantic web, and movie recommendation, (2007).

[88] M. De Gemmis, P. Lops, G. Semeraro & P. Basile. Integrating tags in a semantic content-based recommender. Proceedings of the ACM conference on Recommender systems. 23-25 October. Lausanne, Switzerland, (2008), (pp. 163-170)

[89] I. Cantador, A. Bellogín & D. Vallet. Content-based recommendation in social tagging systems. Proceedings of the fourth ACM conference on Recommender systems. 26-30 September. Barcelona, Spain, (2010) (pp. 237-240).

[90] J. Diederich & T. Iofciu. Finding communities of practice from user profiles based on folksonomies. Proceedings of the 1st Int Workshop on Building Technology Enhanced Learning solutions for Communities of Practice (TELCoPs' 06). (pp. 288-297) (2006).

[91] T. Cover & P. Hart. Nearest neighbor pattern classification. IEEE transactions on information theory, 13(1), 21-27(1967).

[92] J. R. Quinlan. Induction of decision trees. Machine learning, 1(1), 81-106 (1986).

[93] N. Cristianini & J. Shawe-Taylor. An introduction to support vector machines and other kernel-based learning methods: Cambridge university press (2000).

[94] J. M. Zurada. Introduction to artificial neural systems. West Publishing Co (1992).

[95] P. Melville & V. Sindhwani. Recommender systems Encyclopedia of machine learning (pp. 829-838): Springer (2011).

[96] Y.B. Fernández, J. J. P. Arias, M. L. Nores, A. G. Solla & M. R. Cabrer. AVATAR: an improved solution for personalized TV based on semantic inference. IEEE Transactions on Consumer Electronics, 52(1), 223-231 (2006).

[97] B. Smyth & P. Cotter. A personalized television listings service. Communications of the ACM,(43(8), 107-111) (2000).

[98] O. Celma. Foafing the music: Bridging the semantic gap in music recommendation. Proceedings of the International Semantic Web Conference ISWC 2006. 5-9 November. Athens, GA, USA, (2006) (pp. 927-934).

[99] J. F. McCarthy & T. D. Anagnost. MusicFX: an arbiter of group preferences for computer supported collaborative workouts. Proceedings of the 1998 ACM conference on Computer supported cooperative work. 14-18 November. Seattle, Washington, USA, (1998) (pp. 363-372).

[100] R. Armstrong, D. Freitag, T. Joachims & T. Mitchel. Web watcher: A learning apprentice for the world wide web. Proceedings AAAI Spring symposium on Information gathering from Heterogeneous, distributed environments. (pp. 6-12) (1995).

[101] Z. Chen, X. Meng, B. Zhu & R. H. Fowler. Web sail: From on-line learning to web search. Proceedings of the First International Conference on Web Information Systems Engineering. 19-21 June. Hong Kong, China, (2000) (Vol. 1, pp. 206-213).

[102] B. Mobasher, R. Cooley & J. Srivastava. Automatic personalization based on web usage mining. Communications of the ACM, 43(8), 142-151 (2000).

[103] Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. Proceedings of the fifth ACM conference on Digital libraries. 02-07 June. San Antonio, Texas, USA, (pp. 195-204)

[104] X. Jin, Y. Zhou & B. Mobasher. A maximum entropy web recommendation system: combining collaborative and content features. Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. 21-24 August. Chicago, Illinois, USA, (2005) (pp. 612-617).

[105] A. I. Schein, A. Popescul, L. H. Ungar & D. M. Pennock. Methods and metrics for cold-start recommendations. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. 11-15 August. Tampere, Finland, (2002), (pp. 253-260).

[106] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. Artificial Intelligence Review, 13(5-6), 393-408, (1999).

[107] I. Soboroff & C. Nicholas. Combining content and collaboration in text filtering. Proceedings of the IJCAI' Workshop on Machine Learning for Information Filtering.(Vol. 99, pp. 86-91), (1999).

[108] M. Balabanović & Y. Shoham. Fab: content-based, collaborative recommendation. Communications of the ACM, 40(3), 66-72, (1997).

[109] P. Melville, R. J. Mooney & R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. Proceedings of the Eighteenth national conference on Artificial intelligence. 28 July-01 August. Edmonton, Alberta, Canada, (2002), (pp. 187-192).

[110] J. Salter & N. Antonopoulos. CinemaScreen recommender agent: combining collaborative and content-based filtering. IEEE Intelligent Systems, 21(1), 35-41 (2006).

[111] R. Burke. Hybrid web recommender systems. The adaptive web (pp. 377- 408): Springer, (2007).

[112] X. Su, R. Greiner, T. M. Khoshgoftaar & X. Zhu. Hybrid collaborative filtering algorithms using a mixture of experts. Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence. 02-05 November. Washington, DC, USA, (2007), (pp. 645-649)

[113] I. Schwab, A. Kobsa & I. Koychev. Learning user interests through positive examples using content analysis and collaborative filtering. Internal Memo, GMD, St. Augustin, Germany, (2001).

[114] A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte & A. Peleteiro. A hybrid content-based and itembased collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Information Sciences, 180(22), 4290- 4311, (2010).

[115] G. Guo, J. Zhang & N. Yorke-Smith. A Novel Bayesian Similarity Measure for Recommender Systems. Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI). 3-9 August. Beijing, China, (2013), (pp. 2619-2625).

[116] T. Q. Lee, Y. Park & Y. T. Park. A similarity measure for collaborative filtering with implicit feedback. Proceedings of the 3rd International Conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. 21-24 August. Qingdao, China, (2007), (pp. pp.385-397).

[117] N. Lathia, S. Hailes & L. Capra. Private distributed collaborative filtering using estimated concordance measures. Proceedings of the 2007 ACM conference on Recommender systems. 19-20 October. Minneapolis, MN, USA, (2007), (pp. 1-8).

[118] H. J. X. Ahn. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences, 178(1), 37-51, (2007).

[119] J. Bobadilla, F. Ortega & A. Hernando. A collaborative filtering similarity measure based on singularities. Information Processing & Management, 48(2), 204-217, (2012).

[120] U. Shardanand & P. Maes. Social information filtering: algorithms for automating "word of mouth". Proceedings of the SIGCHI conference on Human factors in computing systems. 07-11 May. Denver, Colorado, USA, (1995), (pp. 210-217).

[121] H. Ma, I. King & M. R. Lyu. Effective missing data prediction for collaborative filtering. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 23- 27 July. Amsterdam, The Netherlands, (2007), (pp. 39-46).

[122] Y. Shi, M. Larson & A. Hanjalic. Exploiting user similarity based on rateditem pools for improved user-based collaborative filtering. Proceedings of the third ACM conference on Recommender systems. 23-25 October. New York, USA, (2009), (pp. 125-132).

[123] H. F. Sun, J. L. Chen, G. Yu, C. C. Liu, Y. Peng, G. Chen & B. Cheng. JacUOD: A new similarity measurement for collaborative filtering. Journal of Computer Science and Technology, 27(6), 1252-1260, (2012).

[124] Z. Sun, L. Han, W. Huang, X. Wang, X. Zeng, M. Wang & H. Yan. Recommender systems based on social networks. Journal of Systems and Software, 99, 109-119, (2015).

[125] Y. Zhang, W. Chen & Z. Yin. Collaborative filtering with social regularization for TV program recommendation. Knowledge-Based Systems, 54, 310-317, (2013).

[126] C. Seung-Seok, C. Sung-Hyuk & C. C. Tapper,. A Survey of Binary Similarity and Distance Measures. Journal of Systemics, Cybernetics & Informatics, 8(1), pp.43–48, (2010).

[127] H. Finch. Comparison of Distance Measures in Cluster Analysis with Dichotomous Data. Journal of Data Science, 3, pp.85–100, (2005).

[128] R. R. Sokal. Principles of numerical taxonomy. Ab&Es, 46(1), p.14, (1982).

[129] D. Anand. Enhancing Accuracy of Recommender Systems through Various Approaches to Local and Global Similarity Measures. PhD Thesis, Jawaharlal Nehru University, New Delhi, (2011).

[130] Z. Huang, H. Chen & D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. ACM Transactions on Information Systems (TOIS), 22(1), 116-142, (2004).

[131] J. Liu, C. Wu & W. Liu. Bayesian Probabilistic Matrix Factorization with Social Relations and Item Contents for recommendation. Decision support systems, 55(3), 838-850, (2013).

[132] N. Srebro, J. Rennie & T. S. Jaakkola. Maximum-margin matrix factorization. Advances in neural information processing systems. (pp. 1329- 1336), (2004).

[133] P. Yu, L. Lin & J. Wang. A novel framework to alleviate the sparsity problem in context-aware recommender systems. New Review of Hypermedia and Multimedia, pp.1-18. (2016).

[134] C. N. Ziegler, G. Lausen & L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. Proceedings of the thirteenth ACM international conference on Information and knowledge management. 08-13 November. New York, NY, USA, (2004), (pp. 406-415).

[135] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez & W. Nzoukou. A survey paper on recommender systems. arXiv preprint arXiv:1006.5278, (2010).

[136] S. E. Middleton, H. Alani & D. C. De Roure. Exploiting synergy between ontologies and recommender systems. Proceedings of the 3rd International Conference on Semantic Web. 7 May. Volume 55, (2002), (pp. 41-50).

[137] R. Cöster, A. Gustavsson, T. Olsson & A. Rudström. Enhancing webbased configuration with recommendations and cluster-based help. Proceedings of the AH'2002 Workshop on Recommendation and Personalization in eCommerce. 28 May. Málaga, Spain, (pp. 30-39), (2002).

[138] X. N. Lam, T. Vu, T. D. Le & A. D. Duong. Addressing cold-start problem in recommendation systems. Proceedings of the 2nd international conference on Ubiquitous information management and communication. 31 January-01 February. Suwon, Korea, (2008), (pp. 208-211).

[139] M. Papagelis, I. Rousidis, D. Plexousakis & E. Theoharopoulos. Incremental collaborative filtering for highly-scalable recommendation algorithms. International Symposium on Methodologies for Intelligent Systems. (pp. 553-561). Springer Berlin Heidelberg, (2005).

[140] J. Wang, A. P. De Vries & M. J. Reinders. Unifying user-based and itembased collaborative filtering approaches by similarity fusion. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. 06-11 August. Seattle, Washington, USA, (2006), (pp. 501-508).

[141] I. S. Altingovde, Ö. N. Subakan & Ö. Ulusoy. Cluster searching strategies for collaborative recommendation systems. Information Processing & Management, 49(3), 688-697, (2013).

[142] T. George & S. Merugu. A scalable collaborative filtering framework based on co-clustering. Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05). 27-30 November. Houston, TX, USA, (2005), (pp. 4-pp).

[143] B. Sarwar, G. Karypis, J. Konstan & J. Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. Proceedings of the Fifth International Conference on Computer and Information Science. (2002), (pp. 27-28).

[144] N. Ramakrishnan, B. J. Keller, B. J Mirza, A. Y. Grama & G. Karypis. Privacy risks in recommender systems. IEEE Internet Computing, 5(6), 54, (2001).

[145] A. Bilge & H. Polat. An improved privacy-preserving DWT-based collaborative filtering scheme. Expert Systems with Applications, 39(3), 3841-3854, (2012).

[146] A. Jeckmans, Q. Tang & P. Hartel. Privacy-preserving collaborative filtering based on horizontally partitioned dataset. Proceedings of the bInternational Conference on Collaboration Technologies and Systems (CTS). 21-25 May. Denver, CO, USA, (2012), (pp. 439-446).

[147] I. Yakut & H. Polat. Arbitrarily distributed data-based recommendations with privacy. Data & Knowledge Engineering, 72, 239-256, (2012).

[148] J. Zhan, C. L. Hsieh, I. C. Wang, T. S. Hsu, C. J. Liau & D. W. Wang. Privacy-preserving collaborative recommender systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40(4), 472-476, (2010).

[149] X. Su & T. M. Khoshgoftaar. A survey of collaborative filtering techniques. Advances in artificial intelligence, p.4, (2009).

[150] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes & M. Sartin. Combining content-based and collaborative filters in an online newspaper. Proceedings of ACM SIGIR workshop on recommender systems: Algorithms and Evaluation. Berkeley, Calif, USA, (Vol. 60), (1999).

[151] M. Ghazanfar & A. Prugel-Bennett. Fulfilling the needs of gray-sheep users in recommender systems, a clustering solution. Proceedings of the International Conference on Information Systems and Computational Intelligence. 18-20 January. China, (2011).

[152] R. Bhaumik, B. Mobasher & R. Burke. A clustering approach to unsupervised attack detection in collaborative recommender systems. Proceedings of the 7th IEEE international conference on data mining. Las Vegas, NV, USA, (2011), (pp. 181-187).

[153] S. K. Lam & J. Riedl. Shilling recommender systems for fun and profit. Proceedings of the 13th international conference on World Wide Web. 17-20 May. New York, NY, USA, (2004), (pp. 393-402).

[154] C. Li & Z. Luo. Detection of shilling attacks in collaborative filtering recommender systems. Proceedings of the International Conference of Soft Computing and Pattern Recognition (SoCPaR). 14-16 October. Dalian, China, (2011), (pp, 190-193).

[155] B. Mehta. Unsupervised shilling detection for collaborative filtering. Proceedings of the 22nd national conference on Artificial intelligence. 22-26 July. Vancouver, British Columbia, Canada, (2007), (pp. 1402-1407)

[156] J. L. Herlocker, J. A. Konstan, L. G. Terveen & J. T. Riedl. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS), 22(1), 5-53, (2004).

[157] X. Ma, H. Lu, Z. Gan & Q. Zhao. An exploration of improving prediction accuracy by constructing a multi-type clustering based recommendation framework. Neurocomputing, 191, 388-397, (2016).

[158] K. Taek-Hun & Y. Sung-Bong. An improved neighbor selection algorithm in collaborative filtering. IEICE transactions on information and systems, 88(5), 1072-1076, (2005).

[159] K. Treerattanapitak & C. Jaruskulchai. Exponential fuzzy C-means for collaborative filtering. Journal of Computer Science and Technology, 27(3), 567-576, (2012).

[160] C. W. Cleverdon & M. Keen. Factors determining the performance of indexing systems. Aslib Cranfield research project. Volume 2, Technical Report, (1966).

[161] G. Karypis. Evaluation of item-based top-n recommendation algorithms. Proceedings of the tenth international conference on Information and knowledge management. 05-10 October. Atlanta, Georgia, USA, (2001), (pp. 247- 254).

[162] C. N. Ziegler & J. Golbeck. Investigating interactions of trust and interest similarity. Decision support systems, 43(2), 460-475, (2007).

[163] K. Järvelin & J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems (TOIS), 20(4), 422- 446, (2002).

[164] P. He, H. Yuan, J. Chen & C. Zhao. An effective and scalable algorithm for hybrid recommendation based on Learning To Rank. Proceedings of the 1st International Congress on Signal and Information Processing, Networking and Computers (ICSINC 2015). 17-18 October. Beijing, China, (p. 59). CRC Press, (2016).

[165] S. T. Park & W. Chu. Pairwise preference regression for cold-start recommendation. Proceedings of the third ACM conference on Recommender systems. 23-25 October. New York, USA, (2009), (pp. 21-28).

[166] B. Shapira, L. Rokach & S. Freilikhman. Facebook single and cross domain data for recommendation systems. User Modeling and User-Adapted Interaction, 23(2-3), 211-247, (2013).

[167] A. Van den Oord, S. Dieleman & B. Schrauwen. Deep content-based music recommendation. Advances in Neural Information Processing Systems. 05-12 December. Lake Tahoe, NV, USA, (2013), (pp. 2643-2651).