SOCIAL NETWORK BASED RECOMMENDATION SYSTEM

A Paper
Submitted to the Graduate Faculty
Of the
North Dakota State University
Of Agriculture and Applied Science

By

Manish Singh

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

October 2013

Fargo, North Dakota

North Dakota State University

Graduate School

Title

Social Network based Recommendation System

By

Manish Singh

The Supervisory Committee certifies that this ***disquisition*** compiles with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Kendall Nygard
Advisor

Dr.Kenneth Magel

Dr.Limin Zhang

Approved by Department Chair:

11/13/2013
Date

Dr. Brian M. Slator
Signature

ABSTRACT

Every day we are overwhelmed with choices and options. Recommendation systems have gained popularity in providing suggestions. But every web application today has its own recommendation system. The recommendations provided by these systems are generic than user specific. Also the consumers of these systems are facing with the challenge of trusting these resources as they come from anonymous users.

In this paper we propose a social networking based collaborative filtering recommendation system for movies. The prediction rating for a movie is provided based on similarity between you and your friends in the recommendation system. We have used two approaches to derive the similarity function. In the first approach, similarity is achieved using cosine-vector similarity function. And in the second approach, we have used Pearson Correlation Coefficient. These similarity function results are then used to compute the final prediction rating for a user. Finally result from both the approaches is compared.

ACKNOWLEDGEMENTS

This research would not have been possible without the constant support, motivation and guidance of Dr. Kendall Nygard. He has been an extremely invaluable resource during my entire Masters Degree program. I would like to take this opportunity to express my earnest gratitude to Dr.Tariq King, for being one of my mentors who always guided and helped me in my courses. I would also like to thank my committee members Dr. Kenneth Magel and Dr.Limin Zhang for their time and support on my project. I am very thankful to the Computer Science Department, especially Carol Huber and Stephaine Sculthorp for their co-operation and help. Lastly, I would like to thank my friend Anubrata Dutta for studying with me, helping me and being a great support during my stay at NDSU.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

In our day to day life we come across choices and options. What to eat? Which movie to watch over the weekend? Which camera to buy? The decision making space is mushrooming as we progress. Without having prior knowledge of the domain space it is difficult to make a final decision. Hence people have relied on recommendations from their friends or the advice of experts.

Over the decade internet has become an important part of everyone's life. There is abundant amount of information available on the web. The challenge of information overload is faced by both retailers and consumers. The retailers have begun to find it practical to use algorithmic approaches to decide on content to show users. If a retailer displays more relevant content, then a consumer is more likely to show interest in purchasing. In order to find solution to this problem the area of recommendation system has emerged. Researchers have developed various algorithms and systems which are getting more commercialized by online vendors like Amazon.com, Ebay.com, and Netflix.com. For example, Netflix prize always comes to mind in this context. Hence recommendation systems have now become popular commercially and in research community

1.1. Research Motivation

Recommendation systems have changed the way people find products, information and even other people. They are collaborative, query-less engines which aims at providing recommendations to group of users about items that interest them [1]. They have various applications. For example:

i. Product Recommendations: The most commonly used recommendation system by online retailers. The product-product similarity is used to provide suggestions.

ii. Movie Recommendations: Based on your profile history, Netflix suggests its customer's recommendations of movies they might like.

There are different recommendation system engine out there in the market designed for specific domain. For example, YouTube recommends videos you might like; Netflix proposes movies you might want to watch. The recommendations are based on the top overall sellers on a site, prior purchase history of customer or demographics of customer. Hence analysis and deduction of recommendations depend upon the goal and domain in which it will be used.

Recommendation Systems produce a list of recommendations through one of the two approaches – content based filtering or collaborative based filtering. Content-based systems recommend based on items itself. They rely on rich content descriptions of the items that are being recommended. In this approach system selects items based on the correlation between the content of the items and the user's preferences. For instance, if a Netflix user has watched many comedy movies, then it recommends a movie from the database having the comedy genre [2]. Collaborative filtering systems suggest items to user based on the preference other users have expressed for those items. A system based on such algorithm collects user feedback in the form of ratings for items in a given domain and deducts similarities in rating amongst several users in determining the recommendations. Similar users are generally chosen because they share similar or highly correlated ratings history with our current user in context.

Even though collaborative filtering approach relies in user-user similarity, a new problem arises: the notion of trust. Users of recommendation systems face a big threat of being able to trust the suggestions provided by the system. This is primarily due to the fact that other users are not connected to them or share any kind of direct relationship with them. Today recommendation systems lack the ability to provide how much a current user considers every other user to be trustworthy and the ratings predicted to be valuable or relevant to him/her.

1.2. Research Objective

In this paper we propose a trust based recommendation system by bringing in the idea of social networking. Social influence plays an important role in human decision making process. The issue of trust in recommendation system is resolved as the other users in our current user's network are his/her friends. Also, the current user can state how much level of trust they consider on other users. This way the second issue of confidence level on friend's rating as per your preference is also resolved.

The recommendation system proposed in this research paper is Social Network based Recommendation System. It provides recommendation for movies. The user sets his/her preference level for various genres of movies. Additionally his/her profile also stores preference level of his/her friends which the active user can adjust to their confidence level. For example, a new movie of genre "Comedy" is released and current active user believes Friend A has better taste in Comedy than Friend B. Then active user can give Friend A, a higher confidence value than Friend B. The recommendation system is based on the collaborative filtering approach. It uses its technique of finding similarity between user and friend. The similarity function is derived from cosine similarity vector algorithm. The prediction rating is weighted average of

similarity values between user and friends and friend's ratings. We also developed a second similarity function using Pearson Correlation. And then the prediction rating is weighted average of Pearson Coefficient and movie ratings provided by user's friends. Finally we compare the results of the two approaches.

1.3. Outline of Paper

The rest of this paper is organized as follows: Chapter 2 provides the literature review information about recommendation systems, the two techniques of filtering and the evolution of social network in recommendation systems. Chapter 3 presents the research problem and its definition. Chapter 4 describes our approach, the algorithms used and its various concepts. Chapter 5 explains our experiment with a detailed use case using the two approaches and discusses results comparing the results from the two approaches used in our research. Finally, Chapter 6 concludes the paper.

CHAPTER 2. LITERATURE REVIEW

This chapter helps in understanding recommendation systems in general and the various approaches in building a recommendation system. It also highlights some of the related work done by researchers in the area of recommendation systems.

2.1. Background

2.1.1. Recommendation System

Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user [24, 25, 26].These suggestions might be items liked by users or of interest to users [18]. Recommender Systems were developed to help close the gap between information collection and analysis by filtering all of the available information to present what is most valuable to the user [18], [22]. Suggestions for electronic products on eBay, songs on Pandora, or movies on Netflix, are real world examples of the operation of industry-strength recommender systems. The recommendation system can be incorporated in various domains and hence their design depends on the domain and the particular characteristics of the data available [23]. They rely on user-profile or item-profile attributes such as age of a user or artist/genre for a song to provide suggestions. Hence they differ based on how they analyze the data and develop notion of affinity between user and items. Content based filtering and collaborative filtering are the two main techniques used to build recommendation systems. Researchers also combined these two techniques and noticed better prediction as a result of this hybrid approach.

2.1.2. Content-based Recommendation System

Content based recommendation systems provides recommendation based on items liked by user in the past [1].The approach to recommend in such systems has its roots in the information retrieval (IR) community, and employs many of the same techniques [39]. Systems implementing a content-based recommendation approach analyze a set of documents and/or descriptions of items previously rated by a user, and build a model or profile of user interests based on the features of the objects rated by that user [19]. The profile is then used in providing suggestions. In this approach, the attributes of user profile are matched with the attributes of a content object. The output of this approach tells us how relevant is the result against the user's level of interest in that object. If a profile accurately reflects user preferences, it is of tremendous advantage for the effectiveness of an information access process. For instance, it could be used to filter search results by deciding whether a user is interested in a specific Web page or not and, in the negative case, preventing it from being displayed.

2.1.3. Collaborative Filtering - based Recommendation System

Collaborative filtering based recommendation systems base its predictions and recommendations on the ratings or behavior of other users in the system. Unlike content based recommendation system, collaborative filtering based systems do not use features of items. Instead they use similarity of the user ratings for two items. In this approach the input is a matrix whose column represents item-profile vector and row represents user-profile of different users in the system. Then a similarity between users is determined using some distance measure such as Jaccard or cosine distance or Bayesian network [38]. Users in the neighborhood with highest similarity are taken into consideration for predicting an item to a user. [20]. Examples of systems

taking the collaborative filtering approach include GroupLens [40], the Bellcore video recommender [41], and Ringo [42].

Collaborative filtering can be implemented by various methods. These are primarily classified as neighborhood based and model-based approaches. In neighborhood-based approaches, subsets of users are chosen based on their similarity to the active user. Then the prediction rating is calculated for user using a weighted combination of their ratings. On the other hand, model-based approaches assume an underlying structure to users' rating behavior, and induce predictive models based on the past ratings of all users.

2.1.3. Defining Trust

Trust is a common phenomenon. Trust in computer science covers areas like security and access control in computer networks, reliability in distributed systems, game theory and agent systems, and policies for decision making under uncertainty. There are various definitions of trust. One of them is: "Trust is a psychological state comprising the intention to accept vulnerability based upon positive expectations of the intentions or behavior of another [38]." There are two forms of trust – experience-based and recommendation-based trust [55]. Experience-based trust stems from the former entity's knowledge, familiarity and outcome of previous communications with the latter entity. Recommendation-based trust on the other hand arises from shared information with other entities [56]

The two common ways of determining trust are through using policies or reputation [52]. Policies describe the conditions necessary to obtain trust, actions and outcomes if certain conditions are met. Reputation on the other hand is an assessment based on the history of

interactions with or observations of an entity. Later, researchers also started building trust models to determine trust in a system. [52]

As trust is a known concept used in a lot of different application domains, plethora of trust definitions exist. In our research paper, we are focusing on trust definitions in regards to recommendation systems. In general, three types of trust-based recommender systems have been studied [50, 51]. Trust is calculated from users' rating history, trust is inferred from direct users' experience, or trust is computed by a combination of direct users' experience and items' similarity. In Collaborative Filtering systems, trust between a pair of users u and v is measured in terms of a numerical scale, with the maximum value representing full trustworthiness and minimum representing a complete lack of trustworthiness.

2.2. Related Work

Netflix was one of the few companies that promoted research in the area of recommendation systems. From 2006 to 2009, Netflix sponsored a competition, offering a grand prize of $1,000,000 to the team that could take an offered dataset of over 100 million movie ratings and return recommendations that were 10% more accurate than those offered by the company's existing recommender system.

Collaborative filtering based algorithms that use user-item rating suffer from sparse and imbalance of rating data. Herlocker, et al. [5] present an excellent overview of the goals, datasets, and algorithms of collaborative filtering systems. This led researchers to explore other data sources that can be included in such algorithms. Using descriptive data about an item was first investigated by Balabanovic et al. [6]. Melville et al. [7] enhanced CF by using content of a movie, e.g., movie genre. Breese et al. [15] used a Bayesian clustering model to cluster users

based on their ratings. Ungar and Foster [16] also used a Bayesian approach to cluster users based on their preferences. Accuracy in predictions made recommendation system was enhanced by Pazzani [8] by using hybrid methods using both of user data (demographic information) and item data (content). Hybrid systems which combine content and collaboration have also been proposed in which various weights are set on the contribution of similarity [16]. Melville et al. [18] proposed a general framework for content-boosted Collaborative Filtering, where content-based predictions are applied to convert a sparse user ratings matrix into a full ratings matrix, and then a CF method is used to provide recommendations.

Trust has been widely studied by researchers in numerous disciplines [53] such as security and access control in computer networks, reliability in distributed systems, game theory and agent systems, and policies for decision making under uncertainty. Hexamor et al. [54] built a experience-based trust model for a diverse consumer community for Agent Systems. In this model, a group of customer agents with different risk attitudes are initialized with random trust values. The well-known Kerberos protocol [59] is used to exchange credentials in the field of security and access controls. Friend- of-a-Friend (FOAF) serves as a model for linking data over a distributed network. [57]

One of the major challenges faced by collaborative filtering based recommendation system is the problem of trust among users. Trust issues in recommendation systems have been addressed by several research studies. Massa and Avesani's study [9] showed that a user's trust network can solve the ad-hoc user problem, improve recommendation prediction and attenuate the computational complexity. Massa & Bhattacharjee, 2004, [13] suggested that incorporation of trust metric and similarity metric in recommendation system can increase the recommendation

accuracy. Collaboration groups using k-trusted neighbors than k-similar ones would result in better results were proposed by Lathia et.al.

The applicability of trust to recommender systems has been established in several research studies. Ziegler and Lausen [12, 45] showed that a correlation between trust and user similarity in an empirical study of a real online community. All Consuming 1 was an online community where users rate books. The authors showed that users were significantly more similar to their trusted peers than to the population as a whole. This work was extended in [22] which augmented the analysis of the All Consuming community and added an analysis. Within that community, results also showed a strong correlation between trust and similarity in movie ratings. The second result in [44] used the FilmTrust system [43] where users have stated how much they trust their friends in a social network and also rated movies. In this recommender system, trust is used in place of the Pearson correlation coefficient to generate predictive ratings. Results showed that when the users rating of a movie is different than the average rating, it is likely that the recommended rating will more closely reflect the user's tastes.

The awareness towards building social relationships with the intention of making up social communities all the way through the Internet has been drastically increased in recent years [57]. Social network based recommendation system are gaining popularity these days. The recommendations made by friends were showing better prediction results than the recommendations made by systems [10]. Several researchers have investigated exploring social networks and trust in particular. This is due to the fact that human decision making process is primarily affected by peers having similar taste preferences [11]. In an empirical study conducted by Ziegler and Lausen [12] showed a correlation between trust and user similarity. Abdul-

Rahman and Hailes [13] showed that in a predefined context, such as movies, users develop social connections with people who have similar preferences. Walter et al. [4] propose the use of social network information in recommendation systems and analyze the impact of trust dynamics on the performance of such a system.

CHAPTER 3. PROBLEM

Today any ecommerce website you go to for shopping recommends products for you to purchase. It also posts reviews of customers who have bought a product earlier. This results in too much information to mine. To overcome the challenge of information overload, recommendation systems have become popular in providing personalized suggestions to items of user's interest. But each website today has its own recommendation system. Hence the important question of trust arises.

Faced with overwhelming choices, people often take advice from their family and friends. Social influence plays an important role in product marketing. In computer science, social network is a graph that connects users. As use of social networking is increasing day by day, combining it with a recommendation system seems a plausible solution.

By the use of social network connections now the seeker can get more relevant and trustworthy result rather than generic anonymous recommendation. But there is another challenge here: the level of trustworthiness that can be associated with a friend. For example, when looking for movie recommendations we will often turn to our friends, on the basis that we have similar movie preferences overall. However, a particular friend may not be reliable when it comes to recommending a particular genre of movie. For example, Amanda wants to watch a new movie *"Hurt Locker"* that was released last weekend. Her friends Ann and David have already seen this movie. Ann tells her it is not a good movie but on the other hand David really likes the movie. But Ann likes romantic movies and *Hurt Locker* is an action movie. It is difficult for Amanda to completely rely on Ann's suggestion for the action movie. Hence a new issue of level of trust (reliability) that can be placed on a friend's suggestion arises.

Hence we have the following issues:

i. how much to trust the opinion giver

ii. compute similarity between users

iii. predict rating of a certain movie for a particular user based user-friend similarity and the rating for the movie by friends

3.1. Definition

Consider we have a list of i users $U = \{u_1,u_2,u_3,\ldots,u_i\}$ and a list of n movie genre preference $G = \{g_1,g_2,g_3,\ldots,g_n\}$. Then a particular user u_i has set a preference value for movie genres $P_{(gi,ui)} = \{p_{1(g1,ui)}, p_{1(g2,ui)}, p_{1(g3,ui)},\ldots, p_{1(gn,ui)}\}$. Similarly we will have such a list of preference value for each user belonging to U i.e. $Tp_{(gi,ui)} = \{P_{(gi,u1)}, P_{(gi,u2)}, P_{(gi,u3)},\ldots, P_{(gi,ui)}\}$. A preference value is rating between 1 and 5. If rating is not provided by user that implies user has shown "No Preference" for that genre. This in our case is also considered as a 0 rating. A user u_i might also have a list of friends $U_f = \{u_1, u_2, u_3, \ldots, u_f\}$ such that $U_f \subseteq U$.

Consider we have a list of movies $M = \{m_1, m_2, m_3,\ldots,m_p\}$in our database. These movies might be rated by one or more users in U. Let us consider an unrated movie m_a by active user u_i in recommendation system belonging to a genre g_i. Consider that each friend u_f of active user u_i has seen the movie and provided a rating $R_{(mi,ui)} = \{r_{1(mi,u1)}, r_{2(mi,u2)}, r_{3(mi,u3)},\ldots, r_{i(mi,ui)}\}$. Each user of U_f has set of preference value for each genre stored on current user u_i's profile. Now the active user u_i's profile has the following information:

i. active user u_i's preference value list

ii. group of user's friends U_f's preference value $T_{(gi,ui)}$

iii. group of user's friends U_f's movie rating $R_{(mi,ui)}$

The goal of the research project is to:

i. allow to set trust value on friends preferences

ii. compute similarity function between active user and each friend $s(u_i,u_f)$

iii. predict rating of the movie for active user u_i using similarity function $s(u_i,u_f)$ and movie rating by friend

CHAPTER 4. APPROACH

In reality we often look to our friends for recommendations. Collaborative filtering techniques help us achieve the solution we are looking for. The assumption made in this technique is other user preferences are taken into consideration to come up with a prediction for a desired user. The project uses this technique and builds a movie recommendation system whose users are friends or friend's friends of user. This ensures that recommendations are coming from trusted resources. The system stores various genres of movies as user preference. It provides suggestion for movies using these preferences. Also, the recommendation system allows a user to set a confidence level for his friend's preference. This confidence value is then used in providing the preference based final recommendation rating of new movie to the user.

In our research paper we have used two approaches to compute similarity function between users. This similarity function is then used to compute the final prediction ratings for the movie.

4.1. First Approach

In this approach we compute the similarity function using one of the information retrieval techniques Cosine Vector Similarity. The project is using the vector similarity and tf-idf weighting scheme to give the filtered ratings to the user about the product. That rating will come from the similarities between the user and his/her friends. The trust is implemented in the system by giving the ability to user to rate his/her friends for the given preferences. In this way a user can be sure about the final rating about the product and hence he/she will be able to buy a product efficiently and confidently. The steps involved in this approach are:

i. Convert the preference table of users into utility matrix by applying tf-idf weighting schema

ii. Use the tf-idf matrix and treating user preferences and friend preferences as vector computing the similarity between user and friend by applying Cosine Vector Similarity

iii. Computing the final prediction rating as weighted average of similarity value and movie ratings given by friends

4.1.1. TF – IDF Matrix

The tf–idf weight (term frequency–inverse document frequency) is one of the most commonly used weighting schemas in information retrieval and text mining. [13] This weight is a statistical measure used to evaluate how important a word is to a document in a collection. In information retrieval field, term weights are mainly used to represent the usefulness of terms in the retrieval process; for example, frequency [27], signal-to-noise ratio [28, 29], idf [30], relevance weighting methods [31] and tf–idf and its variations [32]. Other relevant fields include automatic term extraction in computational terminology, and also feature subset selection in machine learning [33].

Essentially, TF-IDF works by determining the relative frequency of words in a specific document compared to the inverse proportion of that word over the entire document corpus. Intuitively, this calculation determines how relevant a given word is in a particular document. Words that are common in a single or a small group of documents tend to have higher TFIDF numbers than common words such as articles and prepositions. Variations of the tf-idf weighting

scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. Basically there are two components of tf-idf

i. Term Frequency (tf)

ii. Inverse Document Frequency (idf)

In information retrieval, the term frequency $tf_{t,d}$ of term t in document d is defined as the number of times that t occurs in d. The inverse document frequency is popular heuristic measure. It is defined as the logarithm of ratio of number of documents in a collection to the number of documents containing the given word. The formal procedure for implementing TF-IDF will be slightly different based on its applicability to a problem, but the overall approach works as follows. Given a document collection D, a word w, and an individual document d ϵ D, we calculate

$$w_d = f_{w,d} * \log(|D|/f_{w,D})$$

where f_w, d equals the number of times w appears in d, |D| is the size of the corpus, and $f_{w,D}$ equals the number of documents in which w appears in D [34]. There are a few different situations that can occur here for each word, depending on the values of $f_{w,d}$, |D|, and $f_{w,D}$, the most prominent one.

In our recommendation system, we have two entities: user and movie genres. Users have preferences for genres which are represented by a numerical value between 0-5 where 0 implies No Preference i.e. the user has no preference for that genre. Tf-idf is a transformation we apply to preference ratings to get real-valued vectors. When measuring the similarity between users, items that have been rated by all are not as useful as less common items. The term frequency is

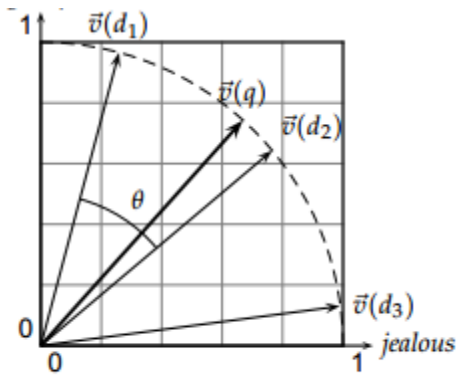
the actual preference rating. In terms of inverse document frequency we calculate as: $idf = log10 (N/df)$, Where, N is the total number of user in recommendation system, other than the active user and df is the total number of users that have a preference for a genre. Finally we multiply these term frequency and inverse document frequency. Hence result is the tf-idf matrix.

4.1.2. Cosine Vector Similarity

In information retrieval, the system ranks document based on a score. Several models have been proposed for this ranking and assigning a score process. The three most used models are the Vector space model, the Probabilistic models, and the Boolean model. The boolean model is based on what is called the "exact match" principle while the other two models are based on the concept of "best match." [26,37].The vector space model (VSM) is an IR model that represents the documents and queries as vectors in a multidimensional space, whose dimensions are the terms used to build an index to represent the documents. Document retrieval is based on the measurement of the similarity between the query and the documents. A similarity measures is a function which computes the degree of similarity between a pair of vectors .These measures rely heavily on terms occurring in both query and the document. If the query and document do not have any term is common then similarity score is very low. Different similarity measures have been suggested to match the query document. Some of popular measures are cosine, jaccard, dice etc.

In the field of information retrieval the similarity between two documents is measured with the help of cosine vector similarity algorithm. The angle between two vectors is used as a measure of divergence between the vectors, and cosine of the angle is used as the numeric similarity (since cosine has the nice property that it is 1.0 for identical vectors and 0.0 for

orthogonal vectors). In Measure similarity between two documents by treating each document as a vector of word frequencies and computing the cosine of the angle formed by the two frequency vectors.



Cosine similarity illustrated. $\text{sim}(d_1, d_2) = \cos\theta$.

Figure 1. Cosine Similarity

The similarity between two documents d1 and d2 can be computed using cosine similarity. We denote by V (d) the vector derived from document d, with one component in the vector for each dictionary term. The cosine similarity of the vector representations V (d1) and V (d2)

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \bullet \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|}$$

where the numerator represents the dot product (also known as the inner product) of the vectors V (d1) and V (d2), while the denominator is the product of Euclidean Length. The dot product x · y of two vectors is defined as:

$$\sum_{i=1}^{M} x_i y_i$$

Let V (d) denote the document vector for d, with M components $V_1(d)$. . . $V_M(d)$. The Euclidean length of d is defined to be

$$\sqrt{\sum_{i=1}^{M} \vec{V}_i^{\,2}(d)}$$

The next question to deal with is how we measure the similarity between user and his /her friend. The same logic of cosine vector similarity is applied to our recommendation system. We treat user and his/her friend as vectors of genre preference rate and compute cosine similarity between them. Given user u and his/her friend f, we seek to find the similarity between them. Using the tf-idf 2 dimensional matrix we will compute the cosine angle between u1 and u2 using the following formula.

$$\cos(\vec{u}, \vec{f}) = \frac{\vec{u} \bullet \vec{f}}{|\vec{u}\,\|\,\vec{f}\,|} = \frac{\sum_{i=1}^{|V|} u_i f_i}{\sqrt{\sum_{i=1}^{|V|} u_i^{\,2}}\sqrt{\sum_{i=1}^{|V|} f_i^{\,2}}}$$

This would result in value that ranges from 0 to 1 and reflect the degree to which u1 and u2 agree with each other.

4.1.3. Weighted Average Mean

The weighted arithmetic mean (WAM) is an averaging function that is commonly used in recommender systems to calculate a predicted rating based on a set of input ratings, where each input rating is assigned a weight representing its level of importance in calculating the overall rating. An input with a relatively large weight will cause the calculated rating to be more similar to itself than if it were assigned a smaller weight.

In a collaborative filtering recommender system, a common implementation of the prediction function, R(u, d), uses the similarity of user u and u's friends, uj, as weights in the weighted arithmetic mean:

$$R(u,d) = \sum_{j=1}^{k} sim(u,u_j)R(u_j,d))$$

where sim(u; uj) is a function that calculates the degree of similarity between u and one of its neighbors, uj. We will be using the same formula to compute the final prediction rating of the movie for our current active user.

4.1.4. Algorithm

Recommendation system uses collaborative filtering technique, cosine similarity, to compute a final rating of recommended movie. The cosine similarity helps us in determining how similar user and his /her friend are. The recommendation system also utilizes tf-idf weighting schema. The preferences table is transformed to utility matrix by applying tf-idf weighting schema. The result will generate tf-idf matrix of genre preference value. The tf-idf matrix will be then used to compute the cosine similarity between user and friends. To deduce the similarity, the summation of dot product between user and each product is calculated. Finally weighted average of cosine similarity of user and friend and their movie rating is calculated. This gives us the final recommended rating by our recommendation system for a movie not rated by active user. Below is the algorithm used in building the recommendation system.

i. for every movie $m_x \in M$ that user $u_i \in U$

ii. if m_x is not rated by user $u_i \in U$ i.e. $\neg\exists r_{1(mx,ui)} \in R_{(mi,ui)}$

iii. for every user $u_f \in U_f$, a friend of ui

iv. if m_x is rated by user $u_f \in U_f$ i.e. $\exists ! \, r_{2(mx,uf)} \in R_{(mi,ui)}$

v. for every $P_{(gi,uf)} \in T_{(gi,ui)}$ and $P_{(gi,ui)} \in T_{(gi,ui)}$

vi. apply tf-idf schema and generate tf-idf matrix $M_{(u,p)}$

vii. return tf-idf matrix $M_{(u,p)}$

viii. End for

ix. for $M_{(ui,pi)}$ and $M_{(Uf,Pf)} \in M_{(u,p)}$

x. Compute similarity $S_{(ui,Uf)}$ using $M_{(ui,pi)}$ and $M_{(Uf,Pf)}$

xi. return $S_{(ui,Uf)}$ for

xii. End for

xiii. compute weighted average W for $s(u_i, U_f)$ and $R_{(mi,ui)}$

xiv. return W (our final prediction rating $r_{(mx,ui)}$)

xv. End if

xvi. End for

xvii. End for

4.2. Second Approach

There have been several other similarity measures used in the literature like Spearman rank correlation, Kendall's τ correlation, mean squared differences, entropy, and Pearson's correlation [46]. In our second approach, we compute the similarity function using one of the statistical relationship techniques Pearson Correlation Coefficient. Similarity based on Pearson

correlation measures the extent to which there is a linear dependence between two variables. The user preferences and user's friends preference matrix is utilized to compute the Pearson Correlation Coefficient between user and friend. After this, the weighted average of the coefficient values and friend ratings is calculated. As this yields a value in the range from -1 to 1, the average for user preference ratings is added to compensate for rating scale variations. This is our final prediction rating for the movie for our current user. The steps involved in this approach are:

i. Calculate the "similarity" between user and friend by comparing how each user has rated the genres. Similarity between users is calculated using the Pearson Correlation Coefficient.

ii. Compute the weighted average of Pearson Correlation Coefficient value and friends movie rating table

iii. To compensate for rating scale, add average of current user preference value to the weighted average value computed in previous step

4.2.1. Pearson Correlation Coefficient

Pearson Correlation Coefficient (or Pearson product-moment correlation coefficient) is a measure of the strength of a linear association between two variables. It is widely used in statistics. It can take a range of values from -1 to +1. When it is near zero, there is no correlation, but as it approaches -1 or +1 there is a strong negative or positive relationship between the variables respectively.

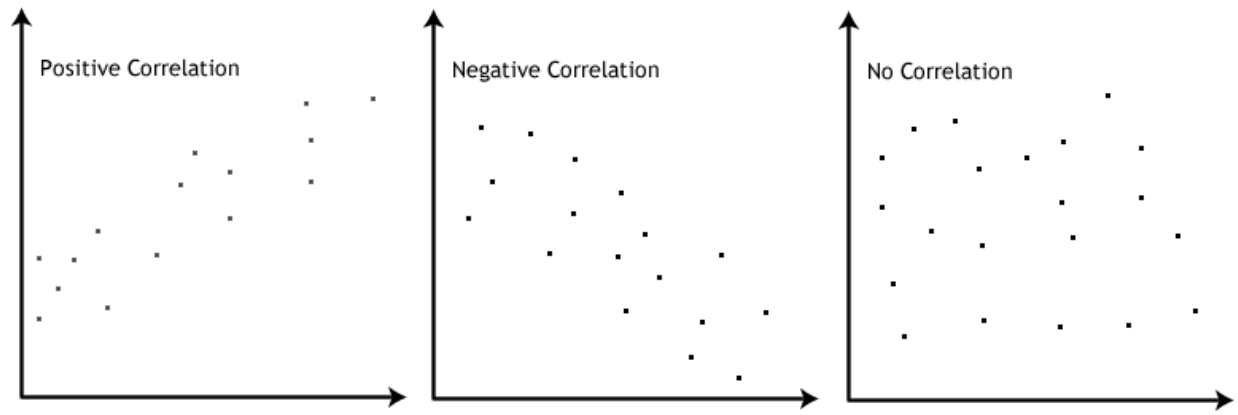


Figure 2. Pearson Correlation Coefficient

Recommendation Systems typically use Pearson Coefficient Correlation to find similarity between users in the database. Similarity between users is calculated using the Pearson Correlation Coefficient equation below. The formula is as follows:

$$P_{a,u} = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)^2 \times \sum_{i=1}^{m} (r_{u,i} - \bar{r}_u)^2}}$$

$P_{a,u}$ is the similarity between users a and u. m is the total number of items in common. $r_{a,i}$ is the rating user a gave an item. r_a with the bar over it is the average of all user a's ratings for all items. This calculation returns a value between -1 and 1. Negative correlations are generally believed to not be valuable in increasing prediction accuracy and one may choose to not use negative correlations. Strength of this formula is that it takes into account average ratings for each user. So if user a rates everything a 5 and user u rates everything a 1, then they still have a

similarity rating of 1.Now if users only have a couple of items in common, then their similarity should have less of an influence on the overall calculation.

4.2.2. Weighted Average Mean

As stated earlier in our first approach: the weighted arithmetic mean (WAM) is an averaging function that is commonly used in recommender systems to calculate a predicted rating based on a set of input ratings, where each input rating is assigned a weight representing its level of importance in calculating the overall rating. To account for users different ratings levels, base predictions on differences from a user's average rating. In our current approach we slightly modify the standard equation of weighted average mean as mentioned earlier. The formula is as follows:

$$p_{a,i} = r_a + \frac{\sum_{u=1}^{n} (r_{u,i} - r_u) \times P_{a,u}}{\sum_{u=1}^{n} P_{a,u}}$$

$p_{a,\,i}$ is the predicted rating user a would give item i. n is the number of similar users being compared to. $r_{u,\,i}$ is the rating user u gave the item. r_u with the bar over it is the average of all user u's ratings for all items.

4.2.3. Algorithm

In this second algorithm, we use Pearson Correlation Coefficient to compute the similarity function between user and his/her friends in the recommendation system. From the preferences table, user preferences value is considered as one variable and user's friend preferences as another variable. Then using the Pearson's Correlation equation the similarity is calculated. Next, the variation of weighted average mean is calculated. The base preference

ratings of each user are subtracted with average of that user's rating. Finally the average of current active users rating is added to this value. This results in final prediction rating for the movie. Below is the algorithm used in building the recommendation system.

i. for every movie $m_x \in M$ that user $u_i \in U$

ii. if m_x is not rated by user $u_i \in U$ i.e. $\neg\exists r_{1(mx,ui)} \in R_{(mi,ui)}$

iii. for every user $u_f \in U_f$, a friend of ui

iv. if m_x is rated by user $u_f \in U_f$ i.e. $\exists ! r_{2(mx,uf)} \in R_{(mi,ui)}$

v. for every $P_{(gi,uf)} \in T_{(gi,ui)}$ and $P_{(gi,ui)} \in T_{(gi,ui)}$

vi. Compute similarity $s_{(ui,uf)}$ using Pearson Correlation Coefficient

vii. return $s_{(ui,uf)}$ for

viii. End for

ix. compute weighted average W for $s(u_i, U_f)$ and $R_{(mi,ui)}$

x. return W (our final prediction rating $r_{(mx,ui)}$)

xi. End if

xii. sum weighted average W and average of u_i's preference rating $P_{(gi,ui)}$

xiii. End for

CHAPTER 5. EXPERIMENT

In order to demonstrate the prediction rating by our proposed recommendation system, we created some dataset in our database to be used in our experiment. Dataset includes users, preference ratings for movie genres by each user, friends list of a particular user, movies rated by user and movies rated by user's friends. Using the dataset as input we calculated the prediction rating for movies not rated by user by using both the approaches.

5.1. Sample Data

Let us apply our two approaches on a sample data and run the experiment to compute prediction rating for one movie. Assume we have seven users in our system and six movie genres. Let us consider that our active user u_i = Ron. The other users in the system except Sam are Ron's friends such that $U_f \subseteq U$. The preference value of movie genres for our current active user is $P_{(gi,ui)}$. U = {Ron, Ed, Vivian, Roseen, Gus, Rehan, Eric, Sam}

u_i = Ron

U_f = { Ed, Vivian, Roseen, Gus, Rehan, Eric }

G = {Drama, Suspense, Action, Comedy, Horror, Romance}

$P_{(gi,ui)}$ = {4,4,2,1,2,3}

Similarly we will have preference values of each user belonging to U i.e. $T_{(gi,ui)}$ = $\{P_{(gi,u1)}, P_{(gi,u2)}, P_{(gi,u3)}, \ldots, P_{(gi,ui)}\}$. The preference value for all users in our system is shown in Table 1. Now consider that the movie *Titanic* is not seen by Ron and wants his friend's

recommendation that have seen this movie already. The movie rating for *Titanic* by his friends is listed in Table 2.

Table 1. Preference Table

	Drama	Suspense	Action	Comedy	Horror	Romance
Ron(User)	4	4	2	1	2	3
Ed(Friend1)	3	2	1	No Pref	4	5
Vivian(Friend2)	1	3	No Pref	2	3	3
Roseen(Friend3)	2	4	3	2	4	4
Gus (Friend 4)	3	5	2	1	5	4
Rehan (Friend 5)	2	2	2	1	3	4
Eric (Friend 6)	1	1	2	4	4	5

Table 2. *Titanic* Movie Rating

Friends	Rating
Ed	3
Vivian	1
Roseen	5
Gus	1
Rehan	2
Eric	2

5.2. First Approach

We apply our first approach algorithm to the dataset to compute the final prediction rating of movie *Titanic* for user Ron. The step by step process is shown below.

Step 1: First we apply tf-idf weighting schema to the preferences table to transform it to TF-IDF Matrix (Utility Matrix). The tf-idf formula is

$$wt = tf * idf$$

Where, tf = term frequency which is equivalent to preference value of genre; idf = $\log_{10}$ (N/df). Where, N (=7) is the total number of users present in recommendation system, other than

the active user and df (=6) is the number of friends who have rated the movie. The following table i.e. Table 3 represents the tf-idf matrix which will be the input for next step.

Table 3. tf-idf Matrix

	Drama	Suspense	Action	Comedy	Horror	Romance
Ron	0.2678	0.2678	0.1339	0.0669	0.1339	0.2009
Ed	0.0669	0.1339	0.0669	No Pref	0.2678	0.3347
Vivian	0.0669	0.2008	No Pref	0.1339	0.2009	0.2009
Roseen	0.1339	0.2678	0.2008	0.1339	0.2678	0.2679
Gus	0.2008	0.3348	0.1339	0.0669	0.3348	0.2678
Rehan	0.1339	0.1339	0.1339	0.0669	0.2009	0.2678
Eric	0.0669	0.0669	0.1339	0.2678	0.2678	0.3347

Step 2: Next we compute the cosine similarity between Ron and each of his friends using the tf-idf matrix. The formula used for cosine similarity is:

$$\cos(\vec{u}, \vec{f}) = \frac{\vec{u} \bullet \vec{f}}{|\vec{u}||\vec{f}|} = \frac{\sum_{i=1}^{|V|} u_i f_i}{\sqrt{\sum_{i=1}^{|V|} u_i^2}\sqrt{\sum_{i=1}^{|V|} f_i^2}}$$

Here u = Ron's tf-idf row from tf-idf matrix and f = Friend's tf-idf row. Table 4 shows the similarity between Ron and his friends.

Table 4. Cosine Similarity

Cosine Similarity	Value
cos (Ron,Ed)	0.7633
cos(Ron, Vivian)	0.8250
cos(Ron,Roseen)	0.9121
cos(Ron,Gus)	0.9329
cos(Ron,Rehan)	0.8947
cos(Ron,Eric)	0.6949

Table 4 shows that Ron shares the highest similarity with Gus, Roseen and Rehan in the decreasing order. And he shares least similarity with Eric, Ed and Vivian in the decreasing order. These similarity measures help in assigning the weights to the final prediction ratings. Hence higher the similarity higher will be its weight in the prediction rating.

Step 3: Finally we compute the prediction rating using the Weighted Average Mean. The formula is:

$$R(u,d) = \sum_{j=1}^{k} sim(u,u_j)R(u_j,d))$$

where $w_{i,j}$ is the cosine similarity value between user u_i and friend u_j . And $r_{a,j}$ is the movie rating provided by friend u_j to movie *Titanic* (Table 2).

$$= \frac{(0.7633*3 + 0.8250*1 + 0.9121*5 + 0.9329*1 + 0.8947*2 + 0.6949*2)}{(0.7633 + 0.8250 + 0.9121 + 0.9329 + 0.8947 + 0.6949)}$$

= 2.3468

Hence the final predictions rating of *Titanic* for Ron is 2 using first approach in our recommendation system.

5.3. Second Approach

Now we apply our second approach algorithm to the dataset to compute the final prediction rating of movie *Titanic* for user Ron. The step by step process is shown below.

Step 1: In the first step, we take the preferences table (Table 1) and compute the similarity between Ron and his friends using Pearson Correlation Coefficient. The similarity equation of Pearson Correlation Coefficient is:

$$P_{a,u} = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a) \times (r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)^2 \times \sum_{i=1}^{m} (r_{u,i} - \bar{r}_u)^2}}$$

Here $r_{u,i}$ is our active user Ron's preference value for a movie genre; r_u is the average of preferences value of all genres; $r_{a,i}$ are Ron's friends, r_a is the friend's preferences value average. Table 5 shows the average values of preferences and Table 6 shows the Pearson Correlation Coefficient.

Table 5. Preference Table with Averages

	Drama	Suspense	Action	Comedy	Horror	Romance	Average
Ron(User)	4	4	2	1	2	3	2.6667
Ed	3	2	1	No Pref	4	5	2.6000
Vivian	1	3	No Pref	2	3	3	2.4000
Roseen	2	4	3	2	4	4	3.1667
Gus	3	5	2	1	5	4	3.3333
Rehan	2	2	2	1	3	4	2.3333
Eric	1	1	2	4	4	5	2.8333

Table 6. Pearson Corelation Coefficient

Similarity Function	Pearson Correlation Coefficient
s (Ron,Ed)	-0.2033
s(Ron, Vivian)	-0.1239
s(Ron,Roseen)	0.2240
s(Ron,Gus)	0.5731
s(Ron,Rehan)	0.2665
s(Ron,Eric)	-0.6072

According to Pearson Correlation, positive values of coefficient indicate similar interests and negative values imply dissimilar interests. The results of Pearson Correlation Coefficient show that that Ron shares similar taste or positive tastes with Gus, Rehan and Roseen. This implies that the calculation of final prediction rating will take higher weight-age of these friends rating than others. Earlier we have concluded the same from our Cosine Similarity values.

Pearson Correlation Coefficient helps in determining the strength of relationship between two variables. The coefficient takes a value between [-1, 1]. A positive correlation coefficient indicates that an increase in the first variable would correspond to an increase in the second variable, thus implying a direct relationship between the variables. A negative correlation indicates an inverse relationship whereas one variable increases the second variable decreases. In

our recommendation system, we are trying to finding similarity between our active user and his friends. Pearson Correlation perfectly matches our requirement and helps us in finding similarity between users. A positive value of coefficient indicates users have similar taste and negative value implies user dissimilar tastes. Also, results of Table 6 suggest that the concept of similarity matches with Cosine Similarity.

Step 2: Finally we compute the prediction rating. To account for users different ratings levels, base predictions on differences from a user's average rating. The formula used is:

$$p_{a,i} = r_a + \frac{\sum_{u=1}^{n} (r_{u,i} - r_u) \times P_{a,u}}{\sum_{u=1}^{n} | P_{a,u} |}$$

Where $p_{a,\ i}$ is the predicted rating user a would give item i. n is the number of similar users being compared to. $r_{u,\ i}$ is the rating of friend user given to the movie. r_u with the bar over it is the average of all preferences value of u and $P_{a,u}$ is the Pearson Correlation Coefficient.

$p_{a,i}$= 1.8591

The recommendation system prediction rating is **1.8591** using our second approach in the recommendation system**.**

5.4. Results

In our experiment we have initially selected few movies and retrieved their prediction rating using both the approaches. The experiment is now conducted on fifteen movies. The following table shows prediction ratings for 15 movies.

Table 7. Comparison between Cosine & Pearson for 15 movies

Movie	Cosine Vector Similarity	Pearson Correlation
Movie 1	5.0000	4.5219
Movie 2	4.0000	3.8319
Movie 3	1.8279	1.4048
Movie 4	3.3505	3.6659
Movie 5	1.0000	0.8318
Movie 6	2.5161	2.5654
Movie 7	2.5032	1.8625
Movie 8	2.0000	1.8938
Movie 9	2.7643	2.7725
Movie 10	1.4883	1.4846
Movie 11	3.5022	3.4320
Movie 12	2.6495	1.8118
Movie 13	2.9749	3.0414
Movie 14	4.1302	4.2263
Movie 15	3.0000	2.7699

Table 7 shows that the prediction rating from both the approaches is close to each other. We now plot the data result on a graph. Pearson Correlation Coefficient is the same as the cosine similarity distance between centered ratings vectors. In statistics centered ratings means, subtracting the mean of the vector from the vector. This would result in the numerator similar to

that Pearson Correlation Coefficient. Hence similar rating results are expected and also seen from Table 7. The following graph i.e. Figure 3 shows the ratings from the above table.

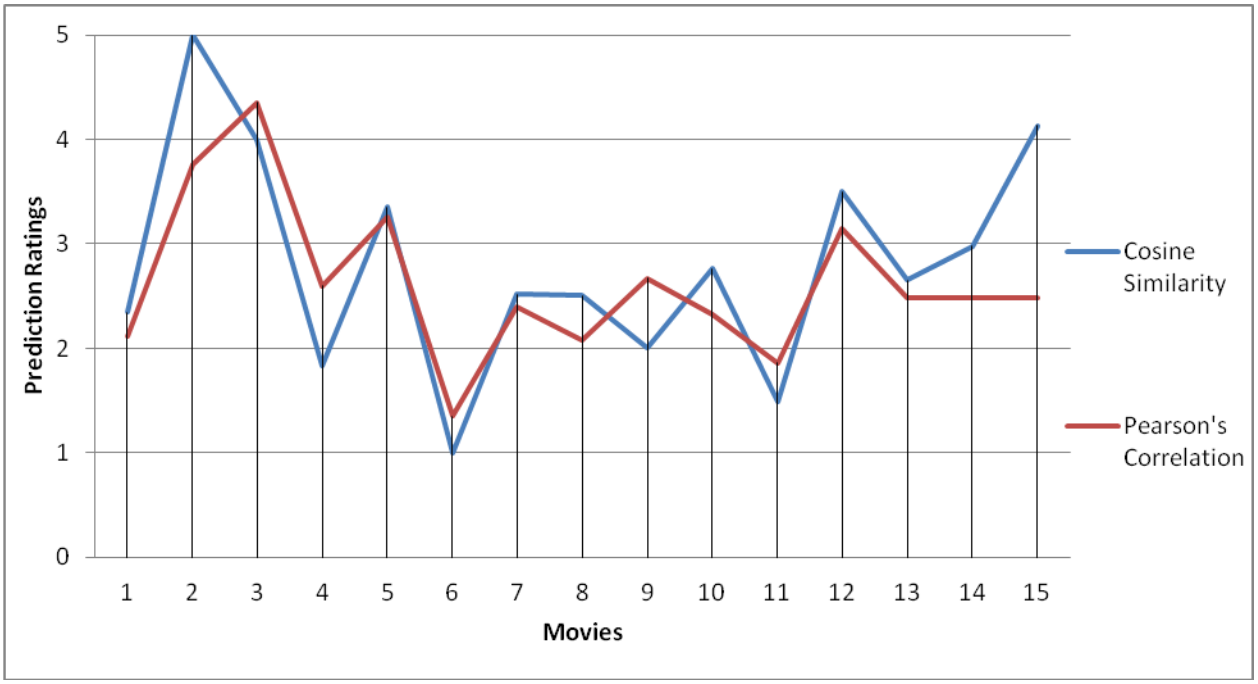


Figure 3. Prediction Ratings for 15 movies

Now the experiment is run on a large data set. The prediction rating for 100 movies is calculated. The following graph shows the ratings for 100 movies. It clearly depicts that Cosine Similarity and Pearson Correlation suggest similar ratings more or less with small differences.

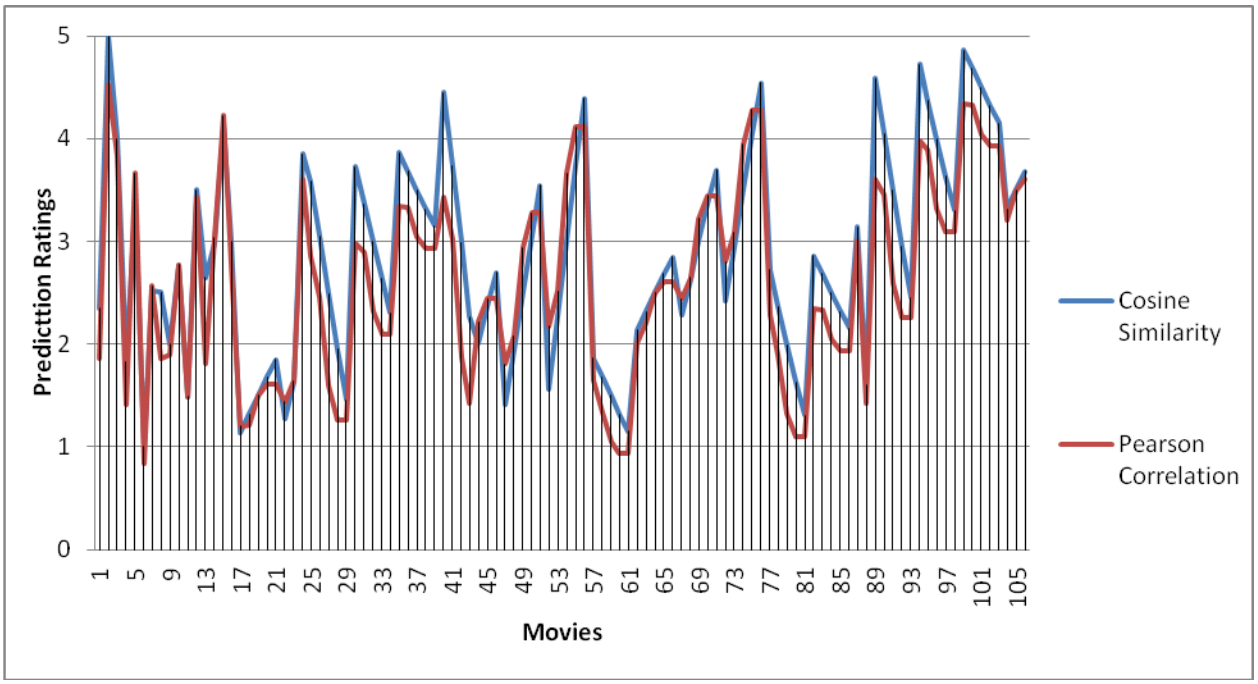


Figure 4. Prediction Ratings for 100 movies

5.5. Analysis

The difference, mean and variance of prediction ratings from Cosine Similarity and Pearson Correlation Coefficient for Table 7 are as follows. In probability theory and statistics, variance measures how far a set of numbers is spread out. A small variance indicates that the data points tend to be very close to the mean and hence to each other, while a high variance indicates that the data points are very spread out from the mean and from each other.

From the table we see that there is a small variance indicating that prediction rating from both the approaches are very close to the Mean and hence very close to each other. In some cases, the variance is 0 indicating the ratings are similar from both the approaches. These results are expected as Pearson Correlation Coefficient is like adjusted Cosine Similarity.

Table 8. Difference, Mean & Variance of Prediction Ratings

Movie	Difference	Mean	Variance
Movie 1	0.4781	4.761	0.1143
Movie 2	0.16811	3.916	0.0141
Movie 3	0.4231	1.6164	0.0895
Movie 4	0.3154	3.5082	0.0497
Movie 5	0.1682	0.9159	0.0141
Movie 6	0.0493	2.5408	0.0012
Movie 7	0.6407	2.1829	0.2052
Movie 8	0.1062	1.9469	0.0056
Movie 9	0.0082	2.7684	0.000
Movie 10	0.0037	1.4865	0.000
Movie 11	0.0702	3.4671	0.0025
Movie 12	0.8377	2.2307	0.3509
Movie 13	0.0665	3.0082	0.0022
Movie 14	0.0961	4.1783	0.0046
Movie 15	0.2301	2.8850	0.0265

The following graphs i.e. Figure 5 and Figure 6 show the prediction rating from both the approaches along with mean, difference and variance of the ratings.

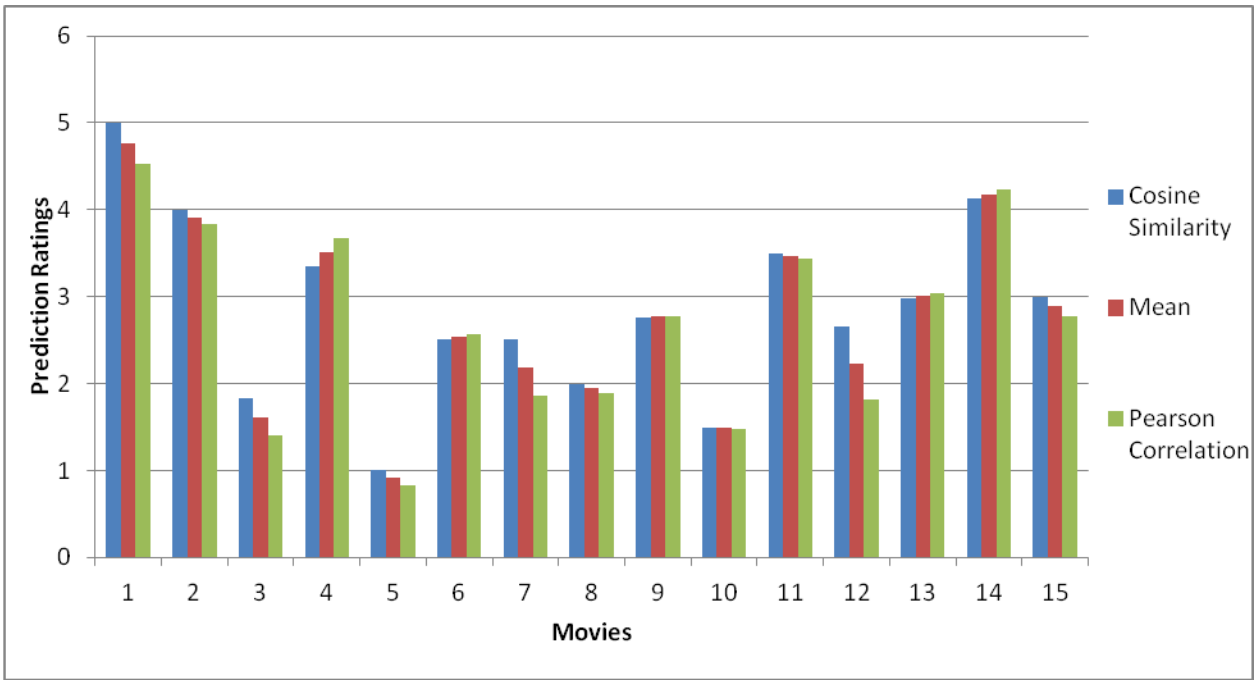


Figure 5. Mean Ratings for 15 movies

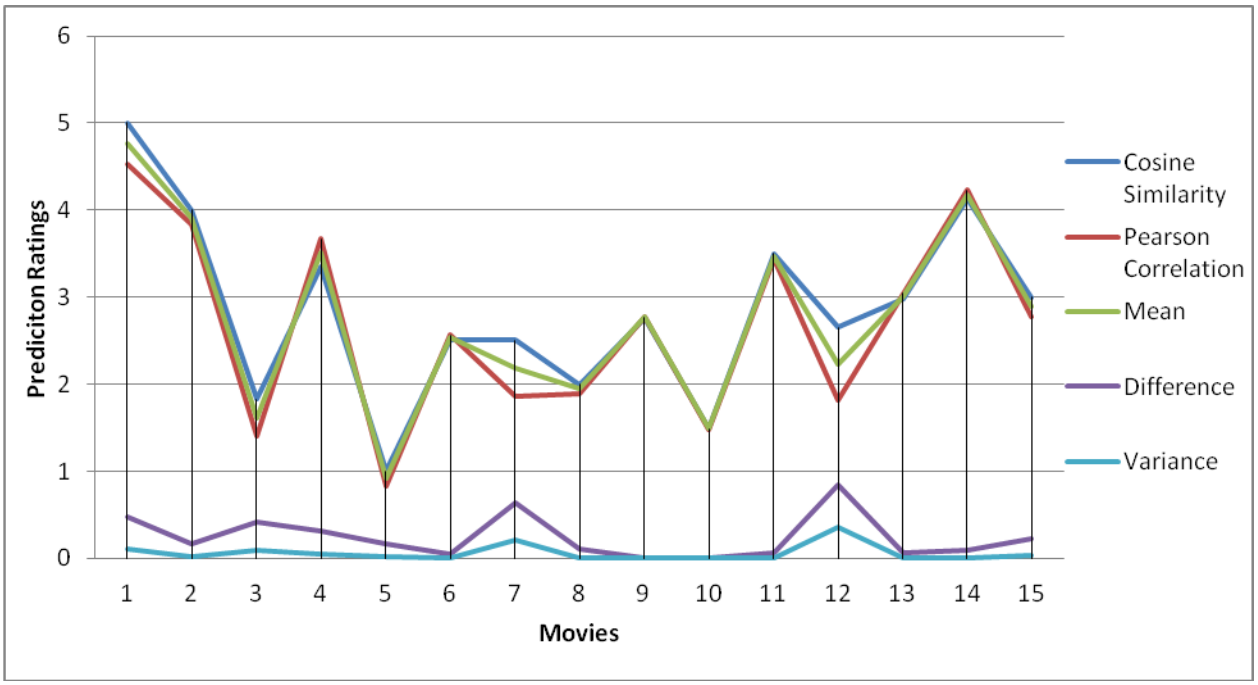


Figure 6. Mean, Difference and Variance of Prediction Ratings for 15 movies

Figure 7 displays the mean, difference and variance for larger data set. The graph indicates that both the approaches yield similar ratings with very less difference between them.

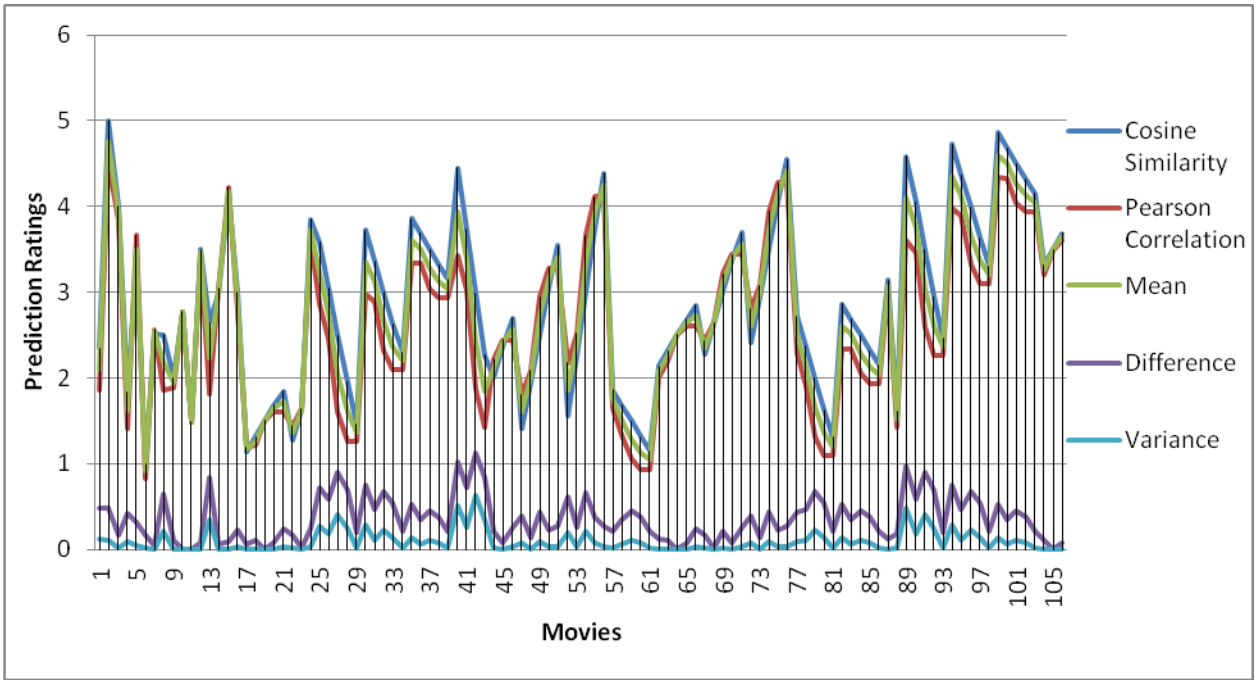


Figure 7. Difference, Variance of Prediction Ratings for 100 movies

CHAPTER 6. CONCLUSION

The paper proposed two methods for building recommendation system. There were two major issues identified in the research paper (1) trust in recommendation system (2) level of trust confidence on opinion giver (3) predict ratings for movie. The trust issue (1) is resolved by building a recommendation system where who will add other users as friends on whom you can rely on for suggestions. The issue (2) on how much trust to infuse on a friend was achieved by allowing the user to change the preference rating of his/her friends. This way they can set which friend's preference should be given higher priority than others for specific genre.

The issue (3) was achieved using two approaches. The first approach uses IR techniques like tf-idf weighting schema and cosine vector similarity algorithm to calculate the similarity between user and his friends. Then using the similarity value the final rating is calculated. In the second approach the similarity is calculated using Pearson Correlation Coefficient. To account for users different ratings levels, base predictions on differences from a user's average rating. Comparing the two approaches prediction ratings suggests that the predictions are very similar. Person Correlation Coefficient is the same as vector similarity over centered ratings vectors.

By our approach, we realized that content-based filtering techniques, like tf-idf weighting schema and cosine vector algorithm can be applied to collaborative filtering based recommendation systems. Cosine Similarity Vector Algorithm helps us in determining similarity between users. The similarity values indicated that similar results were seen from Pearson Correlation Coefficient. Finally, the prediction ratings generated by our approach is very close to the ratings generated using the standard and most frequently used technique of recommendation system: Pearson Correlation Coefficient.

REFERENCES

[1] Prem Melville and Vikas Sindhwani, Recommendation Systems, In Encyclopedia of Machine Learning, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010 Chapter No: 00338, Pg 829-838

[2] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, Collaborative Filtering Recommender Systems, Foundations and Trends in Human–Computer Interaction, 2010, Vol. 4, No. 2 Pg 81–173

[3] Jianming He and Wesley W. Chu, A Social Network-Based Recommender System (SNRS) , 7th International Conference, HAIS (Hybrid Artificial Intelligent Systems) 2012 Volume 7208, 2012, Pg 91-102

[4] Philip Bonhard, Improving Recommender Systems with Social Networking, In conjunction with the 2005 International Conference on Intelligent User Interfaces, 2005, Pg 89-100

[5] Jennifer Golbeck, A.V.Williams Building, Generating Predictive Movie Recommendations from Trust in Social Networks. Proceedings of Fourth International Conference on Trust Management, Pisa, Italy, May 2006. Pg 57-74

[6] M. Balabanovi´c and Y. Shoham, Fab: content-based, collaborative recommendation. Communications of the ACM, 1997, Vol. 40(3), Pg 66–72

[7] P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations, In Eighteenth national conference on Artificial intelligence, 2002, Pg 187–192

[8] M. J. Pazzani, A framework for collaborative, content-based and demographic filtering. Artificial Intelligence. Rev., 1999, Vol.13(5-6): Pg 393–408

[9] Massa, P. & Avesani, P. Trust-aware Collaborative Filtering for Recommender Systems, In: Proceedings of Federated International Conference On The Move to Meaningful Internet, Agia Napa, Cyprus, (2004) ,Pg. 492 - 508

[10] Sinha, R. & Swearingen, K., Comparing Recommendations Made by Online Systems and Friends, In: Proceedings of DELOS Workshop on Personalisation and Recommender Systems, 2001,

[11] A. Abdul-Rahman and S. Hailes, Supporting trust in virtual communities. In Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000. Volume 6 Page 6007

[12] Cai-Nicolas Ziegler and Georg Lausen, Analyzing correlation between trust and user similarity in online communities. In Proceedings of the Second International Conference on Trust Management, 2004, Pg: 251-265

[13] Massa, P. & Bhattacharjee, B. (2004). Using Trust in Recommender Systems: An Experimental Analysis, Trust Management: Second International Conference, 2004, Oxford, UK, March 29-April 1, 2004, Volume 2995, 2004, pp 221-235

[14] Abdul-Rahman, A. and Hailes, S., Using Recommendations for Managing Trust In Distributed Systems. In Proceedings of IEEE Malaysia International Conference on Communication, 1997

[15] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the 14 th Conference on Uncertainty in Artificial Intelligence, 1998, Pg 43–52.

[16] Melville P., Mooney R.L. , Nagarajan R., Content-Boosted Collaborative Filtering for Improved Recommendations, In proceedings of Eighteenth national conf. of Artificial Intelligence, 2002, Pg.187-192

[17] Prem Melville, Raymond J. Mooney, and Ramadass Nagarajan. Contentboosted collaborative filtering for improved recommendations. In Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI-02), 2002 , Pages 187–192

[18] P. Resnick and H.R. Varian, Recommender systems, In proceedings of Communications of the ACM, 1997, 40(3): Pg 56-58

[19] Mladenic, D.: Text-learning and Related Intelligent Agents: A Survey, IEEE Intelligent Systems and their applications, 1999, Vol. 14(4), Pg.44–54

[20] Jeffrey D.Ullman, Jure Leskovec, Anand Rajaraman, Mining of Massive Datasets, Cambridge University Press, Chapter 9, Pg: 305-338

[21] Byeong Man Kim & Qing Li & Chang Seok Park & Si Gwan Kim & Ju Yeon Kim, A new approach for combining content-based and collaborative filters, J Intell Inf Syst 2006, Vol 27: Pg. 79–91

[22] Nitin Agarwal, Ehtesham Haque, Huan Liu, and Lance Parsons, Research Paper Recommender Systems: A Subspace Clustering Approach, In International Conference On Web-Age Information Management, Pg: 475--491

[23] C. Basu, H. Hirsh, and W. Cohen. Recommendation as Classification: Using social and content-based information in recommendation. In Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98), pages 714–720, July 1998.

[24] Mahmood, T., Ricci, F.: Improving recommender systems with adaptive conversational strategies. In: C. Cattuto, G. Ruffo, F. Menczer (eds.) Hypertext, pp. 73–82. ACM (2009)

[25] Burke, R.: Hybrid web recommender systems. In the Adaptive Web, 2007 Vol 4321, Pg. 377–408.

[26] Zhou, Daniel and Resnick, Paul, Assessment of conversation co-mentions as a resource for software module recommendation, In Proceedings of RecSys '09. ACM, New York, NY, Pg. 133-140

[27] Luhn, H. P, A statistical approach to mechanized encoding and searching of literary information, In IBM Journal of Research and Development, 1957, Vol 1(4), Pg: 309–317.

[28] Dennis, S. F, In M. E. Stevens, V. E. Giuliano, & L. B. Heilprin (Eds.), Statistical association methods for mechanized documentation, Symposium Proceedings (Miscellaneous publication 269). Washington, DC: National Bureau of Standards.

[29] Salton, G., & Buckley, C., Weighting approaches in automatic text retrieval, In Information Processing and Management, 1988, Vol. 24(5), Pg. 513–523.

[30] Sparck-Jones, K., A statistical interpretation of term specificity and its application in retrieval, In Journal of Documentation, 1972, Vol. 28(1), Pg. 11–21.

[31] Robertson, S. E., & Sparck-Jones, K, Relevance weighting of search terms, In Journal of the American Society of Information Science, 1976 Vol. 27, Pg. 129–146

[32] Slonim, N., & Tishby, N, Document clustering using word clusters via the information bottleneck method, In Proceedings of the 23rd international conference on research and development in information retrieval,2000, Pg. 208–215

[33] Akiko Aizawa, An information-theoretic perspective of tf–idf measures, In Information Processing and Management 2003, Vol. 39 Pg. 45–65,

[34] Berger, A et al, Bridging the Lexical Chasm: Statistical Approaches to Answer Finding, In Proceedings of Int. Conf. Research and Development in Information Retrieval, 2000 Pg. 192-199.

[35] Nicholas J. Belkin and W. Bruce Croft, Information filtering and information retrieval: Two sides of the same coin, In Communications of the ACM, Dec 1992 v35 n12 p29(10)

[36] Belkin, N.J. and Croft, W.B, Retrieval techniques, In Annual Review of Information Science and Technology, 1987, Chapter. 4, Pg. 109-145

[37] Salton, G. and McGill, M.J., Introduction to Modern Information Retrieval, McGraw-Hill, 1983, Pg. 100-134

[38] Abhay S. Harpale and Yiming Yang, Personalized active learning for collaborative filtering, In SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, 2008, pages 91–98

[39] Marko Balabanovic and Yoav Shoham, Content-based and collaborative recommendation, Communications of the Association for Computing Machinery, Volume 40 Issue 3, March 1997 Pages 66-72

[40] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J, GroupLens: An open architecture for collaborative filtering of netnews, In Proceedings of the ACM Conference on Computer-Supported Cooperative Work (Chapel Hill, NC) 1994, Pg:175-186

[41] Hill, W., Stead, L., Rosenstein, M., and Furnas, G, Recommending and evaluating choices in a virtual community of use, In Conference on Human Factors in Computing Systems—CHI '95, 1995, Pg: 194--201

[42] Shardanand, U., and Maes, P, Social information filtering: Algorithms for automating "word of mouth.", In Conference on Human Factors in Computing Systems—CHI '95., 1995, Pg. 210-217

[43] Jennifer Golbeck, Generating predictive movie recommendations from trust in social networks, In Proceedings of the Fourth International Conference on Trust Management, 2006, Volume 3986, Pg. 93-104

[44] Cai-Nicolas Ziegler and Jennifer Golbeck, Investigating Correlations of Trust and Interest Similarity, Decision Support Services, In Proceedings of the Second International Conference on Trust Management, 2006, Pg.460-475

[45] Cai-Nicolas Ziegler and Georg Lausen, Making Product Recommendations More Diverse, IEEE Data Eng. Bull 2009, Vol.32 (4) Pg.23-32

[46] Herlocker, J.L., Konstan, J.A., Borchers, A., Riedl, J., An Algorithmic Framework For Performing Collaborative Filtering, In Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR '99), 1999, Pg. 230-237

[47] Gediminas Adomavicius and Alexander Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. on Knowl. and Data Eng., 2005, Vol. 17(6): Pg. 734–749

[48] Marko Balabanovic and Yoav Shoham. Fab: Content-based, collaborative recommendation. Communications of the Association for Computing Machinery, 40(3):66–72, 1997.

[49] Zunping Cheng, Neil Hurley, Analysis of robustness in trust-based recommender systems, RIAO '10, Adaptivity, Personalization and Fusion of Heterogeneous Information, April 2010, Pages 114-121

[50] M. Jamali and M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, Pg. 397-406.

[51] J. O'Donovan and B. Smyth, Trust in recommender systems, In Proceedings of the 2005 International Conference on Intelligent User Interfaces, 2005, Pg. 167-174

[52] Bonatti, P., Duma, C., Olmedilla, D., and Shahmehri, N., An integration of reputation-based and policy-based trust management, In Proceedings of the 2005 Semantic Web Policy Workshop, 2005, Pg 164–173

[53] Lewicki, R. J. and Bunker, B. B. Trust in relationships: a model of development and decline. In: Bunker, B. B. and Rubin, J. Z. (eds.) Conflict, Cooperation, and Justice: Essays Inspired by the work of Morton Deutsch. San Francisco, Jossey-Bass, 1995, 133-173

[54] Hexmoor, H., A Methodology for Comparing Service Policies Using a Trust Model. In International Journal of Computer Science & Applications, 2009, Vol. 6, No. 4, Pg.1 – 15

[55] Griffiths, N, "Task Delegation using Experience-Based Multi-Dimensional Trust.", In Proceedings of the Fourth International Conference on Autonomous Agents and Multiagent Systems, 2005, Pg. 489-496

[56] Sabater, J., Sierra, C, "REGRET: A Reputation Model for Gregarious Societies.", In Proceedings of the First International Joint Conference on Autonomous Agents in Multi Agent Systems (AAMAS-02), 2002, Pages 475-482

[57] Hexmoor, H, Trust-based protocols for regulating on-line, friend-of-a-friend communities, In Journal of Experimental and Theoretical Artificial Intelligence, Taylor and Francis, 2009, Pg. 1-21

[58] Rousseau, D. M.; Sitkin, S. B.; Burt, R. and Camerer, C, Not so different after all: A cross-discipline view of trust, In Academy of Management Review 2009, Vol. 23(3), Pg. 393-404

[59] Kohl, J. and Neuman, B. C., The kerberos network authentication service, In IETF RFC 1510, 1993, Vol. 5. Pg. 1-135