Developing an Ontological approach to Content-based Recommendation System

A
Project Report
Submitted for the Partial Fulfilment
of the Degree of

Bachelors of Technology
in
Computer Science & Engineering

By
Ananda Kumar Behera
Roll No. 110CS0116

Under the Guidance of
Prof. Korra Sathya Babu
Department of Computer Science & Engineering

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

NATIONAL INSTITUTE OF TECHNOLOGY

ROURKELA

CERTIFICATE

This is to certify that this project report entitled

'***Developing an Ontological approach to Content-based Recommendation System***'

submitted by **Ananda Kumar Behera** bearing the Roll No. **110CS0116** for the partial fulfilment of the requirements for the award of the degree of Bachelors of Technology in Computer Science & Engineering at National Institute of Technology Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date

Dr. Korra Sathya Babu
Department of Computer Science & Engineering
National Institute of Technology, Rourkela

ACKNOWLEDGEMENT

I feel myself fortunate for getting an opportunity to work under Dr. K. Sathya Babu, and for involving me in this project with a lot of scope. I express my sincere gratitude and indebtedness for his invaluable guidance, encouragement, thoughtful advices and above all for his cooperative attitude. I really learned more about different recommendation system and their evaluation method with a greater detail.

Ananda Kumar Behera

DECLARATION

I Ananda Kumar Behera of Computer Science & Engineering with Roll no 110CS0116 hereby declare that the project submitted by me is solely of my work and is not copied from any other source where ever may available and It has not been previously submitted for any academic degree. I had verified my thesis report through Turnitin software for plagiarism. All sources of quoted information have been acknowledged by means of appropriate references.

If in future my work was found to be plagiarized from any other person's work, then in that situation I alone will be responsible for it.

Date: 11th May 2014

Ananda Kumar Behera

NIT, Rourkela

ABSTRACT

During past decade the no of web user increases so rapidly leading to rapid increase in web services which leads to increase the usage data at higher rate. The usage data of these users now amount to be in order of Peta Byte (10^{15} bytes). In such cases the search space for user's queries increases and a user's search query may leads to retrieval of irrelevant information. Sometime the search algorithm may become exhaustive. This project is aimed to use User's context information to model a framework which can filter the search space and choose some preference based on user's context. The project follows a set of processes for profile construction of Users and Items, and determining their similarity and scoring their preference.

The projects also compares the effectiveness in predicting User's preferences and accuracy in it with various other Collaborative approach such as User-Based model and Item based model to check its performance level and quality of predictions.

Contents

Table of Figures:

1 Introduction

1.1 Objective of the Section:

This section provides a complete elaboration of Recommendation System (RS), history over the development of recommendation, their needs in the days of rapidly growing web users, their applications in daily life and currently existing Recommendation Systems. The classes of Recommendation techniques are described over the following few pages.

Content:

1. Introduction to Recommendation System
2. Characteristics of a good Recommendation System
3. Classification of Recommendation System
4. Applications Of Recommendation System
5. Challenges faced by a Recommendation System

1.2 Introduction:

In a world of largely growing largely growing web user, the no of websites or links, the web contents are doubled in each 18 months (by Moore's Law). Alone Facebook and Twitter contain a certain Petabytes of data and the database sizes increases rapidly as the current active Users touches 2 billion marks. Around 2.5 billion contents are processed in a single day for Facebook. The technical revolution and broader network access leads to various web services ranging from education to e-shopping, web content search, social networking performed over the web. In such a situation the domain of User's search may became very high and the result might not made any context to the User's search interest. This type of search may become a time taking task. So the Search Engines and different Web Services internally provide a Recommender Engine to filter the search from the large domain of data based on the particular user's interest. The Recommender Engine predicts the interest of Users based on User History. A user history may defined by Link it had searched, the product it had bought, the content of particular items viewed by it etc. It can use data or knowledge depending upon the analytics of the engine to predict User's interest.

A Recommendation Process begins with the Front end of the web application sends queries of the user. The analytic engine receives the query and retrieves Usage data of that particular user. The usage data is then processed by the algorithm associated with the analytic engine (Content based or Collaborative filtering technique) to find similar items. The items with higher similarity with respect to user is filtered and provided as recommended opinion to the concern user. (Herlocker, August 1999)

1.3 Characteristics of a good Recommendation System:

A good recommendation system should possess the following characteristic

a. *Performance*: The performance of the recommendation depends on how often the system predicts the interest of the user correctly. It can be determined by the hit ratio (no of correct recommendation/total no of recommendation). A recommendation should predict user's interest at higher rate.
b. *Duration of Recommendation Process*: The recommendation engine should not only predict the interest correctly but predict it timely.
c. *Usage data updation*: The database should be updated time to time about the user's feed. A Recommended System should have different methods to take user's feedback about the items viewed or used. The feedback process should not be lengthy and boring for the user. The feedback shouldn't be biased and user's feedback to be monitored from time to time.
d. *Accuracy of the prediction*: The Mean Absolute Error should be as low as possible for total prediction. The Mean Value for the predictions should be closer to the actual rating.

1.4 Classification of Recommendation System:

Recommendation systems can be broadly classified into 4 different types based on how they predict the interest value for the recommended user. These are:

a. Collaborative Recommendation System
b. Content-Based Recommendation System
c. Hybrid Recommendation System

1.4.1 Collaborative Recommendation System:

A Collaborative system gathers multiple range of information based on user behavior and compares it with other users who have similar behavior. It works on the following:

If x and y have same interest in an item I then x may have more interests in items possessed by y rather than any other user z who doesn't share the similarity.

A collaborative system can be further classified into two different techniques.

I. User-Based Technique
II. Item-based Technique

The correlations between Users or Items are evaluated by different pre-defined formula such as Pearson's Correlation Formula or Manhattan distance or Euclidean distance. The k-nearest neighborhood approach is followed to find the kth most similar user/item with respect to current User/Item.

1.4.1.1User-Based Collaborative System:

If two user u and v have more similar behaviors than any other user w then u must have more interest in items of v rather than items of w.

User Profile is constructed upon users past behavior and rated in numeric value for calculation of similarity. (Herlocker, August 1999)

1.4.1.2 Item-Based Collaborative System:

If Item i1 and Item I2 have more similar interest from various user u1…uk than any other item then item i1 will have similar interest (rating value) to that of item2 more than any other item for an known user t. (B.Sarwar)

The technique is used for personalized recommendation system

1.4.2 Content-Based Recommendation System:

Content-based filtering is used when user is more sensitive to the context of the item or the information it contains. Example: when a user refers to article or journals, it became necessary to compare the content of two articles rather than finding similarity in their preference score. Similarity between user profile and item profile is obtained. The most similar item concept to the user is computed. Bayesian Classification, Term Frequency/Inverse Document Frequency algorithms are used to evaluate the similarity. (Maidel Veronica, 2009)

1.4.3 Hybrid Recommendation System:

A Hybrid Recommendation system embedded different standard recommendation models to produce its output. It can embed Content-based with User-based or Item-based model to create a new type of algorithm.

1.5 Ontology:

Ontology is a method to specify Semantic Knowledge about any context or about a particular domain based on Concepts that defines the domain, Taxonomy or the hierarchical relationship than establishes association between these concepts and the rules that are required to navigate them. (Middleton, 2001)

1. A *Concept* is the set of entities that are present in the domain. i.e- Data Mining is a Concept in the domain of computer science.
2. *Relationship* shows how two concepts are associated with each other, whether one is generalizes the other or specialize it. It is presented in form of Taxonomical tree structure where each concept is a node.
3. *Instances* are the items that are represented by the concept. E.g: C is an instance in programming language Concept.
4. Axioms are the set of rules that navigates the process.

1.5.1 Classification of Ontology:

Ontology can be broadly divided into two categories:

1. Domain Ontology
2. Reference Ontology

1.5.1.1Domain Ontology:

Domain Ontology represents the knowledge using Concept and their relationship in a particular domain of interest by using Hierarchical Concept Structure and instance within the knowledge base. (R., 2000) Rather than dealing with particular atomic entities, user's choice, preference and classifications are associated with the Concept. Domain ontology is derived from reference ontology.

1.5.1.2Reference Ontology:

The standard ontology from which other ontologies are derived is called Reference Ontology. Amazon's Book Taxonomy is an example of Reference Ontology.

1.6 Problem Statement:

A large dataset of User, Item and User Rating data is provided. Use User profiling data and Item profiling data to find similarity between them to create a neighborhood based on their rank. Predict rating/ preference for an item for a user based on Rank matrix. Compare performance of the new model with others.

To develop a Recommendation model for the following:

User Concepts Profile X Item Concepts Profile → *User-Item Similarity*

Similarity (User, Item) → *Ranking (User, Item)*

User-Rating X Ranking (User, Item) → *PredictedRating (User, Item)*

1.7 Our Contributions:

We have created the profile of Items and Users using 12 ontological concepts, evaluated their similarity and ranked them accordingly on their similarity. We have calculated the Hit-Ratio, Mean Squared Error for measurement of different quality attributes of Recommendation model.

We have also hybridized the Content-based approach with that of Item-based collaborative approach to get a new model. We have compared performance and accuracy of the Hybrid model with ontological content based model.

1.8 Organization of Thesis:

- In Chapter-2 we have Literature review where we have included some acknowledged work of different authors in the field of Recommendation System, the motivation for the project etc.
- In Chapter-3 we have proposed work of the project where we have elaborated the complete process of developing an Ontological model and define certain algorithms to measure their performance, similarity and rank them on their similarity.
- In Chapter-4 we have shown the experimental evaluation and results of the proposed work, the comparison with other model on various quality attributes.
- In Chapter-5 we have drawn conclusion based on results and experimental evaluation.

1.9 Applications of Recommendation System in Daily life:

Recommendation System is vastly used in modern day life. It replaces a class of search algorithm. Different news, shopping, books website provides a system to recommend products based on users preference. Facebook provides "People You May Know" options to join person you might know based on your past information. Google uses the navigation data to know your interest and thus filters search based on your context.

1.10 Challenges for a Recommendation System:

Collaborative recommendation system faces *Cold Start problem*, a situation where the recommendation engine is not able to recommend to a new user or newly entered product to the database.

The Sparseness of the data makes the evaluation a time taking and complex task. To check preference of the users who have rated only a small subset of entire item, one has check create matrix for total item set. The power required for the preference computation of million user are too high and needed to be upgraded with increasing no of users. (B.Sarwar)

2 Literature Review:

A collaborative system can be modeled around User (memory based) or Item (Model-based) based algorithms [1,2]. A User-based model [9] compares other user based on their similarity and used this to score the interest for a particular item. While an Item-based model [1] determines the similarity between Items first. Thus prevents it from the bottleneck search for the neighbors in a large population of potential neighbors. A Content-based model [3] however uses the context knowledge of both User and Item and compares similarity between them based on the concepts. It uses semantic knowledge rather than data. A content-based algorithm prevents it from cold start problem or giving preference to the new user or item as only knowledge/content were compared not the ratings or behavioral similarity. It has been used in recommending books and web pages, where content descriptors are available. Hybrid Recommenders [8] combines the collaborative approach with the content-based or semantic knowledge to predict the interest of the user. A Content-based model uses the content or semantic about the user and products to pursue a content-based approach for generating prediction, reasoning what products meets the user's requirement [].given an initial set of concepts and corresponding initial activation values.in our approach the hierarchical structure of an underlying ontology is used explicitly and automatically to determine the concept score.

.

Pearson's correlation formula and Cosine formulas are two important formulas widely used for determining similarity [9].

Similarity can be calculated based on Cosine Similarity

$$Sim(u,v) = \frac{\sum_{i=1}^{n} R(u,i) * R(v,i)}{\sqrt{\sum_{i=1}^{n} R(u,i)^2 \ * \ \sum_{i=1}^{n} R(v,i)^2}}$$

Where:

R(u,i)=Interest value for User u for Item i

R(v,i)=Interest value for User v for item i

Sim(u,v)=Similarity between User u and v

Pearson's Correlation can be used to calculate similarity. (Symeonidis Panagioti.s) (Wouter IJntema)

$$Sim(i,j) = \frac{\sum_{u \in U}\big(Rating(u,i) - AvgRating(i)\big) * \big(Rating(u,j) - AvgRating(j)\big)}{\sqrt{\sum_{u \in U}\big(Rating(u,i) - AvgRating(i)\big)^2} * \sum_{u \in U}\big(Rating(v,i) - AvgRating(i)\big)^2}$$

Where:

AvgRating(i)= Average Preference Score of item i

Rating (u,i)=Preference score user u for item i

U=User Profile

For Content based approach the Similarity formula is defined by (Maidel Veronica, 2009) based on Euclidean distance [4].

The Similarity is measured by Euclidean Distance

$$distance(u,v) = \sqrt{\sum_{j \in C}\big(IS(Cj,u) - IS(Cj,v)\big)}$$

Where u,v=User Concept

Cj is a concept in the Item domain

3 Proposed Work

This section includes brief elaboration of various steps of the modeling process with respective algorithms for the Ontological Content-based system. Profiling User and Items, Scoring the relevancy between concept, calculating Similarity between User and Item based on their concepts are defined.

Content:

1. Concepts User for developing User & Item profile
2. Scoring the relevancy between profile
3. Relevancy Score matrix
4. Similarity calculation between User & Item
5. Ranking of Items based on their concept similarity to user
6. Calculating Hit-Ratio for performance
7. Calculating Mean Squared Error for accuracy

3.1 Concepts in Profiling:

The Process includes Profiling of User and Item based on the concept, defining relationship between them and calculating user item similarity.

The Domain contain 12 concepts each annotated by a number. The concepts are:

1. Children's Play book
2. Mystery and Thriller
3. Programming Books
4. Fiction
5. Fun
6. Thriller
7. Mystery
8. Procedural Lang.
9. Object Oriented Lang.
10. Script Lang
11. Suspense
12. Historical

The Association between the concepts are shown in below figure:

Figure 1: Concept Relationship Hierarchy

3.2 Profiling Item & User:

User Profile contains concepts as attributes with each attributes carries some weights. The Weights represents how many times user viewed item with that concept. It is helpful to know to which concept User is more interested

In Item Profile the Concept attributes holds a binary value. If the attribute have the weight 1 the Item has that Concept as its content. If the weight is 0 , the Item doesn't contain that Concept. (Savia E., 1998)

3.3 Concept Relevancy Scoring:

Concept relevancy is evaluated depending upon the relatedness of concepts in User Profile U and Item Profile I. The degree of relatedness can be of following. (Shoval, 2008)

1. ***Highly Related Concepts:*** If the concepts of U and I are equal/perfect match then score is evaluated to be 1.
2. ***Medium Related Concepts: If*** the concept in I more specifically defines the Concept U then the score is evaluated to 0.66. In case if I more generally defines the concepts of U- score assigned is 0.4.
3. ***Lowly Related Concepts:*** If I much more specifically define U- score assigned is 0.5.And if I much more generally define U then the score assigned is 0.33.

3.4 Concept Relevancy Score Matrix:

The Score Matrix for the Item to User Concept relationship are tabled into a 12 X 12 matrix.

Item\User	Concept1	Concept2	Concept3	Concept4	Concept5	Concept6	Concept7	Concept 8	Concept 9	Concept10	Concept11	Concept 12
Concept1	1	0	0	0.4	0.4	0	0	0	0	0	0	0
Concept2	0	1	0	0	0	0.4	0.4	0	0	0	0	0
Concept3	0	0	1	0	0	0	0	0.4	0.4	0.4	0.33	0.33
Concept4	0.66	0	0	1	0	0	0	0	0	0	0	0
Concept5	0.66	0	0	0	1	0	0	0	0	0	0	0
Concept6	0	0.66	0	0	0	1	0	0	0	0	0.4	0
Concept 7	0	0.66	0	0	0	0	1	0	0	0	0	0.4
Concept 8	0	0	0.66	0	0	0	0	1	0	0	0	0
Concept 9	0	0	0.66	0	0	0	0	0	1	0	0	0
Concept 10	0	0	0.66	0	0	0	0	0	0	1	0	0
Concept 11	0	0.5	0	0	0	0	0.4	0	0	0	1	0
Concept 12	0	0.5	0	0	0	0.4	0	0	0	0	0	1

Figure 2: Concept Relevancy Score Matrix

3.5 Similarity Calculation:

The similarity of an item's profile to a user's profile is based on the number of "Highly Related Concept", "Medium Related Concepts" and "Lowly Related Concepts" between the two profiles, and on the weights of the concepts in the user's profile. The Interest Score computes the value of IS(Ck,Cl) where Ck is concept in Item Concept and Cl is User Concept.

$$IS(Ck, Cl) = \sum_{i=1}^{n} Score(k, l) * ActualRating(i, k)$$

The Similarity is measured by Euclidean Distance

$$distance(u, v) = \sqrt{\sum_{j \in C} \big(IS(Cj, u) - IS(Cj, v)\big)}$$

Where u,v=User Concept

Cj is a concept in the Item domain

3.5.1 Algorithm for Interest Scoring and Calculating Similarity:

The algorithm searches for a "Highly Relevant Concept" or a "Medium Relevant Concept" or a "Lowly Relevant Concept" with respect to a particular Item Concept I and Scores them respectively. The Similarity Score is produces by multiplying the weight of the User Profile for the concept being matched to the score and Actual Rating of user U for Item I (Rating is assumed to be 0 when no rating is present for that user).

Legend:

U: User Profile

I: Item Profile

$Weight_u$: Weight for User U for Concept C_u

ActualRating (u,i)=Rating by User U for Item I

Score (I,j)= Relevancy Score for Concept I w.r.t Concept J

UserItemSimilarity (u,i)=Similarity in concept between User U and Item I

total-no-Concepts= total number of concepts used to define a profile

1. ***Begin**:*
2. *For each user U in Users do following*
    a. *For each item I in Items do following*
        i. *UserItemSimilarity(u,i):=0*
        ii. *totalWeight:=0*
        iii. *For each concepts c in Concepts do following*
            1. *UserItemSimilarity(u,i) +:= Weight(u)*Score(i,u)*ActualRating(u,i)*
            2. *If ActualRating(u,i)!=0 then*
                a. *totalWeight:=totalWeight + Weight(u)*
            3. *End if*
        iv. *End for*
        v. *UserItemSimilarity(u,i):=UserItemSimilarity(u,i)/(totalWeight*total-no-Concepts)*

    b. *End for*
3. *End for*
4. ***End;***

3.6 Rank Matrix Construction:

Similarity of each item is to be computed by using Euclidean Distance Similarity construct. The similarity values are to be stored in User X Item matrix. The value can be used to rank Items for each user based on their similarity value

3.6.1 Algorithm to Rank the Item based on their Similarity to User:

1. For i=0 to Total User -1 do following
    a. Countval := 0;
    b. For j=0 to Total Item-1 do the following
        i. maxVal := UserItemSimilarity[I,0];
        ii. Point :=0
        iii. For k=0 to Total Item-1 do the following
            1. If (maxVal<UserItemSimilarity[I,k]) then
                a. Point:=k;
                b. maxVal := UserItemSimilarity[I,k]
            2. End if
        iv. End for
        v. Countval--;
        vi. UserItemSimilarity[I,point]:= CountVal
        vii. RankingItem[I,j]:=point
    c. End for
2. End for

3.7 Algorithm to determine Hit-Ratio:

The algorithm shows the processes involved in determining the Hit- ratio for the Content-based model.

Legend:

Neighborhood (n): array of n neighborhood where k=10,20, …100

Hit-Ratio (n): array of size n to store hit-ratio in each neighborhood

matchCount: no of correct match in a neighborhood for an item

indexItem= item of a particular ranked item

UserRating [u,i]= User u's actual rating for item i

1. Begin:
2. Count:=0
3. for each n in Neighborhood do the following
    a. Hit-Ratio (Count):=0
    b. temp:=0
    c. for each user u in User do the following
        i. for each item in the Item do the following
            1. for k=0 to n-1 do the following
                a. indexItem:=RankingItem (I,k)
                b. if UserRating(i,j)=UserRating(I, indexItem) then
                    i. matchCount:=matchCount+1
                c. end if
            2. end for
        ii. end for
        iii. temp:= matchCount/(n*total-no-of-Concepts)

iv. if(temp>Hit-Ratio(Count))

1. Hit-Ratio (Count):=temp

v. End if

d. End for

4. End for
5. End

3.8 Algorithm to obtain Mean Squared Error:

It is first required to obtain the Mean Rating for the predicted rating and then that is used for determining the Mean Squared Error. Predicted rating for user u for item i can be obtained from the rating of User u's first ranked item's rating.

Legend:

UserRating (u,i) = Preference of User u for Item i

Rating =total sum of rating

Count=total no of rated items

1. Begin:
2. For each user u in User do the following

a. For each item I in Item do the following

i. //for a particular neighborhood of size n

ii. For k=0 to n-1 in a neighborhood

1. Index:= RankingItem(u,k)//most similar concept
2. If(UserRating(u, Index) !=0) then

a. Rating:= Rating + UserRating(u, Index)

b. Predicted(u,i):=UserRating(u, Index)

c. Count:=Count+1

d. break

3. End if

iii. End for

b. End for

3. End for
4. MeanRating:=MeanRating/Count;
5. Count:=0
6. For each user u in User do the following
    a. For each item i in Item do the following
        i. If(PredictedRating(u,i) !=0) then
            1. Val:= (PredictedRating(u,i)-MeanRating)
            2. MSE:= MSE + Val*Val
            3. Count:=Count+1
        ii. End if
    b. End for
7. End for
8. MSE :=MSE/Count
9. End

4 Experimental Assessment:

4.1 TO Evaluate:

1. How Ontological profile of User and Item can be modeled to recommend items for the specific user?
2. What will be the Performance of the developed model?
3. How accurately the model will predict the interest of User?
4. How will be the models performance as compared to other Models?
5. How accurately Collaborative-Content based model can predict the interest of user?

4.2 Experimental Setting:

4.2.1 Dataset Used:

1. Amazon's Book Crossing (200000 Rating data, 5000user ,7500 items)
2. Unique Code for books: ISBN
3. Unique Code for User :User Id
4. Extracted 5000 book ratings, 150 items, 200 users
5. Build 12 Ontological Concept for profile construction

4.2.2 Ontological User Profile:

Each user contain some score towards each concept (no. of time reference to a particular concept)

4.2.3 Ontological Item Profile:

An Item conveys certain concepts. It is annotated by 1/0 to the respective concept.

4.3 Parameters to be evaluated:

1. **Accuracy of Prediction**:
   a. The accuracy can be measured in term of Mean Squared Error.
2. **Effectiveness in Top-n Recommendation**:
   a. It can be evaluated as a measure of Hit Ratio in a variable range of Neighborhood from 10 to 100.

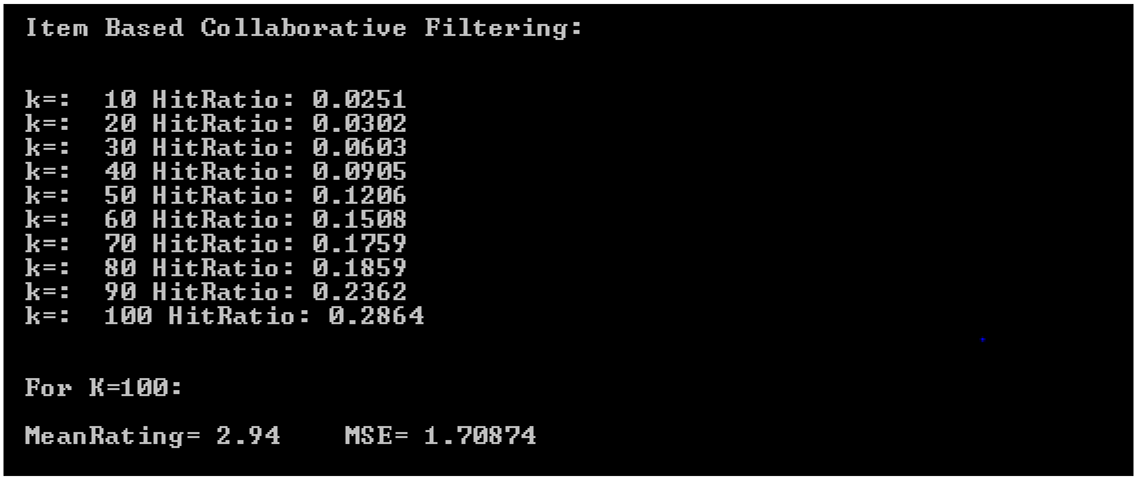
4.4 Results:

4.4.1 User-based Model (Hit-Ratio and Mean Squared Error):

```
User-Based Collaborative Filtering:


k=: 10  HitRatio: 0.0671
k=: 20  HitRatio: 0.1007
k=: 30  HitRatio: 0.1678
k=: 40  HitRatio: 0.1879
k=: 50  HitRatio: 0.2013
k=: 60  HitRatio: 0.2081
k=: 70  HitRatio: 0.2148
k=: 80  HitRatio: 0.2215
k=: 90  HitRatio: 0.2685
k=: 100  HitRatio: 0.3154


For K=100:

MeanRating= 2.95    MSE= 1.71875
```

Figure 3: User-Based Model (Hit Ratio and Mean Squared Error)

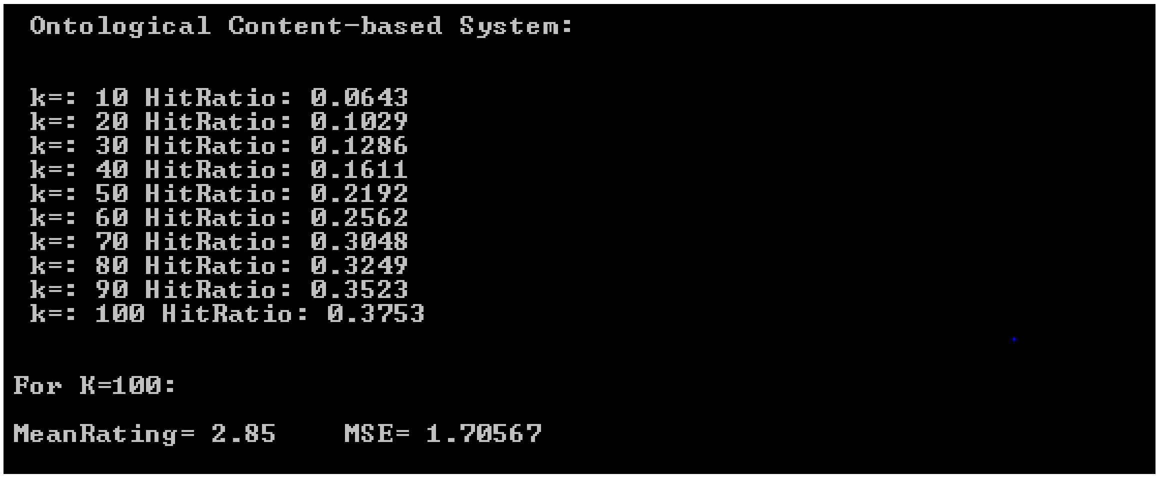
4.4.2 Item-Based Model (Hit-Ratio and Mean Squared Error):



```
Item Based Collaborative Filtering:


k=:  10 HitRatio: 0.0251
k=:  20 HitRatio: 0.0302
k=:  30 HitRatio: 0.0603
k=:  40 HitRatio: 0.0905
k=:  50 HitRatio: 0.1206
k=:  60 HitRatio: 0.1508
k=:  70 HitRatio: 0.1759
k=:  80 HitRatio: 0.1859
k=:  90 HitRatio: 0.2362
k=:  100 HitRatio: 0.2864


For K=100:

MeanRating= 2.94    MSE= 1.70874
```

Figure 4: Item-Based Model (Hit Ratio and Mean Squared Error)

4.4.3 Ontological Content-Based Model (Hit-Ratio and Mean Squared Error):



```
Ontological Content-based System:


k=: 10 HitRatio: 0.0643
k=: 20 HitRatio: 0.1029
k=: 30 HitRatio: 0.1286
k=: 40 HitRatio: 0.1611
k=: 50 HitRatio: 0.2192
k=: 60 HitRatio: 0.2562
k=: 70 HitRatio: 0.3048
k=: 80 HitRatio: 0.3249
k=: 90 HitRatio: 0.3523
k=: 100 HitRatio: 0.3753


For K=100:

MeanRating= 2.85    MSE= 1.70567
```

Figure 5: Ontological Content-Based Model (Hit Ratio and Mean Squared Error)

4.4.4 Hybrid Model (Hit-Ratio and Mean Squared Error):

```
 Collaborative Content-based System:


 k=: 10 HitRatio: 0.0459
 k=: 20 HitRatio: 0.0789
 k=: 30 HitRatio: 0.1152
 k=: 40 HitRatio: 0.1577
 k=: 50 HitRatio: 0.1924
 k=: 60 HitRatio: 0.2315
 k=: 70 HitRatio: 0.2718
 k=: 80 HitRatio: 0.3003
 k=: 90 HitRatio: 0.3317
 k=: 100 HitRatio: 0.3663

For K=100:

MeanRating= 2.83    MSE= 1.72222
```

Figure 6: Hybrid Model (Hit-Ratio and Mean Squared Error)

4.4.5 Performance Comparisons:

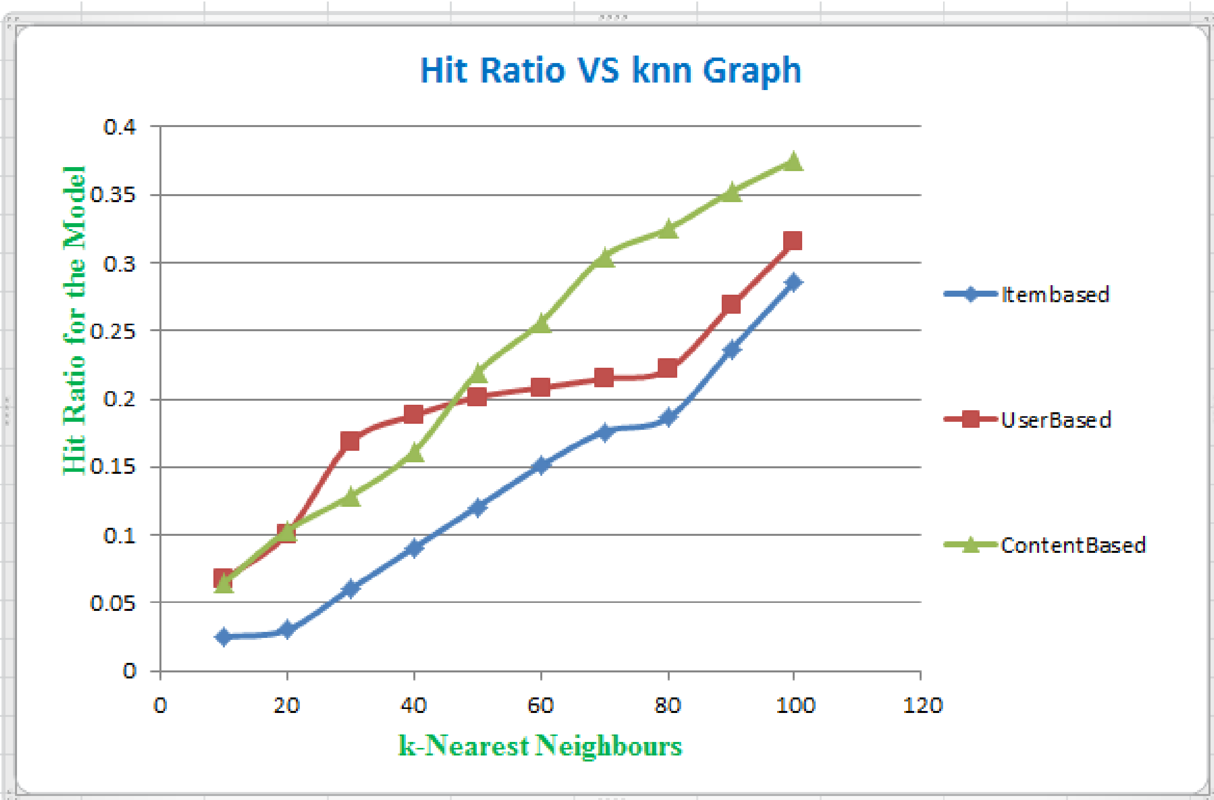


Figure 7: Performance Comparison for different Recommendation Model

4.4.6 Accuracy in Prediction:

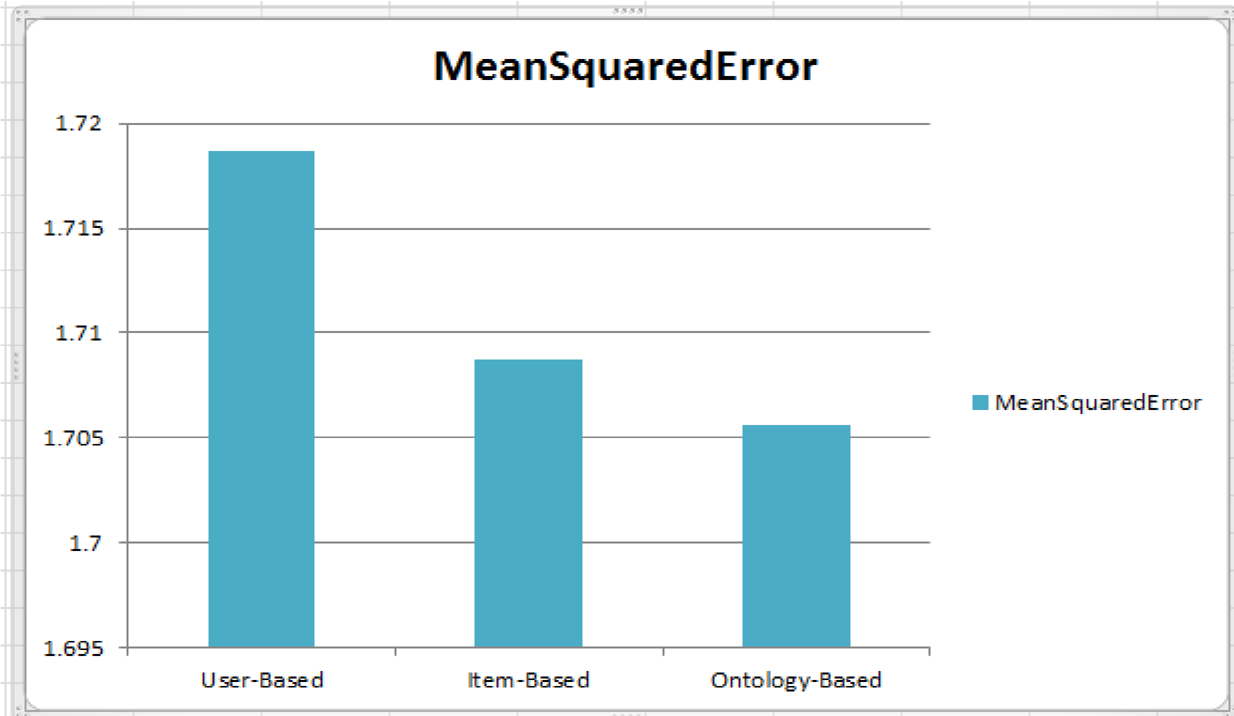


Figure 8: Accuracy comparison for different Recommendation Model

4.4.7 Performance Comparison (Hybrid vs Content-Based):

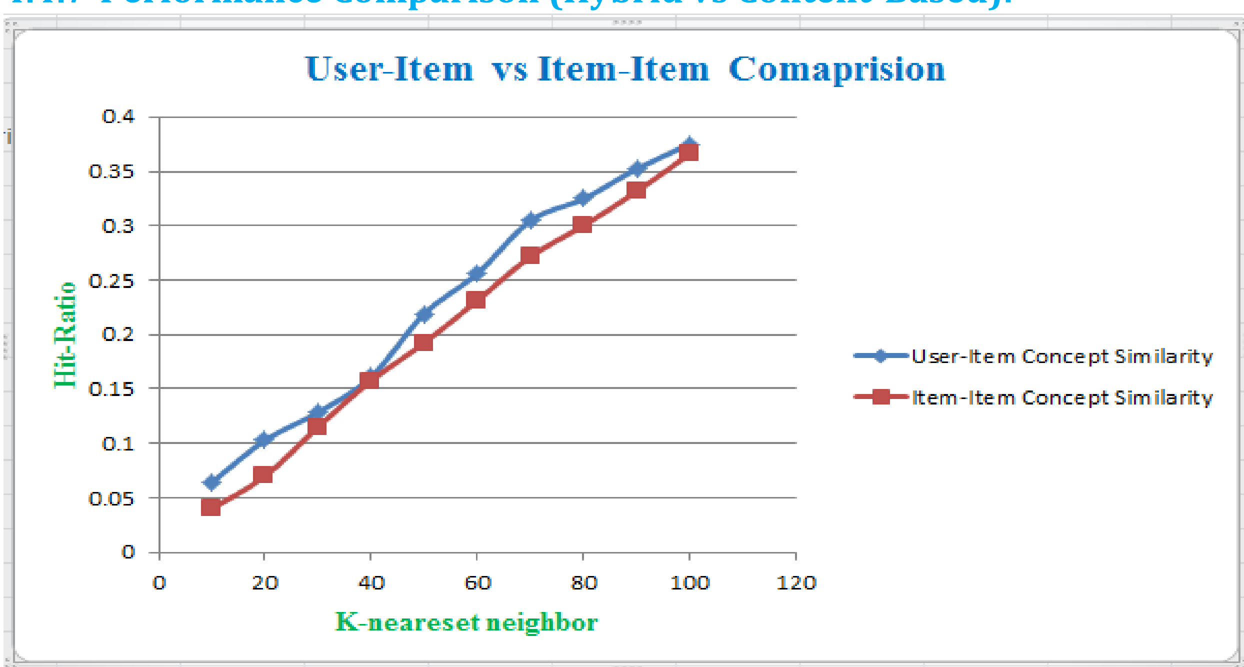


Figure 9: Performance Comparison between Hybrid Model and Content-based model

4.4.8 Accuracy Comparison (Hybrid Model vs Content-based model) :

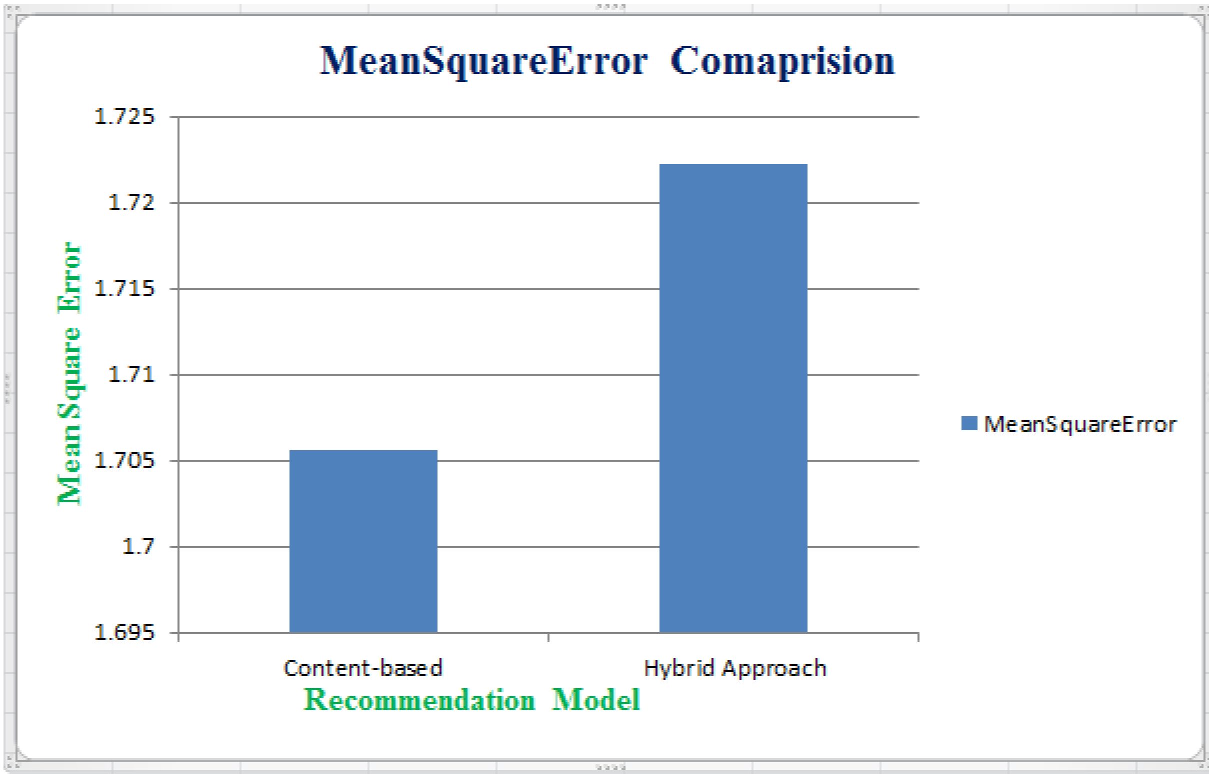


Figure 10: Accuracy Comparisons Between Hybrid Model and Content-based model

5 Conclusions:

- Semantic Knowledge embedded in Ontological Profile can be used for effectively recommending interest of the user.
- The accuracy of the developed Recommendation model has proven to be better than other traditional models. The User's present interest (Weight of each concepts in User Profile) plays the most significant role in giving preference to any given item than any other factors (e.g: similarity in user behavior or similarity between two items).
- The Hybrid model can perform better in predicting as compared to traditional Collaborative System.

6 Bibliography

[1] B.Sarwar, G. (n.d.). *Item based Collaborative Recommendation System.* University of Minnesota, Minneapolis: GroupLens Research group/Army HPC Research Centre Computer Science& Engineering.

[2] Herlocker, J. K. (August 1999). An algorithmic framework forperforming collaborative filtering. *Proceedings of the 22nd ACM Conference.* Berkely,CA.

[3] Maidel Veronica, S. P. (2009, july 2009). Ontological content-based filtering for personalized newspaper, A method and its evaluation.

[4] Middleton, S. D. (2001). Capturing knowledge of user preferences: Ontologies in recommendation system. *Proc. of 1st Int'l Conf. on Knowledge Capture.* 100-107, Victoria, BC, Canada.

[5] R., B. (2000). Knowledge-based recommender systems. In M. D. (2000), *Encyclopedia of Library and Information System, Volume 69.*

[6] Savia E., K. T. (1998). Metadata based matching of documents and user profiles. *Proc. of Finnish Artificial Intelligence Conference, STeP'98.*

[7] Shapira, F. R. (2011). Introduction to recommendation System handbook. In *Introduction to recommendation System handbook* (pp. 1-35). Springer.

[8] Shoval, P. (2008). Ontology Based Content-Collaborative Filtering. *International Journal "Information Theory and Applications"*.

[9] Symeonidis Panagioti.s, A. N. (n.d.). *Collaborative Filtering: Fallacies and Insights in Measuring Similarity.* Aristotle University of Thessaloniki, Dept. of Informatics, 56224, Greece.

[10] Wouter IJntema, G. F. (n.d.). *Ontology based News Recommendation System.* Erasmus University Rotterdam,PO Box 1738, NL-3000,Rotterdam, the Netherlands.