**Item Similarity Learning Methods for Collaborative Filtering Recommender Systems**

***An industry oriented mini project report submitted***

***In partial fulfillment of the requirement for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**

*by*

**A. MANICHANDRA (15131A0505)**

**A. HIMAJA (15131A0508)**

**G.S.S.P.NITIN SHARMA (15131A0557)**

**D.SAI PRAVEEN (15131A0552)**

*Under the esteemed guidance of*

**Mr. Y.V. RAMANJANEYULU**

**Associate professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING(A)

(AFFILIATED TO JNTU, KAKINADA, AP)

VISAKHAPATNAM – 530048

2014 – 2017



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

GAYATRI VIDYA PARISHAD COLLEGE OF ENGINEERING(A)

(AFFILIATED TO JNTU, KAKINADA, AP)

VISAKHAPATNAM – 530048

**CERTIFICATE**

This is to certify that the project work entitled “**Item Similarity Learning Methods for Collaborative Filtering Recommender Systems**” being submitted A.MANICHANDRA (15131A0505) ,A. HIMAJA (15131A0508), G.S.S.P.NITIN SHARMA (15131A0557), D.SAI PRAVEEN (15131A0552) in partial fulfillment of the requirement the award of the degree of “Bachelor of technology” in Computer Science and Engineering is a record of bonafide work done by them under my supervision during the academic year 2016.

**Internal guide Head of the Department**

**Mr.RAMANJANEYULU Dr.P.Krishna Subba Rao**

**Associate professor Professor & Head of Department CSE CSE**

**GVPCOE(A) GVPCOE(A)**

**DECLARATION**

We hereby declare that this is dissertation of our own work except where specifically ask to the contrary and it is not substantially the same as any dissertation which has been submitted to any university.

**By**

A. MANICHANDRA (15131A0505)

A. HIMAJA (15131A0508)

G.S.S.P.NITIN SHARMA (15131A0557)

D.SAI PRAVEEN (15131A0552)

**ACKNOWLEDGEMENT**

We thank **Prof. A.B.K.Rao** principal, **Gayatri Vidya Parishad College of Engineering(A)**  for extending his utmost support and cooperation in providing all the provisions for the successful completion of the project.

We consider it our privilege to express our deepest gratitude to **Dr.P.K.Subba Rao** Professor, Head of the Department, Computer Science and Engineering, for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

We thank MR.Y.V.Ramanjaneyulu sir for providing constant support and guidance through out and helped in completing the project successfully.

We also thank all the members of the staff in Computer Science Engineering for their sustained help in our pursuits.

We thank all those who contributed directly or indirectly in successfully carrying out his work.

**By,**

**A. MANICHANDRA (15131A0505)**

**A. HIMAJA (15131A0508)**

**G.S.S.P.NITIN SHARMA (15131A0557)**

**D.SAI PRAVEEN (15131A0552)**

Item Similarity Learning Methods for Collaborative Filtering Recommender Systems

Abstract

**ABSTRACT**

As one of the most popular recommender technologies, Collaborative Filtering (CF) has been widely deployed in industry due to its simplicity and interpretability. However, it is facing great challenge to generate accurate similarities between users or items because of data sparsity. This will cause second order error in the process of using weighted sum as prediction. To alleviate this problem, we propose several methods to learn more accurate item similarities by minimizing the squared prediction error. This optimization problem is solved using Stochastic Gradient Descent. A comprehensive set of experiments on two real-world datasets at error and classification metrics indicate that the proposed methods can achieve comparable or even better performance than other state-of-the-art recommendation methods of Matrix Factorization, and greatly outperform traditional item based CF method. Besides, the proposed methods inherit the interpretability of item based CF, which makes the recommended results more accessible compared to competing methods of Matrix Factorization.

Index

**INDEX**

pageno

1. ABOUT THE PROJECT…………………………………………...... 12

2. PROJECT SCOPE………………………………………………… … 13

2.1 EXISTING SYSTEM………………………………………….. 14

2.2 PROPOSED SYSTEM………………………………………… 14

3. FEASIBILTY REPORT……………………………………………… 15

3.1 TECHNICAL DESCRIPTION………………………………… 16

3.2 NUMBER OF MODULES……………………………………… 17

3.3 REQUIRED HARDWARE……………………………………… 18

3.4 REQUIRED SOFTWARE……………………………………… 18

3.5 FEASIBILITY TYPES………………………………………… 19

4. ANALYSIS…………………………………………………………… 20

4.1 SRS DOCUMENT……………………………………………… 21

4.2 SCOPE OF DEVELOPMENT………………………………… 22

4.3 ABOUT ANDROID…………………………………………… 23

4.4 EXTENSIBLE MARK UP LANGUAGE ……………………… 28

4.5 CONNECTIVITY ……………………………………………… 32

5. DESIGN…………………………………………………………… 34

5.1 UML SPECIFICATIONS…………………………………………… 37

5.2 SEQUENCE DIAGRAM…………………………………………… 38

5.3 CLASS DIAGRAM………………………………………………… 39

5.4 ACTIVITY DIAGRAM……………………………………………. 40

6. CODING……………………………………………………………… 41

6.1 SAMPLE CODE…………………………………………………….. 42

6.2 OUTPUT SCREENS………………………………………………… 60

7. TESTING………………………………………………………………. 63

7.1 LEVELS OF TESTING……………………………………………… 65

8. CONCLUSION………………………………………………………… 66

9. BIBLIOGRAPHY ……………………………………………………… 68

9.1 REFERENCES……………………………………………………… 69

9.2 LIST OF WEBSITES………………………………………………… 69

About The Project

1. **ABOUT THE PROJECT**

Recommender systems (RS) bring great convenience to people’s life by helping them find the most relevant content from seriously overloaded Web. Nowadays, RSs have been successfully used in many fields, such as video (e.g., Netflix, YouTube), music (e.g., Pandora), book (e.g., Amazon), news (e.g., Digg) etc. Over the years, massive algorithms have been developed to address the recommendation problem . These algorithms make use of the user explicit (e.g. rating) or implicit (e.g. click-through, purchase and review) feedback to construct the user interest model and then make recommendations.

As one of the most promising recommender methods , collaborative filtering (CF) anticipates user’s interests by considering the opinions of those who have similar preferences. Compared to other techniques (e.g., content based methods), typically, CF based methods act only on a user-item rating matrix which is represented by the feedback information. Besides, CF based methods have the capability to expose unexpected items to users, which are not similar to those they have chosen before. This makes them work well in domains where the attribute content of items is difficult to parse, such as musics and videos

To improve the performance of memory based CF without discarding its advantages, new similarity measurement method based on user or item rating statistics was proposed in and Grey Forecast model was used for rating prediction regardless of the similarities

Project Scope

**2. PROJECT SCOPE**

**2.1 Existing System :**

CF based methods can be further classified into two classes: memory based CF methods [6] and model based CF methods Memory based CF contains two popular methods, user based CF [11] and item based CF [12], depending on whether the neighbours are derived by identifying similar users or items. Due to its simplicity and reasonably accurate recommendations, memory based CF has been widely used in industry. However, it suffers from several problems, including data sparsity [13], cold start [14] and data correlation [15], where each user express preference to only a small subset of the available items, and users tend to rate similar items closely

**2.2.Problems in Existing system :**

* . the similarities between users or items cannot be accurately measured by the existing similarity measurement methods, such as Cosine and Pearson Correlation, which will result in inaccurate predictions

**Proposed System:-**

To alleviate this problem, many model based methods are proposed, such as Bayesian belief nets CF models , clustering CF models , Markov decision process based CF models [9] and latent semantic CF models .

In addition, matrix factorization, such as [16], aims to alleviate this problem by reducing the dimensions of user-item rating matrix, then the implicit relationships between items (even those have not been co-rated by one user) can be captured..

Feasibility Report

**3. Feasibility Report**

**3.1 Technical Description:**

**GUI’s** : PYCHARM

**Databases:** MYSQL

**3.2 REQUIRED HARDWARE:**

Hardware requirements

* System : Ubuntu 14.04 or 15.04 or windows 7 and above
* Hard Disk : 1TB
* Ram : Minimum 1GB

**3.3 REQUIRED SOFTWARE:**

Software requirements

* Operating system : Ubuntu 14.04 or 15.04 or windows 7 and above
* Coding Language : PYTHON
* IDE : Pycharm IDE

**3.4 FEASIBILITY TYPES:**

**Technical feasibility:**

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

YES, proposed system is cost efficient and time efficient once implemented on a particular type of data it can be reused many times.

**Financial Feasibility:**

The system as a whole sees a very highly integrated time saving construct and is compatible with any android devices. As the application gets integrated with the Google Play store,the user can easily find his/her location with free of cost once he gets an internet access (for installation).After installation the user requires only the enabled GPS to track his location.

Analysis

Report

**4. ANALYSIS Report**

**4.1 SRS DOCUMENT**

**Intended Audience and Reading Suggestions**

The document is prepared keeping is view of the academic constructs of my Bachelor’s Degree / Masters Degree from university as partial fulfillment of my academic purpose the document specifies the general procedure that that has been followed by me, while the system was studied and developed. The general document was provided by the industry as a reference guide to understand my responsibilities in developing the system, with respect to the requirements that have been pin pointed to get the exact structure of the system as stated by the actual client.

The system as stated by my project leader the actual standards of the specification were desired by conducting a series of interviews and questionnaires. The collected information was organized to form the specification document and then was modeled to suite the standards of the system as intended.

**Document Conventions:**

The overall documents for this project use the recognized modeling standards at the software industries level.

* + The Physical dispense, which state the overall data search for the relational key whereas a transaction is implemented on the wear entities.
  + Unified modeling language concepts to give a generalized blue print for the overall system.
  + The standards of flow charts at the required states that are the functionality of the operations need more concentration.

**4.2 SCOPE OF DEVELOPMENT**

**Future scope:**

• In the near future, it will be installed in Apache Server and so it will be published in internet.

• Datasets will be updated continuously and it will make online actual rating predictions to the users whose habits are changing day by day. As a result, it can be sensitively satisfying current user tastes.

• Web services in particular suffer from producing recommendations of millions of items to millions of users. The time and computational power can even limit the performance of the best hybrid systems. For larger dataset, we can work on scalability problems of recommendation systems.

• The Prediction approach can also be tried in different datasets to test harmony performance of system scalability problems of recommendation systems.

**4.4 PYTHON**

* By using python, we can develop the project specific part is “User Based Collaborative Filtering”.
* Own load the dataset [here](http://files.grouplens.org/datasets/movielens/ml-100k.zip).
* import numpy as np
* import pandas as pd
* You read in the u.data file, which contains the full dataset. You can read a brief description of the dataset [here](http://files.grouplens.org/datasets/movielens/ml-100k-README.txt).
* header = ['user\_id', 'item\_id', 'rating', 'timestamp']
* df = pd.read\_csv('ml-100k/u.data', sep='\t', names=header)
* Get a sneak peek of the first two rows in the dataset. Next, let's count the number of unique users and movies.
* n\_users = df.user\_id.unique().shape[0]
* n\_items = df.item\_id.unique().shape[0]
* print 'Number of users = ' + str(n\_users) + ' | Number of movies = ' + str(n\_items)
* Number of users = 943 | Number of movies = 1682
* You can use the [scikit-learn](http://scikit-learn.org/stable/) library to split the dataset into testing and training. [Cross\_validation.train\_test\_split](http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.train_test_split.html) shuffles and splits the data into two datasets according to the percentage of test examples (test\_size), which in this case is 0.25.
* from sklearn import cross\_validation as cv
* train\_data, test\_data = cv.train\_test\_split(df, test\_size=0.25)

#### 4.5 CONNECTIVITY

* Its going to help the people to identify the object with seeing the image. It should be a revolution to the people who are blind.
* The project is entirely based on “Trust Collaborative system between user and System”.
* The user want to trust the machine is telling Truth.
* Trust in RS is defined as the correlation between similar preference toward the items that are commonly rated or liked by two users. Trust improves RS by combining similarity and trust between users.
* That is, the way neighbors are selected is modified by introducing trust in order to develop new relationship between users so that it can increase connectivity and alleviate the challenges of data sparsity and cold start associated with traditional collaborative filtering techniques
* Different trust metrics are used in RS to measure and calculate the value between users in a network.
* These metrics are of two types, local and global trust metrics.

Design

Document

**5. Design Document**

**5.1 Unified Modeling Language Specifications**

**User Model View:** The UML user model view encompasses the models which define a solution to a problem as understood by the client or stakeholders. This view is often also referred to as the Use Case or scenario view. The main UML model encompassed by this view is the:

**Structural model view:**  Capture static aspects or structure of a system. Structural Diagrams include: Component Diagrams, Object Diagrams, Class Diagrams and Deployment Diagrams.

**Behavioral Model View:** Capture dynamic aspects or behavior of the system. Behavior diagrams include: Use Case Diagrams, State Diagrams, Activity Diagrams and Interaction Diagrams.

**Implementation Model View:** The UML Implementation View combines the structural and behavioural dimensions of the solution realisation or implementation. The view is often also referred to as the component or development view.

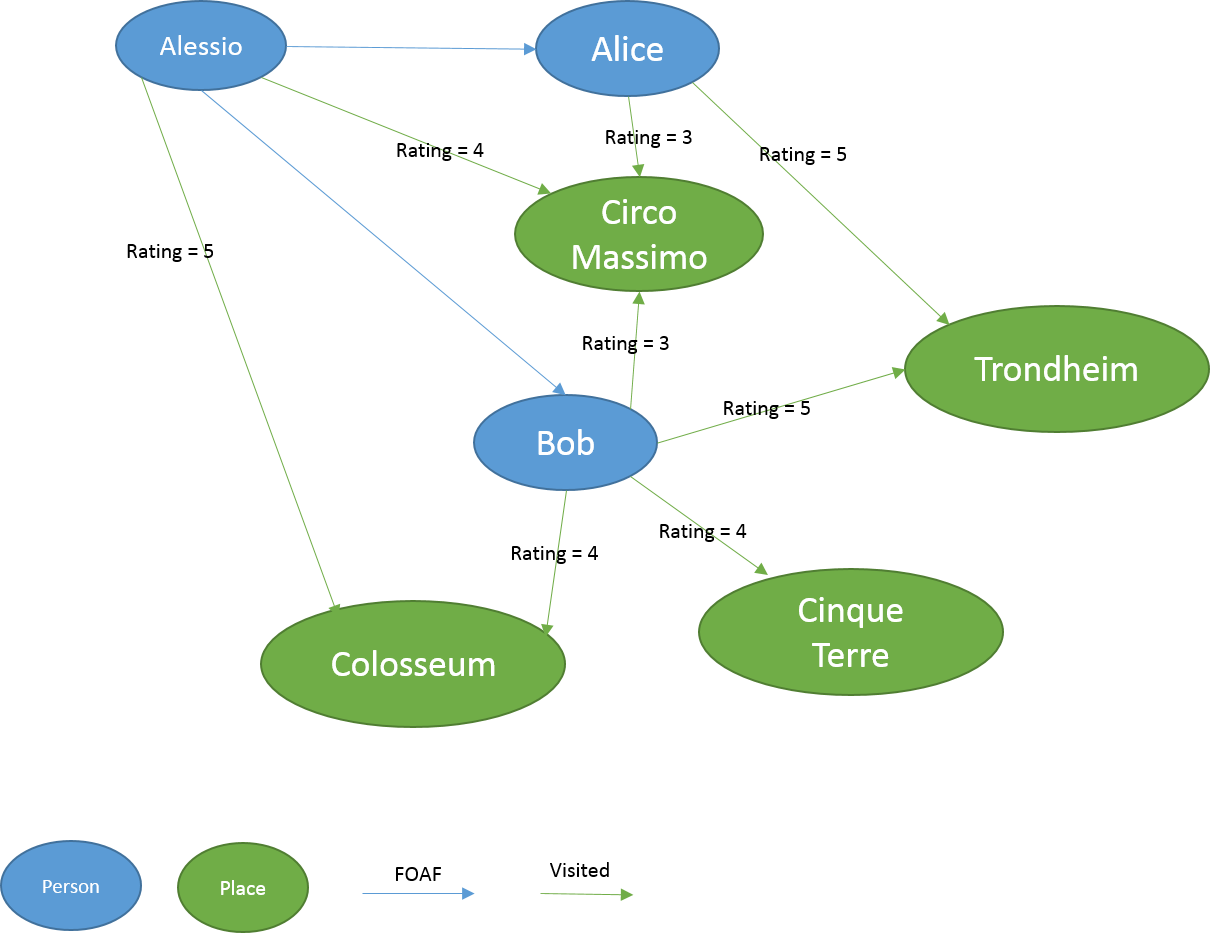
**Environmental Model View:** These UML models describe both structural and behavioural dimensions of the domain or environment in which the solution is implemented. This view is often also referred to as the deployment or physical view.

**UML:**

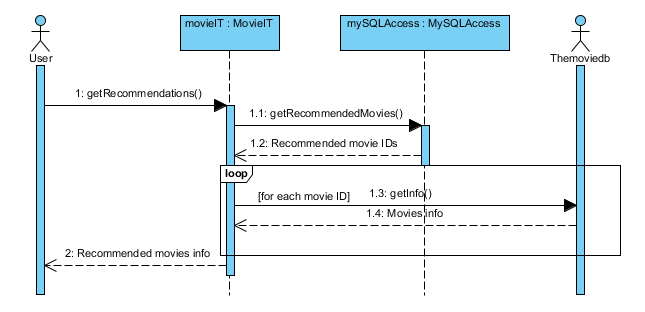
**Unified Modelling Language (UML)** is a general purpose modelling language. The main aim of UML is define a standard way to **visualize** the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is **not a programming language**, it is rather a visual language. We use UML diagrams to portray the **behaviour and structure** of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. Its been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically. Software development is a similar process in many ways. UML has emerged as the software blueprint methodology for the business and system analysis, designers ,programmers and everyone involved in creating and deploying the software system in an enterprise. The UML provides for everyone involved in software development process the vocabulary to communicate about software design.

**USECASEDIAGRAM :**

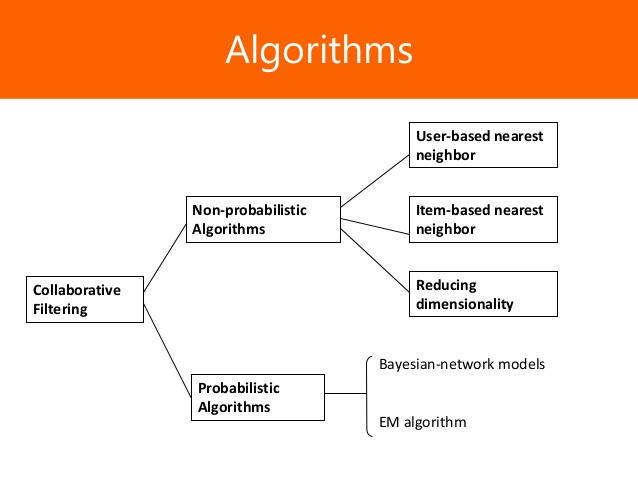


**5.2.SEQUENCE DIAGRAM :**



**5.3 CLASS DIAGRAM :** 

**5.4 ACTIVITY DIAGRAM:**



Coding

**6. CODING**

**MainActivity.java :**

dataset={

'Lisa Rose': {

'Lady in the Water': 2.5,

'Snakes on a Plane': 3.5,

'Just My Luck': 3.0,

'Superman Returns': 3.5,

'You, Me and Dupree': 2.5,

'The Night Listener': 3.0},

'Gene Seymour': {'Lady in the Water': 3.0,

'Snakes on a Plane': 3.5,

'Just My Luck': 1.5,

'Superman Returns': 5.0,

'The Night Listener': 3.0,

'You, Me and Dupree': 3.5},

'Michael Phillips': {'Lady in the Water': 2.5,

'Snakes on a Plane': 3.0,

'Superman Returns': 3.5,

'The Night Listener': 4.0},

'Claudia Puig': {'Snakes on a Plane': 3.5,

'Just My Luck': 3.0,

'The Night Listener': 4.5,

'Superman Returns': 4.0,

'You, Me and Dupree': 2.5},

'Mick LaSalle': {'Lady in the Water': 3.0,

'Snakes on a Plane': 4.0,

'Just My Luck': 2.0,

'Superman Returns': 3.0,

'The Night Listener': 3.0,

'You, Me and Dupree': 2.0},

'Jack Matthews': {'Lady in the Water': 3.0,

'Snakes on a Plane': 4.0,

'The Night Listener': 3.0,

'Superman Returns': 5.0,

'You, Me and Dupree': 3.5},

'Toby': {'Snakes on a Plane':4.5,

'You, Me and Dupree':1.0,

'Superman Returns':4.0}}

File name : **collaborative\_filtering.py**

**# Implementation of collaborative filtering recommendation engine**

**from recommendation\_data import dataset**

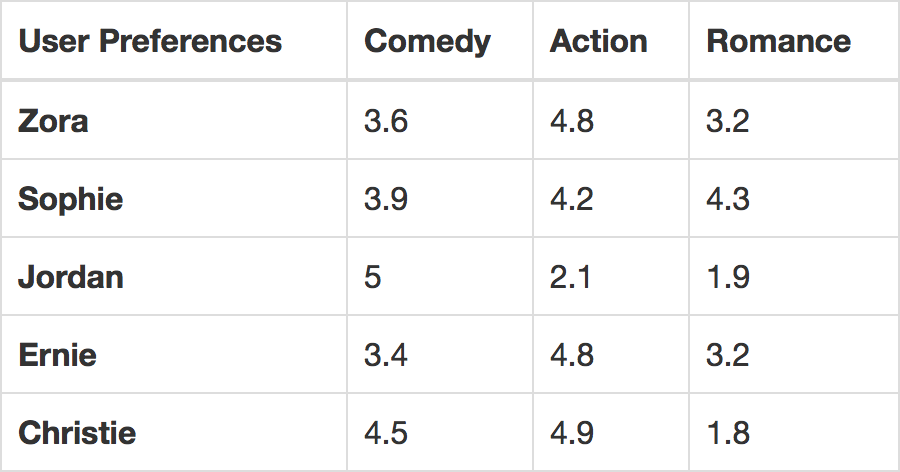
**print "Lisa Rose rating on Lady in the water: {}n".format(dataset['Lisa Rose']['Lady in the Water'])**

**print "Michael Phillips rating on Lady in the water: {}n".format(dataset['Michael Phillips']['Lady in the Water'])**

**print '\*\*\*\*\*\*\*\*\*\*\*\*\*\*Jack Matthews ratings\*\*\*\*\*\*\*\*\*\*\*\*\*\*'**

**print dataset['Jack Matthews']**

**6.2 Output Screen :**



Testing

**7. TESTING**

**Testing:**

Testing is the process of detecting errors. Testing performs a very critical role for quality assurance for ensuring the reliability of software. The result of testing are used later on during maintenance also.

Purpose of Testing:

The aim of testing is often to demonstrate that a program works by by showing that it has no errors. The basic purpose of testing phase is detect the errors that maybe be present in the program. Hence one should not start testing with the intent of showing that a program- works, but the intent should be to show that a program does’t work.

Testing Objectives :

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Stating formally, we say, testing is a process of executing a program with the intent of finding of an error. A successful test is one that uncovers an as at undiscovered error. A good test case is one that has a high probability of finding error, if it exists. The software more or less confirms to be quality and reliable standards

**7.1 Levels of Testing**

In order to uncover the errors present in different phases we have the concept of levels of testing. The basic levels of testing are as shown below…

Unit Testing:-

The philosophy is behind testing is to find errors. Test cases are devised with this in mind. A strategy employed for system testing is code testing.

Code Testing:-

This strategy examines the logic of the program. To follow this method we develop some test data that resulted in executing every instruction in the program and module i.e., every path is tested. Systems are not designed as entire nor or they tested as single systems. To ensure that the coding is perfect two types of testing is performed or for that matter is performed on all systems.

White Box Testing:-

This unit is a testing method where a unit will be taken at a time and tested thoroughly at a statement level to find the maximum possible errors. I tested step wise every piece of code, taking care that every statement in the code is executed at least once . The white box testing is also called glass box testing.

Black Box Testing:-

This testing method considers a module as a single unit and checks the unit at interface and communication with other modules rather than getting in to details at statement level. Here the module will be treated as black box that will take some input and generate output. Output for a given set of input combinations are forwarded to other modules.

CONCLUSION

**8.CONCLUSION**

Due to the inaccurate similarities calculated by current mainstream similarity measurement methods, which may bring second-order error in prediction, we proposed a series of item similarity learning methods to overcome this challenge. Experimental results show that the proposed approaches achieve comparable and even better performance against that achieved by the state-of-the-art methods, matrix factorization, and greatly outperform item based CF which has been widely deployed in industry. Besides, since the proposed methods inherit the interpretability from item based CF, their recommended results are more accessible than that provided by matrix factorization based methods. These advantages drive us to deploy them in the real-world systems, in the future. Moreover, the Asynchronous Distributed Stochastic Gradient Descent technology will be adopted to learn item similarities so that the proposed methods are more applicable.

BIBLIOGRAPHY

**9. BIBLIOGRAPHY**

**9.1 REFERENCES**

[1] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions,” Knowledge and Data Engineering, IEEE Transactions on, vol. 17, no. 6, pp. 734–749, 2005.

[2] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” Knowledge-Based Systems, vol. 46, pp. 109–132, 2013.

[3] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, “Collaborative filtering recommender systems,” Foundations and Trends in Human-Computer Interaction, vol. 4, no. 2, pp. 81–173, 2011.

[4] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in The adaptive web. Springer, 2007, pp. 325–341.

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

[6] J.-M. Yang and K. F. Li, “Recommendation based on rational inferences in collaborative filtering,” Knowledge-Based Systems, vol. 22, no. 1, pp. 105–114, 2009.

**9.2 LIST OF WEBSITES:**

<https://www.sciencedirect.com/science/article/pii/S1110866515000341>

<http://shodhganga.inflibnet.ac.in/bitstream/10603/62351/9/chpt6.pdf>

<https://www.researchgate.net/publication/289493968_Fast_Algorithms_to_Evaluate_Collaborative_Filtering_Recommender_Systems>

<https://www.python.org/>