Mining Technique Defined For Improving User-Based Recommendations in Diverse Environment (MTIURD)

Sangeetha G M, Prasanna Kumar M

Abstract— Recommender systems are being extensively used in the present generation. Today's consumer are facing with millions of goods and services when shopping online. Recommender systems help consumers by making product recommendations that are likely to be of interest to the user such as books, CDs, movies, restaurants, online news articles, and other services. Recommender systems are gradually increasingly harder to find the relevant contents of information in the vast abundant current age of information overload. Thus, recommender systems are needed to help individual users find the most relevant items or products or data sets from an abundant number of choices, collection. Through this gradually increase sales by exposing users to what they might like. E.g. In real time or real world applications consider a product say laptop, the laptop present in numerous patterns with different applications in number depending upon different user's requirements. Thus providing a user or the customer with relevant information about the product as per their requirements with the help of recommender systems would ease the work of an user. Hence we can conclude saying that the volume of information available in the current age is huge to individual users (for e.g., e-commerce sites applications such as Amazon, Netflix) and hence focusing in developing some recommendation techniques within both industry and academia. Most, research to date is focusing on improving the recommendation accuracy i.e. the accuracy with which the recommender system predicts users ratings for items that are yet to be rated. The diversity of recommendation also plays an important role to be considered, it is important to explore the relationship between the accuracy and diversity and also the recommendation quality. Empirical analysis consistently shows the diversity gains of different recommendation techniques which is being used in several real world rating applications or datasets and uses different rating prediction algorithms. Individual users and online content providers will also benefit from the proposed approaches, where in which each user can find more relevant and personalized items or products from accurate and diverse recommendations provided by these recommender systems. These approaches, ranking techniques and algorithms could potentially lead to increased loyalty and sales in e-commerce application sites, thus benefiting the providers as well. Thus, serving these needs can result in greater success regarding cross-selling of related products, up selling, product affinities, and one-to –one promotions, larger baskets and customer retention.

Keywords: Recommender systems, recommendation accuracy, diverse recommendation, empirical analysis, ranking techniques, collaborative filtering, performance evaluation metrics, aggregate diversity, RMSE, extensions of recommendation approaches.

Manuscript received July 09, 2011.

Sangeetha G M, Department of CSE, EWIT, VTU, Bangalore, India.
Prasanna Kumar, Assistant Professor, Department of CSE, EWIT, VTU, Bangalore, India.

I. INTRODUCTION

Recommender systems represent an increasingly popular and important set of personalization technologies that help people to navigate through the vast amount of information. Recommender systems may use either a content-based approach, a collaborative approach, or a hybrid approach that combines both content-based and collaborative methods.

The content-based approach recommends items that are similar to items the user preferred or queried in the past. It relies on product features and textual item descriptions. The collaborative approach or the collaborative filtering approach may consider a users social environment. It recommends item based on the opinions of other customers who have similar tastes or preferences as the user. Recommender systems use a broad range of techniques from information retrieval, statistics, machine learning and data mining to search for similarities among items and customer preferences. Scenarios of using a recommender system. Suppose that you visit the web site of an online bookstore (e.g., Amazon) with the intention of purchasing a book that you have wanted to read. You type in the name of the book. This is not the first time you have visited the web site. You have browsed through it before and even made purchases from it last New Year. The web store remembers your previous visits, having stored click stream information and information regarding your past purchases. The system displays the description and price of the book you have just specified. It compares your interest with other customers having similar interests and recommends additional book titles, saying "customers who bought the book you have specified also bought these other titles as well". From surveying the list, you see another title that sparks your interest and decide to purchase that one as well. Now suppose you go to another online store with the intention of purchasing a digital camera. The system suggests additional items to consider based on previously mined sequential patterns, such as " customers who buy this kind of digital camera are likely to buy a particular brand of printer, memory card, or photo editing software within three months". You decide to buy just the camera, without any additional items. A, week later, you receive coupons from the store regarding the additional items.

The advantage of recommender systems is that they provide personalization for customers of e-commerce, promoting one-to-one marketing. Amazon, a pioneer in the use of collaborative recommender systems, offers "a personalized store for every customer" as part of their marketing strategy. Personalization can benefit both the consumers and the3 company involved These systems try to estimate the ratings of unknown items or products for each user, often based on other users ratings and recommended the items with the highest predicted ratings.

The recommendation problem is basically on ratings i.e. these recommender systems estimate ratings of the items or products that are yet to be used by the users, depending upon the ratings of items previously being used. Much research work is going on developing new algorithm focused on improving the accuracy of recommendations. However, relying on the accuracy alone may not be enough to find the most relevant items or products for a user. The diverse recommendations are also needed to have a highly personalized items as a result and utilize more opportunities for a user to get recommended such items[3]. Over last 10-15 years, recommender systems technologies have been introduced to help people deal with the abundant information[1][2][3][5][6][7][10] and they are widely used in research studies as well as in e-commerce applications i.e. few example Amazon, Netflix. The Netflix prize competition was an open competition held in 2006-09 for the best recommendation algorithms that could improve the accuracy of user's ratings predictions by 10% over Netflix's own recommendation engine.

Two approaches on improving recommender systems ratings in terms of both accuracy and diversity of recommendation has been a main objective of the paper. Initially we propose conventional approaches and then implementation of some new techniques for each new approach for combining these different proposed approaches. The following are the different approaches:

1. Developing new recommendation approaches that can incorporate multi-criteria rating information for more accurate recommendations.
2. Applying heuristic-based (memory based) ranking approaches for more diverse recommendations.
3. Developing more sophisticated optimization approaches for direct diversity maximization.
4. Combining the first two types of approaches i.e. the multi-criteria rating techniques and ranking approaches.

Thus, by the last approach we can overcome the trade-offs between accuracy and diversity [3]. Therefore, resulting in generation of a more accurate and more diverse recommendations as compared to the conventional single rating techniques.

II. RELATED WORK

1. Recommendations Process

Recommendation systems generally perform the following two tasks in order to provide recommendations to each user: 1) unknown ratings prediction and 2) recommendation generation. In many online applications, users provide feedback using numeric values for rating on the items/products/datasets that have been used or purchased or watched. In the prediction rating submitted for a subset of consumed items or products and may be even the information about the item or a product content or user demographics, a recommender system estimate ratings of items or a product that system finds items that the users have not yet been used, using some recommendation algorithms. In case of recommendation phase the system then finds items that maximize the users utility based on the predicted ratings and recommend those to the user. We can define the two phases of recommender system as below: let U be set of users and P be set of products or items available in the recommender systems. Then, the utility function that measures the usefulness or utility of a product or an item to a user can be specified as R: Users * Products--> Ratings, where Ratings represents some numeric value used by users to evaluate each product or item. Therefore, to estimate unknown ratings is the work of the recommender system in the prediction rating phase i.e.:–R*(u, i), based on the known ratings:- R(u, i). here R(u, i) represents actual rating that user u gave to product i and R*(u, i) represents the system predicted ratings for product i that user u has not rated before.

Given all the predictions for each user, now it is the recommendation phase, where the system selects the most relevant products i.e. products that optimize a user's utility, with respect to a certain ranking conditions. Formally, product i_x is ranked ahead of product i_y i.e. ($i_x < i_y$) if rank (i_x) <rank (i_y) where Rank: I→R. This function represents the ranking condition and standard ranking approach refers as follows: $\text{Rank}_{\text{standard (i)}} = R^*(u, i)^{-1}$. here the recommended system rank the candidate products or items by their predicted rating values and recommend the most highly predicted N products to each user because users are generally only interested in most relevant recommendations. And the power of -1 indicates that the products with highest predicted are the ones being recommended to user is given by R*(u, i) ratings.

Utility function measures usefulness or the utility of an item to a user given as;

R: *Users*□ *Items* → *Rating*, where *Rating* usually represents some numeric scale used by the users to evaluate each item. Formally, item *ix* is ranked ahead of item i_y (i.e., i_x i_y) if $rank(i_x) < rank(i_y)$, where $rank: I \rightarrow \mathbf{R}$ is a function representing the ranking criterion. Typical recommender systems rank the candidate items by their *predicted rating values* and recommend the most highly predicted *N* items to each user because users are typically only interested in several of the most relevant recommendations. This is referred to as the *standard ranking approach* given as:

rank Standard(*i*)=*R**(*u*, *i*)-1.

The power of -1 in the above expression indicates that the items with the *highest*-predicted (as opposed to the lowest-predicted) ratings *R**(*u*, *i*) are the ones being recommended to the user.

$$\text{Sim}(u,u') = \frac{\sum_{i \in I(u,u')} R(u,u').R(u,i)}{\sqrt{\sum_{i \in I(u,u')} R(u,u')^2}.\sqrt{\sum_{i \in I(u,u')} R(u,u')^2}}$$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Neighbourhood –based Collaborative Filtering technique:

$$\mathrm{Sim}(u,u') = \frac{\sum_{i \in I(u,u')} R(u,i').R(u,i)}{\sqrt{\sum_{i \in I(u,u')} R(u,i')^2}.\sqrt{\sum_{i \in I(u,u')} R(u,i')^2}}$$

Based on the similarity calculation, set *N*(*u*) of the nearest neighbors of user *u* is obtained. The size of set *N*(*u*) can range anywhere from 1 to |*U*|-1, i.e., all other users in the dataset. Furthermore, *R**(*u*, *i*) – the rating that user *u* would give to item *i* – can be computed as the weighted average of all known ratings *R*(*u'*, *i*), where *u'*€ *N*(*u*) (i.e., user *u'* is ―similar‖ to *u*). Two popular ways to compute this weighted average are as follows:

- Weighted sum approach, i.e.,

$$R^*(u, i) = \frac{\sum_{u' \in N(u)} \mathrm{sim}(u, u').R(u', i)}{\sum_{u' \in N(u)} |\mathrm{sim}(u, u')|}$$

- Adjusted weighted sum approach, i.e.,

$$R^*(u, i) = \overline{R(u)} + \frac{\sum_{u' \in N(u)} \mathrm{sim}(u, u').R(u', i) - \overline{R(u')}}{\sum_{u' \in N(u)} |\mathrm{sim}(u, u')|}$$

2. Recommendation Algorithm for Rating Predictions

The recommendation techniques for rating predictions based on their approaches are classified into 3 categories namely: content- based, collaborative and hybrid approach [1] [10]. Content based recommender systems are user preferred systems in their past. Collaborative filtering recommender systems are the systems with similar preferences to the users have liked in their past are recommended. Hybrid approach as the name says is the combination of both content based and collaborative methods in several ways [8].

Based on the nature of algorithmic techniques also the recommender systems can be classified in following ways: heuristic based and model based approaches. Heuristics based techniques are Re-ranking techniques i.e. the recommendations based directly on the past user activities (e.g. .transactional data, product rating values, movie ratings) [5] [8].here recommender systems use a database about user preferences to predict additional topics or products or items where a new user might like. One of most commonly used techniques is a neighborhood based approach that finds nearest neighbors' that have tastes or choices similar to those of the base user i.e. the target user [2]. In case of model-based techniques use the activities of a previous user in order to initially learn a predictive model by using some statistical or machine learning methods, later these are used to make recommendations e.g. some techniques based on correlation coefficients, vector-based similarity calculations, statistical Bayesian methods[5,][6], matrix factorization, cluster models[5],[6]. The recommendation approaches proposed below can be used in conjunction with any recommendation algorithms and the results are evaluated using the empirical analysis. We use two most popular and extensively used CF techniques for rating predictions, it's a heuristic neighborhood based technique and a model-based matrix factorization technique [5], [2].

III. ACCURACY OF RECOMMENDATION

Several statistical accuracy metrics, such as mean absolute error(MAE) and root mean squared error(RMSE) are presently being used to measure predictive accuracy i.e. how well a system can predict an exact rating value for a specific item. The main objective is to generate top N-Recommendations in terms of both accuracy and diversity and here we have chosen to use decision-support metrics to evaluate how progressively a recommender systems would help a user's to select their relevant products or items from the set of all items or products or database. The decision-support metrics typically work with binary outcome therefore, the percentage of correctly predicted "relevant" products or items among all the sets are used to convert a numeric rating scale into a binary scale i.e.(relevant vs. irrelevant) this is achieved through the empirical analysis tests where we rate either by a 13-ioint(A+ to F) or a 5-ioint or 5-stars scale and the natural assumption is that users provide higher ratings for items or products that are most relevant to their(users) expecting levels. The rating is done between 11 and 13(A+,A,A- on a 13-ioint scale) or 4 and 5(on a 5-ioint scale) as relevant items and items with the lower ratings as irrelevant items) else we can use the threshold value between relevant and irrelevant items as 10.5 to 3.5(say for e.g.) said to be as a relevance threshold(T_H).[12],[1].

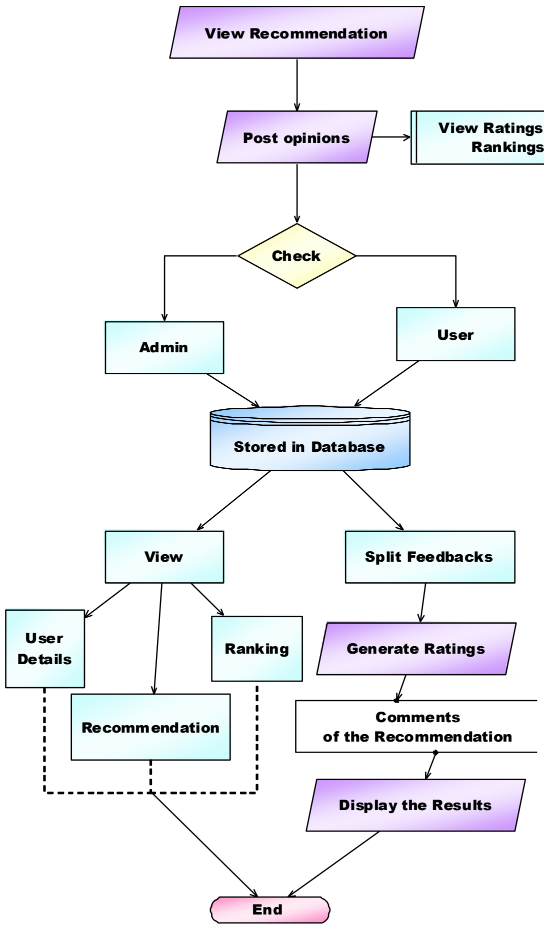
IV. SYSTEM DESIGN

1. Dataflow Diagram:

i. The DFD is also called as bubble chart. It is a simile graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

ii. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

iii. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.



2. *Use case Diagram:*

Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

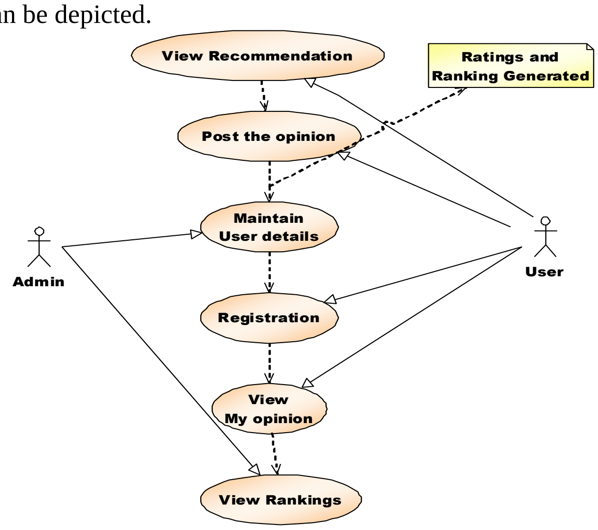
UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML

The Primary goals in the design of the UML are as follows:

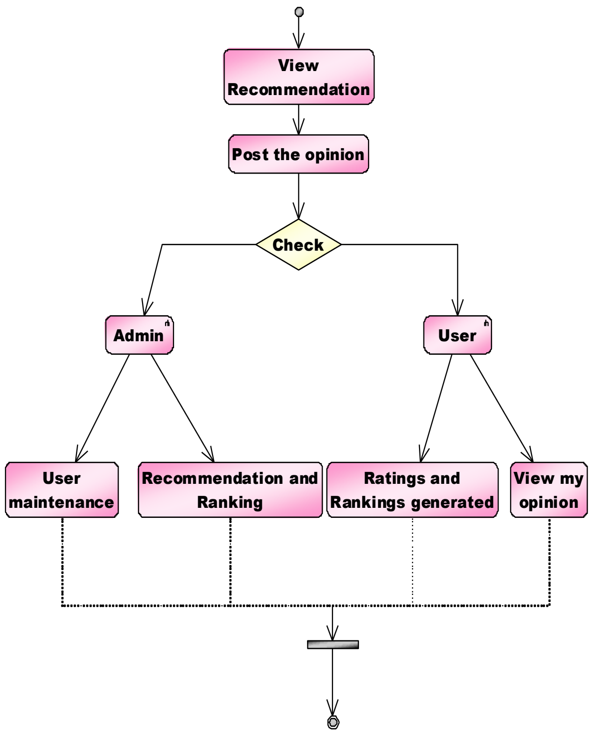
1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core conceits.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development conceits such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



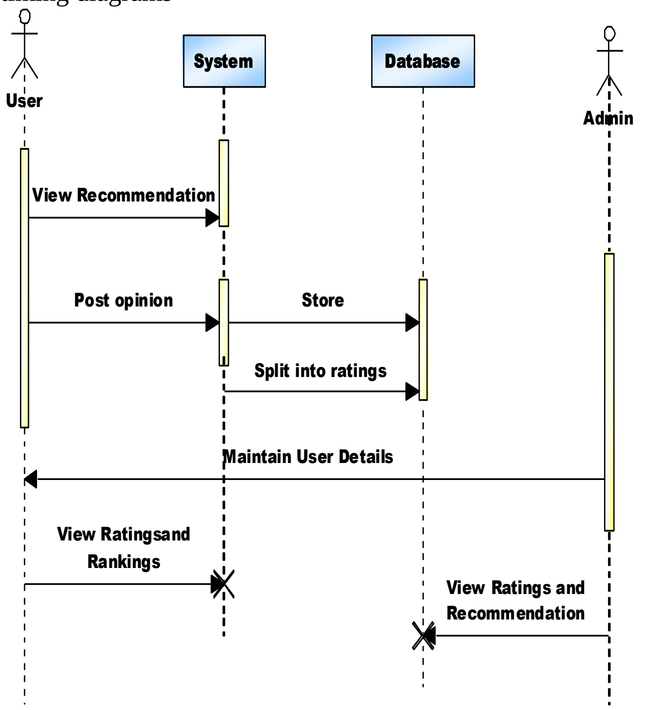
3. *System flow diagram:*

Flowcharts are graphical representations of workflows of stepwise flow and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. A flow chart shows the overall flow of control



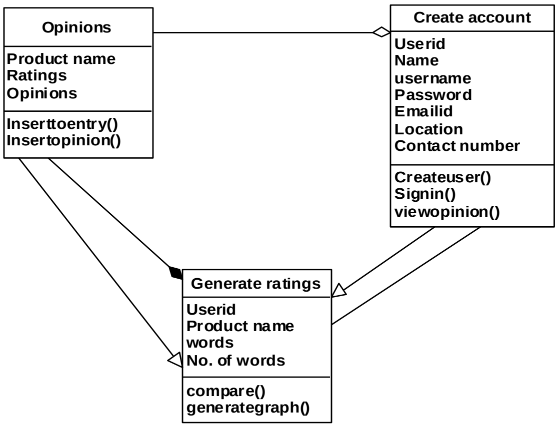
4. Sequence Diagram:

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams



5. Class Diagram:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



VI. SYSTEM STUDY

6.1 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- Economical feasibility
- Technical feasibility
- Social feasibility

6.2 Economical feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

6.3 Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

6.4 Social feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

VII.EXIERIMENTAL RESULTS

The following results define each and every parameter that distinguishes the algorithm being used in it. Differentiation can be done on the parameters like the false positive rate, false negative rate and mean square error etc. We can also include parameters like the correctly classified instances, incorrectly classified instances, kappa statistics, mean absolute error, root mean squared error, relative absolute error, root relative absolute error and total number of instances.

Data set defined for respective algorithms is shown below defined in arff format:

No.	BUGFIXES Numeric	AGE Numeric	LOC_ADDED Numeric	AVE_LOC_ADDED Numeric	REVISONS Numeric	MAX_CODECHURN Numeric	WEIGHTD_AGE Numeric
1	6.0	212.0	50.0	0.4	50.0	11.0	161.179993
2	1.0	197.0	11.0	0.166667	42.0	7.0	188.363632
3	1.0	212.0	25.0	0.322581	31.0	1.0	137.479996
4	4.0	30.0	142.0	0.764706	68.0	42.0	172.788727
5	3.0	150.0	323.0	24.6	10.0	218.0	147.755417
6	2.0	272.0	21.0	1.857143	7.0	1.0	139.523804
7	1.0	205.0	4.0	0.033898	59.0	0.0	173.5
8	0.0	288.0	8.0	0.090909	22.0	1.0	173.0
9	3.0	212.0	65.0	0.68	50.0	32.0	144.446152
10	2.0	212.0	149.0	2.962963	27.0	20.0	132.651001
11	10.0	188.0	189.0	2.366667	30.0	69.0	176.59259
12	2.0	238.0	26.0	0.145161	124.0	9.0	141.461533
13	3.0	212.0	107.0	1.25641	39.0	47.0	83.588783
14	6.0	52.0	36.0	0.25	40.0	5.0	102.0
15	9.0	301.0	205.0	0.837838	74.0	55.0	146.873169
16	2.0	52.0	62.0	0.852941	34.0	15.0	60.387096
17	4.0	186.0	84.0	6.0	10.0	58.0	178.0

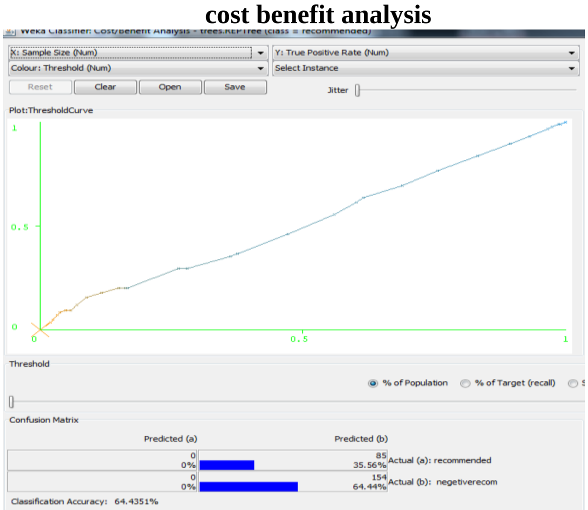
1. Output of RP Tree algorithm:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

The size of the tree to get output is 19 and it takes 0.03 seconds. Here we can see that the correctly classified instances are 65.5983% with instances 239 The TI rate defined is 0.2 and FI rate is 0.123 for class RECOMMENDED and for NEGETIVERECOMMENDED it is 0.877 and 0.8 respectively.

```
Size of the tree : 19

Time taken to build model: 0.03 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         152               63.5983 %
Incorrectly Classified Instances        87               36.4017 %
Kappa statistic                          0.088
Mean absolute error                      0.4335
Root mean squared error                  0.4936
Relative absolute error                 94.4906 %
Root relative squared error            103.0849 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.2       0.123      0.472     0.2       0.281      0.516    recommended
                 0.877     0.8        0.665     0.877     0.756      0.516    negetiverecom
Weighted Avg.    0.636     0.559      0.596     0.636     0.587      0.516

=== Confusion Matrix ===

   a   b   <-- classified as
  17  68 |   a = recommended
  19 135 |   b = negetiverecom
```

2. Output of sample size versus true positive rate cost benefit analysis



3.Output of the cost matrix

Cost Matrix

	Predicted (a)	Predicted (b)	
	0.0	1.0	Actual (a)
	1.0	0.0	Actual (b)

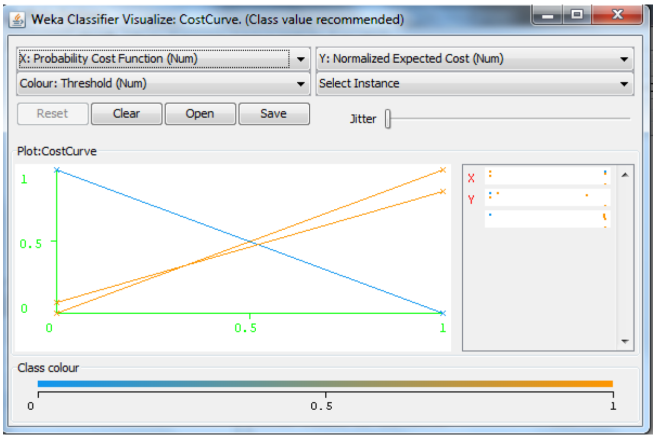
Total Population: 239

4. Output of the SMO algorithm

The time taken to build this model is 0.08 seconds. Here we can see that the correctly classified instances are 65.272% with instances 239. The TI rate defined is 0.153. and FI rate is 0.071 for class RECOMMENDED and for NEGETIVERECOMMENDED it is 0.929 and 0.847 respectively.

```
Time taken to build model: 0.08 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         156               65.272  %
Incorrectly Classified Instances        83               34.728  %
Kappa statistic                          0.0971
Mean absolute error                      0.3473
Root mean squared error                  0.5893
Relative absolute error                 75.694  %
Root relative squared error            123.0777 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.153     0.071      0.542     0.153     0.239      0.541    recommended
                 0.929     0.847      0.665     0.929     0.775      0.541    negetiverecom
Weighted Avg.    0.653     0.571      0.621     0.653     0.584      0.541

=== Confusion Matrix ===

   a   b   <-- classified as
  13  72 |   a = recommended
  11 143 |   b = negetiverecom
```

5. Output of the SMO algorithm cost curve for class value true



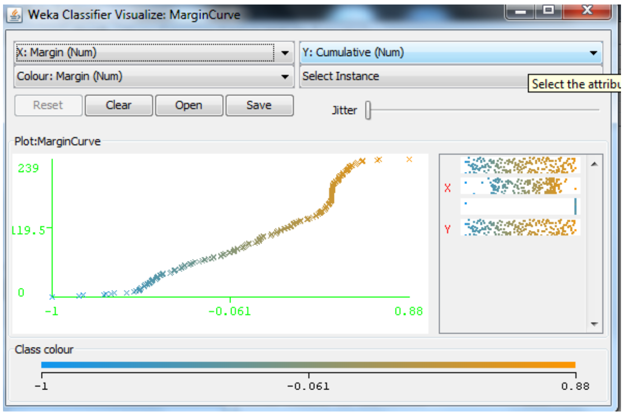
6. Output of the Rotation forest algorithm

Rotation forest takes quite a lot of time to build the algorithm. In this example it has taken: 0.22 seconds and here we can see that the correctly classified instances are 65.272% with instances 239. The TI rate defined is 0.247 and FI rate is 0.123 for class RECOMMENDED and for NEGETIVERECOMMENDED it is 0.877 and 0.753 respectively.

```
Time taken to build model: 0.22 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         156               65.272  %
Incorrectly Classified Instances        83               34.728  %
Kappa statistic                          0.1403
Mean absolute error                      0.4364
Root mean squared error                  0.4802
Relative absolute error                 95.1291 %
Root relative squared error            100.2903 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.247     0.123      0.525     0.247     0.336      0.599    recommended
                 0.877     0.753      0.678     0.877     0.765      0.599    negetiverecom
Weighted Avg.    0.653     0.529      0.624     0.653     0.612      0.599

=== Confusion Matrix ===

   a   b   <-- classified as
  21  64 |   a = recommended
  19 135 |   b = negetiverecom
```

7. Output of the Rotation forest algorithm margin curve (Margin vs. instance number)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication



8.Output of the Bagging algorithm

Here the size of the tree is 23. Here we can see that the incorrectly classified instances are 35.1464% with instances 239. The TI rate defined is 0.318 and FI rate is 0.169 for class RECOMMENDED and for NEGETIVERECOMMENDED it is 0.831 and 0.682 respectively.

```
Size of the tree : 23


Time taken to build model: 0.06 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         155               64.8536 %
Incorrectly Classified Instances        84               35.1464 %
Kappa statistic                          0.1625
Mean absolute error                      0.423
Root mean squared error                  0.476
Relative absolute error                 92.1898 %
Root relative squared error             99.4136 %
Total Number of Instances              239

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.318     0.169      0.509     0.318     0.391      0.606    recommended
                 0.831     0.682      0.688     0.831     0.753      0.606    negetiverecom
Weighted Avg.    0.649     0.5        0.625     0.649     0.624      0.606

=== Confusion Matrix ===

   a   b   <-- classified as
  27  58 |   a = recommended
  26 128 |   b = negetiverecom
```

VIII. DESIGN RULES

8.1 Input design

The input design is the link between the information system and the user. It comprises the specification and procedures for data preparation and those steps are necessary to iut transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simile. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

Input design considered the following things:

1. What data should be given as input?
2. How the data should be arranged or coded?
3. The dialog to guide the operating personnel in providing input.
4. Methods for preparing input validations and steps to follow when error occur.

8.2 Objectives

1 Input design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2 It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3 When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

8.3 Output design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1 Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should identify the specific output that is needed to meet the requirements.

2 Select methods for presenting information.

3 Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

1. Convey information about past activities, current status or projections of the
2. Future.
3. Signal important events, opportunities, problems, or warnings.
4. Trigger an action.
5. Confirm an action.

IX. TRADE-OFFS BETWEEN ACCURACY AND DIVERSITY

We could obtain relatively high accuracy because having trade-offs between accuracy and diversity we can recommend only popular items or products but this could also lead to a decline of other asiects of recommendation diversity e.g. Blockbuster movies, that many users tent to like, and a gadget device with very high-end aiilications with a feasible irice many user tend to iurchase that iroduct or an item.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

But maintaining accuracy while improving diversity leads to a difficult task because higher diversity could be achieved by trying to uncover and recommend highly personalized or idiosyncratic products or items for each user, this leads to a decline in the recommendation accuracy [3]. Consider an example where only popular items or long-tail type products are recommended to users for using from e-commerce application sites (amazon, netflix, flip-kart, movie-lens [1] dataset ratings, here the item based CF techniques are used to predict unknown ratings. As candidate recommendations for each user, consider only the items that were predicted above the ire-defined relevance threshold, in order to ensure acceptable level of accuracy. Among these candidate items for each user we identify items that were rated by many users i.e. Target number of known ratings , as popular items and items that were rated by the least number of users (smallest number of known ratings as long-tail items or products))[1],[2],[3]. As a result we obtain a toi-1 recommendation tasks in a table below i.e. If the system recommends the most popular item, is likely to be the best-selling item i.e. it is far more likely to many users to get the same recommendations. The accuracy measure by the precision-in top-1 metric is 82%, but only 49 popular items out of approx. 2,000 available distinct items were recommended across all users (2,828 users in total). The system can improve the diversity of recommendations from 49 to 695 items by recommending long-tail items to each user.

Quality Metric: / Top-1 recommendation of:	Accuracy	Diversity
Popular Item (item with the largest number of known ratings)	82%	49 distinct items
"Long-Tail" Item (item with the smallest number of known ratings)	68%	695 distinct items

Table-1 - accuracy-diversity trade-offs: empirical example

(note: recommendations (toi-1 item for each user) are generated for 2,828 users among the items that are predicted above the acceptable threshold 3.5 (out of 5), using a standard item-based collaborative filtering technique with 50 neighbors on movie-lens data set.)

Thus, we can say that it is possible to obtain higher diversity by recommending less popular items. Anyhow the loss of accuracy is very negligible (substantial). Therefore, more exploration of new recommendations approaches is necessary to increase the diversity of recommendations with minimal accuracy loss or to increase both diversity and accuracy [1],[2],[5],[12].

X.CONCLUSION

One of the important goals of recommender systems is to recommend to truly like individual users what they would i.e. using different recommendation techniques in order to bring accurate recommendations. Many recommendation algorithms are designed to improve this recommendation accuracy. However, the goal of improving recommendation diversity, which can benefit both individual users, business applications, online content providers and retailers, has been largely ignored in recommender system literature. Thus, we need to explore more towards developing new techniques that can improve both accuracy and diversity by augmenting traditional recommendation techniques. All of the proposed approaches in this paper are general and flexible in that they can build upon a wide variety of existing recommendation techniques.

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," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, ii. 734-749, June 2005.
2. Adomavicius, G., Y. Kwon. 2007. New Recommendation Techniques for Multi-Criteria Rating Systems. IEEE Intelligent Systems 22:3 48-55.
3. Adomavicius, G., Y. Kwon. 2009. Toward More Diverse Recommendations: Item Re-Ranking Methods for Recommender Systems. Iroc. of the 19th Workshop on Information Technologies and Systems.
4. Adomavicius, G., Y. Kwon. 2011. Improving Aggregate Recommendation Diversity Using Ranking-Based Techniques. IEEE Transactions on Knowledge and Data Engineering Forthcoming.
5. Adomavicius, G., N. Manouselis, Y. Kwon. 2011. Multi-Criteria Recommender Systems. in I. B. Kantor, F. Ricci, L. Rokach, B. Shaiira (Eds.). Recommender Systems Handbook: A cGuide for Research Scientists and practitioners Chapter 24. Springer.
6. Aggarwal, C.C., J.L. Wolf, K.L. Wu, I.S. Yu. 1999. Horting Hatches An Egg: A New Graph-Theoretic Approach to Collaborative Filtering. Proc. of the 5th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining (KDD'99). 201-212.
7. Ahuja, R.K., T.L. Magnanti, J.B. Orlin. 1993. Network Flows: Theory, Algorithms, and Applications. Englewood Cliffs, NJ: Prentice-Hall.
8. Anderson, C. 2006. The Long Tail. New York: Hyperion.
9. Balabanovic, M., Y. Shoham. 1997. Fab: Content-Based, Collaborative Recommendation. Communications of the ACM 40:3 66-72.
10. Bichler, M. 2000. An Experimental Analysis of Multi-Attribute Auctions. Decision Support Systems 29:10 249-268.
11. Billsus, D., M. Iazzani. 1998. Learning Collaborative Information Filters. Iroc.Int'l Conf. Machine R. Bell, Y. Koren, and C. Volinsky, "The BellKor Solution to the Netflix Irize,"www.netflixirize.com/assets/IrogressIrize KorBell.idf, 2007.
12. C. Anderson, The Long Tail. Hyierion, 2006
13. S. Breese, D. Heckerman, and C. Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Proc. 14th Conf.Uncertainty in Artificial Intelligence, 1998.
14. R.M. Bell, Y. Koren, and C. Volinsky, "The Bellkor 2008 Solution to the Netflix Prize," htti://www.research.att.com/~volinsky/ netflix/IrogressIrize2008BellKorSolution.idf, 2008.
15. J. Bennett and S. Lanning, "The Netflix Prize," Proc. KDD-Cui and Workshop at the 13th ACM SIGKDD Int'l Conf. Knowledge and Data Mining, 2007.
16. D. Billsus and M. Iazzani, "Learning Collaborative Information Filters," Proc. Int'l Conf. Machine Learning, 1998.
17. S. Zhang, W. Wang, J. Ford, F. Makedon, and J. Iearlman, "Using Singular Value Decomposition Approximation for Collaborative Filtering," Proc. Seventh IEEE Int'l Conf. E-Commerce Technology
18. W. Knight, "Info-Mania' Dents IQ More than Marijuana.".NewScientist.comNews, htti://www.newscientist.com/article.ns?id=dn7298, 2005.
19. Y. Koren, "Collaborative Filtering with Temporal Dynamics," Proc. 15th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, ii. 447-456, 2009.
20. B.M. Sarwar, G. Karyiis, J. Konstan, and J. Riedl, "Analysis of Recommender Algorithms for E-Commerce," Proc. ACM Conf. Electronic Commerce, ii. 158-167, 2000
21. Hemant Palivela, Pushpavathi Thotadara ,Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on computing communication and network technologies, 26th -27th July 2012.