ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/107979, December 2016 ISSN (Online) : 0974-5645

Enhancing Stability of Recommender System: An Ensemble based Information Retrieval Approach

N. Saipraba* and V. Subramaniyaswamy

School of Computing, SASTRA University, Thirumalaisamudram, Thanjavur – 613401, Tamilnadu, India; sai.prabamurthy@gmail.com, vsubramaniyaswamy@gmail.com

Abstract

Objectives: Collaborative filtering is popularly used for providing recommendation services. These services are provided by numerous different recommendation algorithms that are proved to be effective. **Methods:** The Stability of recommender systems are now becoming the interesting component in the fields of research. It becomes necessary to evaluate the consistency of the predictions to retain users trust on the system. Stability is a measure of consistency level that, certain recommendation algorithms possess. We work with three ensemble techniques, Boosting, Bagging and Smoothing which helps in improving the stability and providing greatly personalized predictions as outcome. Due to the nature of ensemble techniques of filtering the outliers from data, these techniques are employed for improving accuracy also. The stability is computed in 2 phases as it is proved to be experimentally efficient. **Findings:** This paper analyses the stability factor of the outcome and the impacts on varying the data quantity. We also analyze the impact on stability due to the variations made to the data under evaluation. **Applications:** This approach makes us understand the performance metrics and quality of any recommender system being used.

Keywords: Bagging, Boosting, Collaborative Filtering, Recommender Systems, Smoothing and Recommendation Algorithms, Stability

1. Introduction

As the use of internet grows rapidly day by day, the data employed also gets increased gradually. Hence it is in need that we generally work on the huge data across network. Recommender system is a helpful technique that assists people to choose what best suits their desire, from the overloaded objects present over internet[1]. A recommendation method often tries to calculate the rating of the unknown items referring to the known things that user has previously rated. It takes into account the interests of like-natured people[1]. Also, it is necessary to provide good recommendations for users to retain the trust of user on the system's prediction. They add up benefits to both user and industries. The ways the input is gathered are using the ratings given by the user or by monitoring the user's behaviour. These recommender systems are generally of three major types: 1. content-based filtering, 2. collaborative filtering and 3. hybrid filtering. Content based filtering works on the product[2] combination of features of both content-based and collaborative filtering methods. The study of quality-of-service, a fundamental element deeply describes the interaction behaviour between the user and the services[3]. There are numerous algorithms being employed in literature. These algorithms are often evaluated before choosing the one that suits our task. Moreover, there are few properties of recommender systems that describe every algorithm's unique features, domain to which it can be applied, its merits and demerits. It is important to analyse the properties of the algorithm before employing it. Stability can be estimated by the amount of time period to which the prediction made by the algorithm remains constant even after the addition of new rating (that are completely in agreement to the system's previous outcome)[4]. Prior results explore the fact that model based prediction techniques (Matrix factorization, User based collaborative filtering etc...) provide eminent stability than memory-based collaborative filtering techniques (Completely data driven). Generally, 2-phase approach is often used for

** Author for correspondence*

computing stability. It divides the entire unknown data into training data (80%) and validation data (20%). This approach involves 2 phases worked out one after the other. Ensemble techniques are those that work on the input data, manipulate it, configure it and filter out the outliers from it[5]. In literature, two methods that help in improving the stability when used in conjunction to the popular recommendation algorithms were proposed**.** Initially, bagging is used on to the user-item ratings available which divide it into sub-samples. These samples are then employed with different prediction models involving weight as a factor. Thus, the outcome of this approach would be with greater stability. Another approach that involves various iterations called iterative smoothing is also employed in improving stability[6]. By analysing the experimental results we notice the enhancement of stability on employing these approaches. In my proposed work another ensemble technique named boosting is being employed. Iterative smoothing employed in prior work involving various iterations to enhance the stability can be reduced with delta modifications to the process. Our paper includes the analysis of work that has been done previously related to the stability of recommender systems. Section 2 deals about the problem of instability and the way of stability computation. The proposed model and used algorithms are explained in section 3.Section 4 deals about the experimental setup that is being proposed to improve stability.

2. Problem Definition

Stability is one of the most important measures of shift in predictions made by a system. The problem of instability is explained with an example below. Consider a recommender system with 5 users and 8 movies present as shown in the Figure 1. All users have given their ratings to movies they have watched previously as given in the figures below. Consider U3 had rated the first 4 movies as 5, 2, 3 and 4 respectively that is as same as the ratings given by U1 and U2 for the same 4 movies. Hence, other movies (M5, M6, M7, and M8) that are rated by those similar users are recommended to target user with ratings 5,3,2 and 4 respectively as shown in the Figure 1.

Consider U3 had watched movies 5, 6, 8 and had rated as 5, 3 and 4 respectively which is similar to U4 and U5.Thus, after submitting these 3 ratings the system now recommends M7 with rating 5 to target user taking into account the ratings of U4 and U5, as in Figure 2. This shows the instability of that recommender system. Figure 3 explains the type of line and the corresponding rating value it indicates (For example U2 had rated M3 as 3 which is indicated using a dotted line type). The following section 2.1 helps in understanding the estimation procedure for stability.

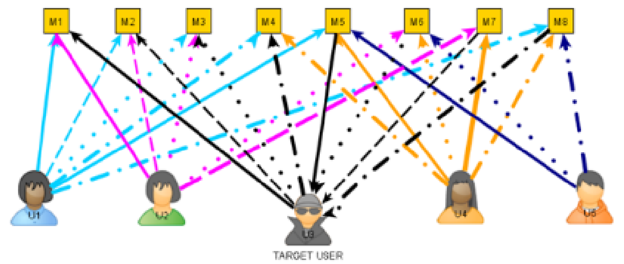

Figure 1. Recommendation before new ratings.

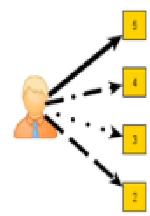

Figure 2. Indication of ratings.

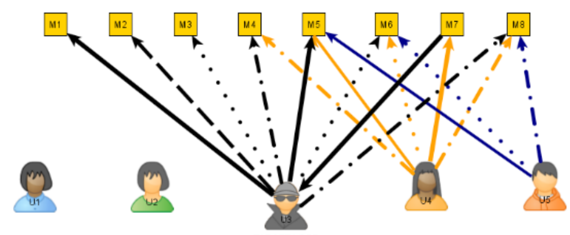

Figure 3. Recommendation after new ratings.

2.1 Stability Estimation

In order to measure the recommendation stability the 2-phase approach is used as in Figure 4. In the first phase, the known ratings are given as input onto the prediction model and the predicted ratings are taken as the outcome[7]. The ratings predicted during this phase are considered as the initial predicted ratings. During the second phase, a random sample of the predicted ratings from the previous phase is fed as the input to other prediction model finding the final prediction ratings. These final prediction ratings are used to find the stability which is the computed by the difference between the initial and final prediction ratings. Lower the difference between the predictions higher the stability factor.

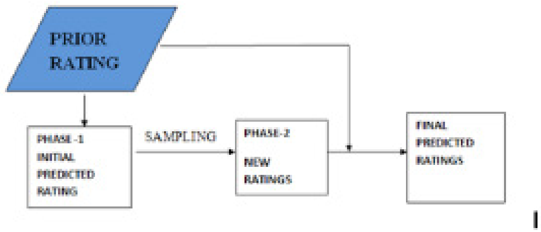

Figure 4. 2-Phase stability estimation.

3. Proposed Methodology

To bring out stable recommendations, involving powerful recommendation algorithms only would not be necessarily enough. Moreover the trust of user on the recommendation system seems to get reduced due to weaker stability. Hence, it is really needed to provide stable recommendations. In existing two scalable meta-algorithmic approaches were used in conjunction with 6 recommendation algorithms which is also explained in the following section. Employing few ensemble techniques in conjunction to recommendation algorithms would help in improving the stability factor. The following section 3.1 explains the existing and proposed ensemble techniques that are used to enhance stability of predictions.

3.1 Ensemble Methods

Ensemble learning is a method that is widely used in solving computationally complex problems. It increases the quality of the system and involves a combined learning methodology. It works efficiently by holding together the strong classifiers and intelligent experts in bring out the solution.

3.1.1 Bagging

Bagging (Bootstrap aggregation) is a parallel ensemble technique that works on the concept of voting classification. It decreases the variance in the predicted outcome by including more clear data on to the original data being employed. It simplifies the combinations and repetitions present in the data and produce multiple sets that are being used for manipulation. Bagging deals about models build independently[8]. It works uniting the classification predicted from various models and employs it onto different data which is shown in Figure 5. Bagging is suitable for high variance and low bias models. It also reduces the correlation between data. One of the most popular examples of tree based model that works well on applying bagging is Random Forest. Bagging is also applied to models that work only on features with equal gain. It is needed that we sub-sample the data again and again by adjusting the sample as desired. Bagging takes the combination of bias and variance into account. Bias is simply a deviation that occurs because of learning algorithm whereas variance is due to learning model. Bootstrap aggregation also finds the instability of outputs on including into complex models to small data. Bagging involves creation of certain number of data set sample that also includes replacement at needed cases as desired which is then employed using learning algorithms with each sample for single model. Thus, it results in generation of classifier. A simple voting scheme is applied to all the classifiers generated and the final classifier is that with equal weight. At some cases weighted combination of made predictions would also be helpful[9]. Prior research on tree based bagging model explores that different trees might be grown sometimes on different samples that evident the instability on prediction. The classifiers thus generated are used to make decisions of the unknown objects. The final decision is that which is most often being predicted. Certain results that are recently found in statistical learning theory explain that bagging can effectively be used for knowing about stable machines. Bagging at some cases degrades the performance of stable procedures when the estimator is ready getting improved. In our proposed system adaptive bagging is used and is generally easy to implement and is used for improving the stability.

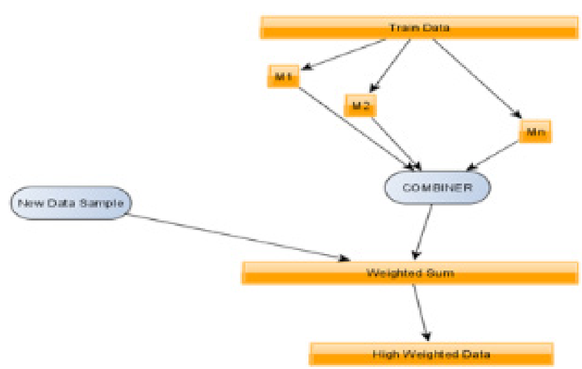

Figure 5. Bagging technique.

3.1.2 Boosting

Boosting is a sequential ensemble technique that aims to decrease the bias present in the model. It tries adding new models to the procedure at times where the prior model employed lacks in performance. Boosting empowers the machine learning algorithms with greater accurate predictions. It is directly designed with work with classification problems but can be extended to regression at times of need. It is suitable for low variance and high bias models. It motivates a procedure to combine outcomes of many weak classifiers to form a strong group. One of the commonly used tree based model that employs boosting is gradient boosting. Boosting improves the effectiveness of the learned function used[10]. Boosting employs a method onto the small subset of data used for observation as depicted in Figure 6. For every outcome an equal weight is assigned to it. All the predictions are calculated and weights are assigned depending on the way the classification occurs[11]. In other words, more weight is applied to observations that were hard to classify, and less weights for those that were quite easy to classify. Furthermore, the next iteration is carried out by re-assigning weights to observations if necessary for possible accuracy improvements. The errors in the previous classification step were corrected at the current step, thus reducing the falsehood of outcome[12].

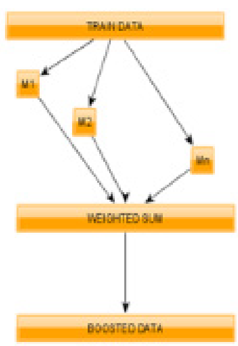

Figure 6. Boosting technique.

3.1.3 Smoothing

Smoothing is a data mining technique which helps in removing the noise by making the valid aspects to withhold. Smoothing identifies the outliers present and eliminates it throughout. There are many types of smoothing present in history. They are Random, Random Walk, Moving Average, Exponential smoothing, Seasonal smoothing, Simple Exponential[12]. Exponential smoothing is a flexible technique which is used widely in many domains. Smoothing is directly employed in improving stability. In literature, the smoothing technique is employed in iterations and is collectively known as iterative smoothing. The output of the first stage is given as the input of the next iteration. Each prediction made at iteration is modified and is made stable. Figure 7 depicts the complete iterative smoothing method. For iteration, different prediction model is being employed. This makes it quite difficult and involves tedious process[13]. Hence, scalable iterative smoothing is being used for manipulation. This technique involves only one prediction model build for all the n iterations employed. The number of iterations employed and the sample size for smoothing depends upon the domain in which the applied task works.

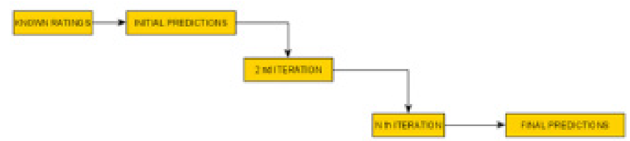

Figure 7. Iterative smoothing technique.

3.2 Prediction Models

There were number of recommendation algorithms employed in literature[14]. In our paper we have worked with 6 popular recommendations as mentioned in the Table 1. Memory-based algorithms every time recommend an item considering the entire database. Hence, the performance decreases with sparse data.

Table 1. Recommendation algorithms

Algorithm	Description
Simple user Average	Predicts the ratings considering the average rating for items by user.
Simple Item Average	Predicts the ratings considering the average rating value given by users to a particular item.
User-Based CF	Analyses the similar users of an item and recommends items they have highly rated.
Item-Based CF	Analyses the similar items that are highly rated with similar features and recommends those items.
Matrix Factorization (SVD)	Rating matrix is divided into two thus, associating every user and item internally. Taking product between matrices gives the accurate predictions.
Bayesian Probablistic Matrix Factorization	BPMS eliminates the Gaussian noise, fills in the missing values and minimizes the RMSE value.

3.2.1 User Based Collaborative Filtering

User based Collaborative Filtering is a recommendation algorithm that takes into account the prior ratings of the neighbours closer to desired user, to make recommendations[15]. It analyses the ratings of the users who have rated as likely as our user for items that were already been rated by our user. User based algorithms were proved to be very successful in literature. User based-collaborative filtering looks on the heuristic that more users with similar ratings for few items would rate in similar mode for other items also[16]. Adding together the values of closest neighbours gains the unknown rating for a particular item that is to be recommended for a particular user. The common approach for aggregating the ratings of similar users is done by finding out the sum of similarity weight between our target users's rating and their neighbour's rating. Every user profile would generally be sorted in an order considering the dissimilarity it shows towards our user profile before grouping the similar users. This similarity between ratings of the users can be estimated by various popular measures and some are Euclidean distance, Minkowski distance, Mahalanobis distance, cosine similarity. Cosine similarity and Pearson correlation are two most widely used similarity approaches that help in finding similar users and is proved efficient[8]. The proposed work uses the cosine similarity which finds the similarity using the cosine angle between the two ratings.

$$\cos(p1, p2) = [p1.p2]\frac{p1.p2}{\| p1 | . | p2 \|} \tag{1}$$

The formula is used to calculate the cosine similarity between two ratings P1 and P2.

3.2.2 Item-Based-Collaborative Filtering

Item based collaborative filtering or item to item filtering is one the widely used collaborative filtering techniques. It finds the similarity between objects present. Item based CF is a lesser dynamic technique[17]. The prior user based filtering works on the similar users of the focused user whereas item based filtering works on the similar items to the items our target user have already rated. This type of filtering technique is used by Amazon[18]. It works effectively even if the user profile changes often which are the major drawback of user based filtering. There is no issue in employing this type of filtering if the items used are sparse in nature. It uses the linear regression or weighted sum process for normalizing and filtering the noise data present. It involves lesser error than user based collaborative filtering. Item based collaborative filtering provides greater performance due to its nature of lesser dynamic state[19]. Since, it works on finding the item the target user has rated previously, finding out the similarity is quite important. It uses many similarity measures such as Cosine based similarity, Correlation based similarity, Adjusted cosine similarity etc. Item based collaborative filtering is considered more accurate than user based collaborative filtering. The primary requirement for this algorithm is that at least 3 users should have rated similar for 2 similar items. Many works in past have been carried out in increasing the scalability of the items on employing this prediction model. It involves two main processes that are to be carried out; they are the similarity computation and the generation of predictions. It is proved in literature that Item-Item Collaborative Filtering is more accurate, Scalable and flexible in nature[20]. Now a day item Based CF is the most popularly used tool in E-Commerce.

3.2.3 Single-Value Decomposition (SVD)

Matrix Factorization is a sub module which factorizes the real or complex matrix thus, decomposing it. In literature, it is believed that representing ratings as vectors in feature space is quiet easy to work with. SVD generally slices the entire matrix into two halves[21]. Every user and item present in the matrix now gets associated internally with each other. Finding the internal product between the elements of the two matrices gives the prediction value that is desired. SVD is one of the popular dimensionality reduction techniques which use linear algebra[22]. It is commonly used to attain best rank approximation. SVD as a model based collaborative filtering technique. It characterizes the user and item's features and works effectively in performance as well. Matrix Factorization approximates the matrix employed by sloping down the number of attributes present in the matrix as shown in Figure 8. Single value decomposition is a powerful technique which decreases the sparsity in data and increases the performance. The Factorization of d rows by e column matrix is given by:

$$Factor A = U \sum V^T \tag{2}$$

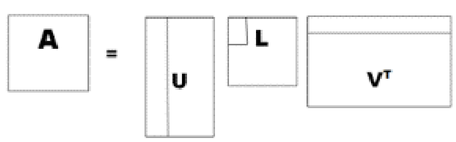

Figure 8. SVD process.

It eliminates the redundant data and the absolute error.

SVD uses the user profiles and items for recommendation process. It uses the stochastic gradient for effective elimination of error[23]. SVD produces stable ratings if and only if all the incoming ratings are more common to the predicted ratings. It looks at the detailed features of all the modules thus, directly profiling the user and item. The various applications of smoothing is Data normalization, Data visualization, Noise removal etc.., It uses the regularized error term for increasing the error term.

3.2.4 Bayesian Probabilistic Matrix Factorization

Probabilistic Matrix Factorization (PMF) is a method that eliminates the Gaussian noise present in the data used by working on the theory of probability[24]. The main focus of PMF is to estimate the suitable factorization which reduces the value of Root Mean Squared Error (RMSE) and thus, improving the performance of the system. Bayesian method works on observing the facts and predicting the result .It mainly works on hypothesis that is present when data becomes sparse[25]. Low rating matrix approximation is one of the quite effective filtering techniques. The main function employed in this approach would be steepest descent at times where linear factorization is in need. It commonly estimates the probability of the train set also. The linear factorization is a commonly used method for removing the Gaussian noise in many domains. Moreover, PMF is a powerful technique in filling up the missing values present in the matrix even the data is sparse.

3.3 Accuracy and Stability Evaluation

As mentioned earlier, the important factor to be considered while designing a system is its desired performance metrics. Accuracy and Stability are of these performance measures and are valid metrics to be noted. These evaluations are done using a sample of the dataset employed. The following are two most widely used measures of accuracy and stability:

3.3.1 RMSE

Root Mean Square Error (RMSE) is one of the commonly employed accuracy measure. It determines the error present in the system's outcome or predicted ratings. Mean Absolute Error (MAE) is another measure of accuracy. These both measures find the accurate difference between the original and the predicted ratings[26]. If x is the original rating afforded by the user (u) for item (a) and X_i is the predicted rating, then accuracy can be measured by the difference between $(x\text{-}x_i)$.In literature, accuracy is considered a the most important performance measure of a recommender system.

The RMSE for the original rating(x) by user (u) for item (i) and the predicted rating (x_i) is noted by:

$$RMSE = \sqrt{\frac{\sum_{(u,a)\in T}(x - x_i)^2]}{S}} \tag{3}$$

3.3.2 RMSS

Root Mean Squared Shift (RMSS) is one of the most widely used stability measure. RMSS actually finds the shift in prediction made by the system at different times. Mean Absolute shift is another most commonly used measure of the shift in predictions. When a recommender system recommends the same product for the same user at two different times with different rating values; RMSS is used to find the difference between the two predicted ratings[27]. The stability is calculated in a 2-phase approach that is being used in the prior literature. RMSS determines the difference between the predictions of the 2-phases in which the predictions are made depending upon the domain to which it is applied. The RMSS value of two predictions X_1 and X_2 for user u on item a is given by:

$$RMSS = \sqrt{\frac{\sum_{(u,a)}((x_1(u,a) - x_2(u,a))^2)}{S}} \tag{4}$$

4. Experimental Setup

Our nominated work is implemented with an objective to improve the stability. The movie lens dataset which contains 100k records with 100,000 ratings of 943 users is used for working on the desired process. The data consists of the user id, movie id and the corresponding rating given by the user. A sample of the dataset is given in the Table 2. The entire data is first separated into known and unknown set. Then, both data are sliced into two parts as training data (Dt,Et) and test data (Dv, Ev) as in the Figure 9.

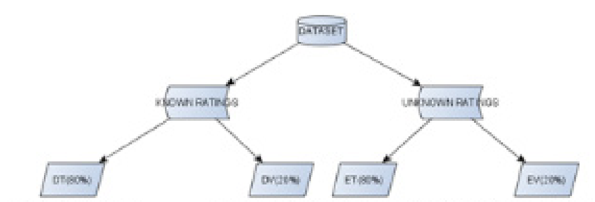

Figure 9. Data splitting.

Table 2. Illustration of proposed stable prediction process

Steps
1] Split the entire data into known set and unknown set.
2] Split the known into training data (Dt-80%) and test data (Dv-20%).
3] Split the unknown data into data to compute stability at training phase (Et-80%) and data to evaluate stability (Ev-20%).
4] Split known training data into 75 % and 25%.
5] Use the 75% data of the known ratings and give it as input to boosting, bagging or smoothing.
6] Select one recommendation algorithm from the list of used algorithms (User-based Collaborative filtering, Item based Collaborative filtering, Bayesian Probabilistic Matrix factorization and SVD).
7] Use the known training 25% data and evaluate the accuracy (RMSE).
8)Train the entire known training data set (80%) with our procedure and validate it with the known test data (20%).
9) Now take unknown train data (80%) apply the procedure and find the Phase-1 predictions.
10) Compute stability (RMSS) of the Phase-2 predictions by taking a sample of the phase-1 predictions as known ratings and validate it with the unknown test data (20%).

4.1 System Flow

The following Figure 10 depicts the overall flow of the entire proposed work that is intended to enhance stabilities. The known train data is trained with the task to be performed. To understand the working of the above declared ensemble methods the evaluations of those works are also carried out. The selection of ensemble technique and the desired prediction model is done by the admin. In some cases it can be made by the user's choice and the domain to which the work is applied. There are 4 algorithms that are used for prediction. All the 4 algorithms are proved to be efficient and flexible. After the selection of desired model the first phase predictions is made along with the estimation of accuracy of the outcome using RMSE. A sample of this outcome is given as a new incoming data and the final prediction is made along with the computation of the shift using RMSS. The entire steps or the orderly procedure is explained in Table 3. The proposed work is quite closer to the prior work but in different way.

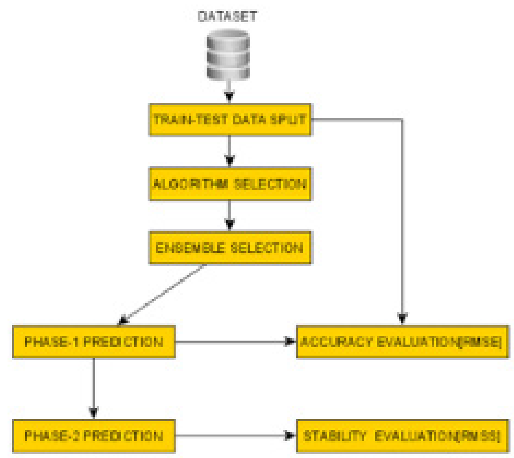

Figure 10. Stable recommendation system.

Table 3. Dataset sample

User-ID	Movie-ID	Rating
196	242	3
186	302	3
22	377	1
244	51	2
166	346	1
298	474	4
115	265	2

5. Experimental Results

The outcome of the proposed work is proved to be a personalized stable outcome and is depicted in the charts. The implementation aspect of the system and the output of both phases are depicted in the Figure 12 and 13. This implementation is carried out in JAVA built using Net-Beans platform. The final output of this work explores the view of the proposed methodology.

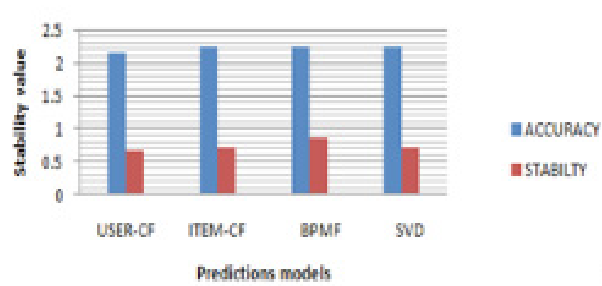

Figure 11. Bagging evaluation.

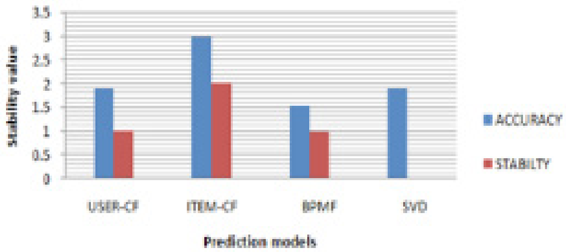

Figure 12. Boosting evaluation.

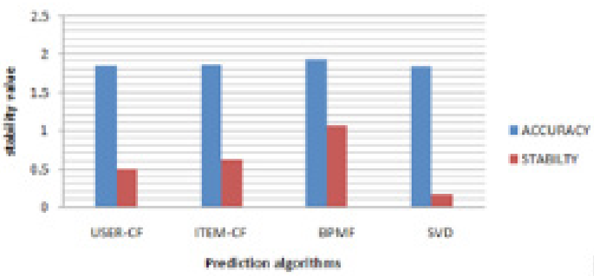

Figure 13. Iterative smoothing evaluation.

The proposed work of using boosting with conjunction to popular prediction models shows an equal or closely similar value to the outcome of the other two ensemble techniques that are employed in prior work. Boosting provides a personalized prediction which is more trustworthy. Following are the values of stability obtained and the Figure 11 to 13 shows the stability comparison of the employed. As mentioned earlier, the accuracy is calculated using the RMSE and the shift is calculated by the RMSS. The Table 4 to 6 shows the stability and the accuracy estimated by combining different ensemble methods and different prediction model at each case. It is marked from the outcome that the stability improvement of boosting is also considerable and is very close to the previous work.

Table 4. Bagging results

Algorithm	Accuracy	Stability
USER-CF	2.161	0.666
ITEM-CF	2.234	0.727
BPMF	2.238	0.88
SVD	2.232	0.71

Table 5. Boosting results

Algorithm	Accuracy	Stability
USER-CF	1.914	1.01
ITEM-CF	2.987	2.03
BPMF	1.55	0.991
SVD	1.914	0.024

Table 6. Iterative smoothing results

Algorithm	Accuracy	Stability
USER-CF	1.839	0.501
ITEM-CF	1.86	0.62
BPMF	1.925	1.05
SVD	1.838	0.16

6. Conclusion

This study is about improving the performance of the prediction system with a new metric called stability. The study of quality- of- service; a fundamental element deeply describes the interaction behaviour between the user and the services. The stability measures the consistency factor of the outcome provided by the recommendation platform. Stability helps the system to withhold the trust the user has on the platform thus providing profit to the system. The computation of stability is done in two phases. The first phase predicts the ratings and a sample of these ratings are used as new known rating during the second phase. On employing only the algorithms for recommendation the outcome is quite instable. Hence in literature, two ensemble methods that help in improving the stability when used in conjunction to the popular recommendation algorithms were proposed**.** Initially bagging is used on to the user-item ratings available which divide it into sub-samples .These samples are then employed with different prediction models involving weight as a factor. Another approach that involves various iterations called iterative smoothing is also employed in improving stability. Our work employs 1 more ensemble technique boosting which also dedicates to the effective improvement of personalized stability. The experimental result shows that the usage of boosting, also contributes to the enhancement of the stability. Our results provide the personalized stable predictions and are proved to enhance the accuracy also. Predicting the stability increase taking into account the important features of users in offline survey on users and online mode with various other recommendation algorithms is an important direction for future work. And also trying to work out with the stability factor, taking into account the biased users (Who rates based on their own feel) and unbiased users (Who rate purposely) paves way for further enhancement.

7. Acknowledgement

The authors are grateful to Science and Engineering

Research Board (SERB), Department of Science and Technology, New Delhi, for the financial support (No. YSS/2014/000718). Authors also thank SASTRA University, Thanjavur, for providing the infrastructural facilities to carry out this research work.

8. References

1. Gediminas Adomavicius, Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the Art and Possible Extensions, IEEE Transactions on Knowledge and Data Engineering. 2005 Jun; 17(6):734–49.
2. Yeonjeong Lee, Kyoung-Jae, Youngtae Kim. Recommender Systems using Ensemble Techniques, International Journal of Computer, Electrical, Automation, Control and Information Engineering. 2013; 7(10):1–4.
3. Iisinkayae FO, Folajimi YO, Ojokoh BO. Recommender Systems Principles, Methods and Evaluation, Egyptian Informatics Journal. 2015 Nov; 16(3):261–73.
4. Gediminas Adomacius, Jingjing Zhang. On the Stability of Recommender Algorithms. RecSys '10 Proceedings of the Fourth ACM Conference on Recommender Systems; 2010 Sep. p. 47–54.
5. Setting Goals and Choosing Metrics for Recommender Systems Evaluations. Date Accessed: 01/2011. Available at: https://www.researchgate.net/publication/268381252_Setting_Goals_and_Choosing_Metrics_for_Recommender_System_Evaluations.
6. Gediminas Adomacius, Jingjing Zhang. Maximizing Stability of Recommendation Algorithms: A Collective Inference Approach, 21st Workshop on Information Technologies and Systems (WITS); 2011. p. 151–56.
7. Gediminas Adomacius, Jingjing Zhang.Stability of Recommendation Algorithms, ACM Transactions on Information Systems. 2012 Nov; 30(4).
8. Gediminas Adomacius, Jingjing Zhang. Improving Stability of Recommender Systems: A Meta-Algorithmic Approach, IEEE Transactions on Knowledge and Data Engineering. 2015 Jun; 27:1573–87.
9. Bonhard P, Sasee MA. Knowing me, Knowing you-using Profiles and Social Networking to Improve Recommender Systems, BT Technology Journal. 2006 Jul; 24(3):84–98.
10. Vedanayaki M. A Study of Data Mining and Social Network Analysis, Indian Journal of Science and Technology. 2014 Nov; 7(S7):185–87.
11. Savina Andonova, Andre Elisseeff, Theodoros Evgeniou, Massimiliano Pontil. A Simple Algorithm to Learn Stable Machines. Proceedings of the 15th European Conference on Artificial Intelligence (ECAI); 2002. 1–5.
12. Abilash Reddy C, Subramaniyaswamy V. An Enhanced Travel Package Recommendation System Based on Location Dependent Social Data, Indian Journal of Science and Technology. 2015 Jul; 8(16):1–7.
13. Data Mining: Finding Similar Items and Users. Date Accessed: 16/01/2012. Available at: https://alexn.org/blog/2012/01/16/cosine-similarity-euclidean-distance.html.
14. Gediminas Adomavicius, Jingjing Zhang. Stability of Collaborative Filtering Recommendation Algorithms, RecSys '10 Proceedings of the Fourth ACM Conference on Recommender Systems; 2010. p. 47–54.
15. Ruslan Salakhutdinov, Andriy Mnih. Bayesian Probabilistic Matrix Factorization using Markov Chain Montecarlo. 25th International Conference on Machine Learning; 2008. p. 880–87.
16. Master Recommender Systems. Date Accessed: 2016. Available at: www.coursera.org/learn/recommender-systems.
17. Improving the Performance of Personalized Recommendation with Ontological user Interest Model. Date Accessed: 03/12/2011. Available at: http://ieeexplore.ieee.org/document/6128299/.
18. Ariel Bar, Lior Rokach, Bracha Shapiral. Improving Simple Collaborative Filtering Models using Ensemble Methods, Springer Berlin Heidelberg; 2013 May. p. 1–12.
19. Nitesh V Chawla. Data Mining for Imbalanced Datasets: An Overview. Springer US, 2005; p. 853–67.
20. Jia Zhang, Yaojin Lin , Menglei Lin, Jinghua Liu. An Effective Collaborative Filtering Algorithm Based on User Preference Clustering, Applied Intelligence. 2016 Sep; 46(2):230–40.
21. Weilo Jianweiyin, Yingli Zhaohuiwu. Efficient Web Service QOS Prediction using Local Neighbourhood Matrix Factorization, Engineering Applications of Artificial Intelligence. 2015 Feb; 38:14–23.
22. Location-Based Hierarchical Matrix Factorization for Web Service Recommendation. Date Accessed: 06/2014. Available at: https://www.researchgate.net/publication/288319123_Location-Based_Hierarchical_Matrix_Factorization_for_Web_Service_Recommendation.
23. Shangguang Wang, Ching-Hsien Hsu, Zhong Jun Liang, Qibo Sun, Fangchun Yang. Multi -User Web Service Selection based on Multi QOS Prediction. Springer Berlin Heidelberg; 2013 Apr. p. 176–83.
24. David Cieslak, Nitesh Chawla. Analyzing Pets on Imbalanced Datasets when Training and Testing Class Distributions Differ. Springer Berlin Heidelberg; 2008 May. 519–23.
25. Learning Based Data Mining Using Compressed Sensing. Date Accessed: 13/07/2014. Available at: http://ieeexplore.ieee.org/document/6946763/.
26. Gwangseop Gim, Hogyeong Jeong, Hyunjong Lee. Group-Aware Prediction with Exponential Smoothing for Collaborative Filtering. CAMRa '11 Proceedings of the 2nd Challenge on Context-Aware Movie Recommendation; 2011. p. 11–17.
27. Robert M Bell, Yehuda Koren. Scalable Collaborative Filtering with Jointly Derived Neighbourhood Interpolation Weights, ICDM '07 Proceedings of the 2007 Seventh IEEE International Conference on Data Mining; 2007. p. 43–52.
28. Lei Shi. Trading-off among Accuracy, Similarity, Diversity, and Long-Tail: A Graph-Based Recommendation Approach. RecSys '13 Proceedings of the 7th ACM Conference on Recommender Systems; 2013. p. 57–64.