International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075, Volume-8 Issue-7 May, 2019

Mitigating Cold Start Problem In A Personalized Recommender System

Selva Rani B, Ananda Kumar S

Abstract*: *A cold-start problem faced by a recommender system leads to serious causes and ruins the functionality of the entire system, sometimes responsible for losing new users also due to poor accuracy in recommendations. Recommendation becomes very rigid in case of a new recommender system where the product details exist but no user started viewing or rating the products yet. Similarly, when a new product is added the corresponding ratings are missing or when a new user enters the system, there is lack of knowledge about the preferences of the new user. This work concentrates on the aforementioned cold-start problems by designing a hybrid recommender engine for academic choices. Users' preferences diverge time to time and domain to domain. Academia is one such field in which students' feel more challenging to pick up their course after completing their school, which determines the future of a student. This may be due to either less perception about the available choices or more information overload in the internet. There is no single point of contact which helps the students to explore and suggest the enormous choices in education. Recommender system is a tool which suggests the users to find out the best products based on their tastes and needs. Another bigger challenge in this system is missing ratings. Existing user profiles represents the preferences alone and not the rating about the courses or institutes. This work proposes such a personalized recommender system which recommends opt courses for a student based on his expected score as well as preference. The proposed methodology was evaluated on real data set available from previous year engineering counselling conducted by Anna University.

Index Terms*: *Cold Start, Collaborative Filtering, Knowledge Base, Personalized Recommendations.

I. INTRODUCTION

Due to exponential growth of information in the internet, the usage of the computers also increases and almost every person depend on recommendation systems in their regular day to day activities to make a better decision [1]. Any Recommendation System (RS) offers suggestions on various items like movie, music, holiday plans, hotels, airline reservation, insurance, books, online courses and many more, either to an individual (personalized RS) or to a group of users (group-based RS). It is a tool which ranks or suggests items to its users which may be liked or needed by them [2], [3]. E-commerce websites like Flipkart, Uber, Amazon etc. much rely on recommender systems to obtain new customers and to retain current users. Hence, in such large electronic networks there exists a rivalry to identify the interests and preferences of users towards items based on the feedback as well as ratings provided by them. Education is one of the most significant domains in which there is an exigency need for optimized recommender systems precise to the geographical territories. The period by which a student finishes his/her schooling is very crucial and taking the next step towards his/her higher education defines their entire life. The students from remote villages are not conscious about massive amount of choices for their higher education after schooling with nominal fee structure and constraints such as geographical domain, financial status and many more. There exists no well-defined recommender system to carry such choices of possible and available courses to this vivid generation.

Almost all the recommender systems are constructed to suggest items to users by gathering feedbacks or ratings from experienced users. Feedbacks are collectedly either explicitly by prompting the users to rate the items or implicitly by observing the activities of users and their past history of purchases. The two variants of recommender systems are content-based and collaborative filtering. Content-based filtering stores the items liked or purchased formerly by the users and based on his/her interests suggest items or products, whereas collaborative filtering identifies users with similar tastes and suggest items liked or purchased by the other users. But, both the techniques are in need of huge volume of data for analysis before suggesting an item to a user, which are escalating abundant challenging disputes for new recommender systems which have no prior ratings or preferences given by users. These sort of cold-start problems lead to another variant of recommender systems known as knowledge-based recommender systems. With the deep knowledge about the domain, these recommender systems gather users' preferences explicitly for a better recommendation.

In this paper, we aimed at designing a personalized recommender system for exploring academic choices to the students those who are about to finish their schooling. A good recommender system should identify a best match between items and users interests. In this work, it has been proposed to read the preferences from the students and to recommend courses based on students' interest. To achieve this mission, a knowledge base about the institutes and courses offered by the institutes as well as the various streams followed by Tamilnadu state board has been constructed. Initially, the students are made aware about the various possibilities for higher studies based on their stream of education in schools. Around 135 groups of stream are offered by Tamilnadu state board for higher-secondary education. Based on subjects in each stream, the students are eligible for higher studies

Revised Manuscript Received on May 10, 2019

Selva Rani B, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India.

Ananda Kumar S, School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India.

in various courses and institutes. Fig. 1 shows some of the possible domains which are normally preferred by the students for their higher studies.

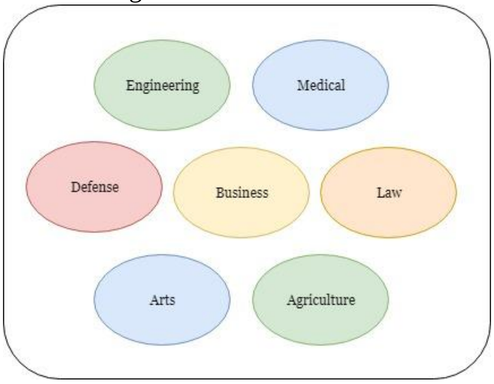

Fig. 1 Various domains for higher studies

II. RELATED WORKS

Recommender systems are intelligent tools designed to offer personalized services to the users. Some major problems faced in Recommender systems are quality, sparsity, scalability and first rater [1]. It has also been studied that the most commonly used recommendation systems are designed as collaborative filtering techniques. Collaborative filtering methods are classified into memory-based collaborative filtering and model-based collaborative filtering. A collaborative filtering recommender system measures the similarity between users, predicts an item/product and recommends to a target user which is termed as user-based collaborative filtering. In contrast item-based collaborative filtering recommends items/products by computing similarity between items. A content-based filtering system recommends items to a user based on his/her personal interests [2]. Knowledge based filtering is a recommendation technique which uses explicit knowledge about users, items and user preferences and recommends items to users by applying recommendation criteria [3]. It applies reasoning to recommend which items to which users in which context. It is also known as rule-based recommendation system which recommends products to users by referring to decision rules. Those kinds of systems depend on the creation of knowledge rules which propose items or products to the users which coincides the prescription of the rules [4]. Knowledge-based recommendation systems are useful in circumstances in which the two traditional recommendation (content-based and collaborative-filtering) approaches cannot be applied. Knowledge-based recommender systems accentuates on explicit knowledge about the domain as well as implicit knowledge about the users to mine appropriate recommendations. Normally, knowledge-based recommendation systems involve a set of constraints and a set of products. The constraints are used to describe the products to be suggested based on the current user desires [2]. The user of the system provides his/her preferences in the form of item specifications and which are designated as constraints internally. Knowledge-based recommender allows a malleable mapping between the users' preferences and product descriptions. It explicitly gathers users' preferences or descriptions about the products and recommends the items by consulting the knowledge base. Construction of a knowledge-base is not an easy task, since it has to be updated periodically so as to avoid incorrect recommendations [5]. Knowledge-based systems are further classified as constraint-based recommendation systems and case-based recommendation systems [3], [6]–[8]. Based on the similarity metrics, case-based recommendation systems computes recommendations [9]–[11]. But constraint-based systems relate the users' preferences or interests with the features of the items defined in knowledge bases [8], [12]. This work focused on constraint-based recommendation techniques to recommend courses to students based on predefined knowledge base, which was usually defined by the set of variables (v_c, v_p) and set of constraints (c_r, c_f, c_p) which are very much essential for solving the constraint satisfaction problems [13], [14]. The required solution will be obtained only if all the constraint variables are satisfied.

III. COLD-START PROBLEM

Conventional recommender systems such as collaborative filtering and content-based filtering deal with the similarity measures between the users/items or the users' personal interest and recommends products/items. But it would to be very tough to do so in a recommender system either for new users or new items since there won't be any browsing history, user's preferences, past purchase details etc. which is known as a cold start problem in recommender systems. The categories of cold-start problems are system cold-start, item/product cold-start and user cold-start. This work studied how to deal with system cold-start and user cold-start problems in a recommender system by deploying a hybrid recommender engine.

A. System cold-start

In case of a new system, knowledge-based recommender engine might be helpful to recommend products to the users by inferring preferences of the users. Knowledge-based recommender systems depend on the features of the items and the knowledge about how users' interests or preferences are met by these features. The required knowledge has been represented as a set of rules and while receiving users' interests or preferences and these rules defines which items have to be suggested. The target user specifies his/her preferences as the item features which are used to construct the rules in the knowledge base. Hence, the domain specific knowledge base for the recommender system should be populated with the sufficient number of features of items. The item features are mapped with users' preferences and depending on the similarity measure the recommendation task is initiated.

B. User cold-start

Popularity is the common strategy used to overcome the user-cold start problem by recommending popular items to the users and thereby cutting down the results with the help of user's contextual information. The most popular items/products are recognized by considering current trends, globally popular or a certain special occasions etc. By collecting auxiliary information about the user such as the device used to access, the geographical

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

location of the user and so on the recommender engine would be able to personalize the product/item to the user very soon. This work also agonized with user-cold start problem and hence it had been planned to recommend the courses based on popularity too.

IV. HYBRID RECOMMENDER

In order to attain enhanced accuracy in recommendation, a hybrid approach had been adopted. A hybrid recommender merges more than one recommendation techniques and produces recommendations. *Weighted, mixed* and *cascaded* are the three important approaches in hybrid systems. This work concentrates on *cascaded* technique in which the prediction of knowledge-based recommender is given as input to the collaborative filter for further refinement as given in Fig. 2.

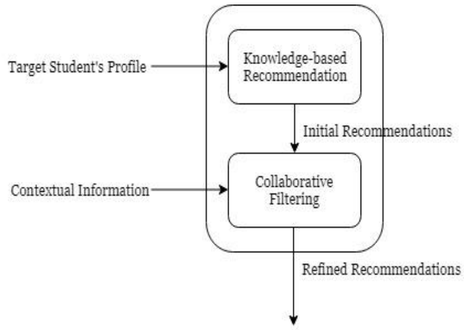

Fig. 2 A Hybrid Recommender for a personalized recommendation

A. Knowledge-based Recommendation

In a constrained-based recommender system a knowledge-base is generally constructed using set of variables (v_c, v_p) and set of constraints (c_r, c_f, c_p) [13]. Recommendations are computed according to the knowledge-base and the preferences of the users. *Recommendation task* identifies the items that meet the expectations of the users. Apart from arts, engineering and medical, the students are not much conscious about numerous openings of courses. This work aims at providing better recommendations by collecting the profile of the students explicitly. The profile includes mainly the stream at school, their expected scores in subjects and their domain-preference for higher studies. To achieve the objective, a recommender knowledge base was constructed with all the details about institutes in Tamilnadu, courses offered by the institutes and the eligibility criteria needed for the courses. A sample knowledge-base with details of students v_s, details of institutes v_i, required constraints c_r and filtering conditions c_f for the chosen domain *academic* is presented below:

Constraint-based Recommender Knowledge-base

$$v_s = \left\{ \begin{array}{c} name_s : [string] \\ dob_s : [date] \\ sos_s : [101, 102, 103 \ldots] \\ exp_{score} : [sub_1, sub_2, sub_3, sub_4] \\ com_s : [general, oc, obc, bc, bcm, sc/st] \\ pref_s : [arch, engg, medicine, agri \ldots] \\ address_s : [city, district] \end{array} \right\}$$

$$v_i = \left\{ \begin{array}{c} inst_{id} : [integer] \\ inst_{details} : [name, city, district, rank] \\ course_{available} : [course_1, course_2, \ldots course_n] \end{array} \right\}$$

$$c_r = \left\{ \begin{array}{c} cr_1 : sos_s = 101 \wedge com_s = general \wedge \\ pref_s = engg \rightarrow avg(sub_1, sub_2, sub_4) \geq 60\% \end{array} \right\}$$

$$c_f = \left\{ \begin{array}{c} cf_1 : sos_s = 101 \rightarrow \\ eligibility(course_1) = yes \\ cf_2 : sos_s = 102 \rightarrow \\ eligibility(course_1) \wedge \\ eligibility(course_2) = yes \end{array} \right\}$$

Now the recommender engine initially recommends the student about the courses to which they are eligible along with the list of institutions based on the stream of his/her study at school as in Fig. 3.

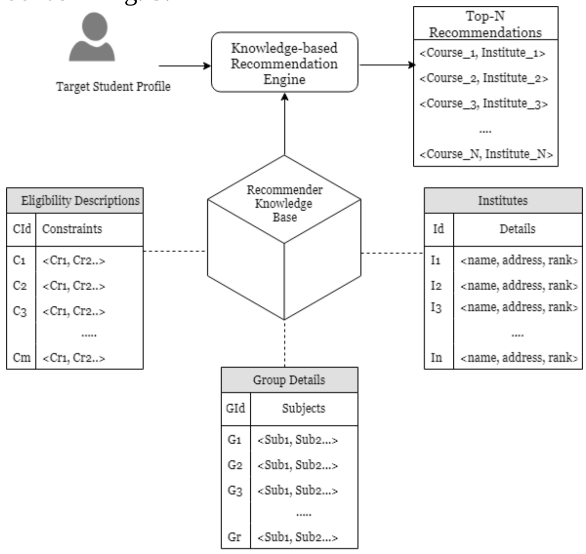

Fig. 3 Knowledge-based recommendation – A solution for system cold-start

These initial recommendations are suggested to the target student and given as input along with other contextual information to the collaborative filter to refine the recommendation.

B. Collaborative Filtering

Collaborative filtering (CF) is a way of making recommendations based on the similar interests between the users. This technique operates in *user-based* or *item-based* mode. *User-based collaborative filtering* identifies the group of users termed as nearest neighbors with similar tastes to the target user. Based on the ratings given by the nearest neighbors, with a help of a statistical measure a prediction is made whether to recommend an item to the target user or not. Pearson correlation coefficient is one such measure used to compute the nearest neighbors. If u_1 and u_2 are two users, R is a matrix which represents the ratings to the items given by the users and r is the rating given by the user for a particular item. The similarity between two users is computed by using Pearson correlation (PC) coefficient as in eq. (1)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

$$sim(u_1,u_2) = \frac{\sum_{i \in I}(r_{u_1,i} - \overline{u_1})(r_{u_2,i} - \overline{u_2})}{\sqrt{\sum_{i \in I}(r_{u_1,i} - \overline{u_1})^2}\sqrt{\sum_{i \in I}(r_{u_2,i} - \overline{u_2})^2}} \quad (1)$$

After finding nearest neighbors the prediction is made to guess the rating for a new item by the target user using the eq. (2).

$$pred\,(u_1,i) = \overline{r_{u_1}} + \frac{\sum_{u_2 \in N} sim(u_1,u_2) * (r_{u_2,i} - \overline{r_{u_2}})}{\sum_{u_2 \in N} sim(u_1,u_2)} \quad (2)$$

Based on the predicted value, the new item would be recommended or not be recommended. *Item-based collaborative filtering* computes cosine similarity as a classic metric which is proved to be more accurate. The item features are represented as vectors and the angle between them measures the similarity value. The cosine similarity (CS) between two vectors *i* and *j* is found using eq. (3) which may result values between 0 and 1.

$$sim(i,j) = \frac{\vec{i}.\vec{j}}{|\vec{i}| * |\vec{j}|} \quad (3)$$

The similarity measure nearer to 1 represents stronger similarity between two items. Any two feature vectors *x* and *y* may also be compared with the help of Euclidean distance (ED) as in eq. (4)

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (4)$$

Here, in user-based collaborative filtering instead of rating to the items, the preferences for the courses given by students are considered. For instance, by stating his/her expected cutoff in the forthcoming examination, a target student may wish to know how many students had chosen B.E. Electronics and Communication Engineering in the previous year or what all the branches are obtained by previous year students along with the college code at which they got admission as illustrated in Fig. 4. Similarly, in item-based collaborative filtering the target student may mention his/her community and provide preferences such as branch he/she wants to study and the college in which he/she wants to join, by requesting the required cutoff to obtain admission in the particular course as in Fig. 5.

V. RESULTS AND DISCUSSION

This proposed system has been implemented in Python 3. A target student is expected to enter his initial profile with details such as *name, mobile_no, e_mail, district, stream_of_study* and *expected_scores* in main subjects in the stream they studied in school. The courses along with institute names are given as initial recommendations to the students with the help of knowledge-based recommender. Once the courses are listed based on constraints, the recommendation has been refined by using collaborative filtering. The target student is expected to interact with the system by choosing a particular course For e.g. *Engineering*. Then by adding other contextual information such as the *college_code, community, branch preference* etc. collaborative filtering predicts the details of courses as per the target student's expectations.

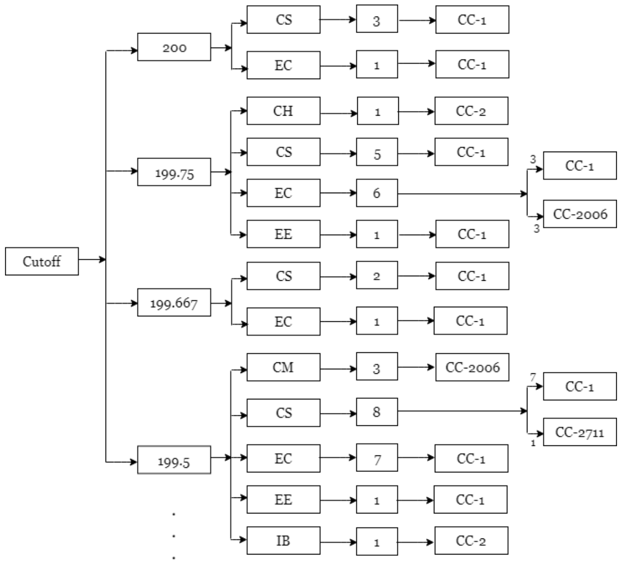

Fig. 4 Particulars of number of students chosen specific branches at preferred institutes

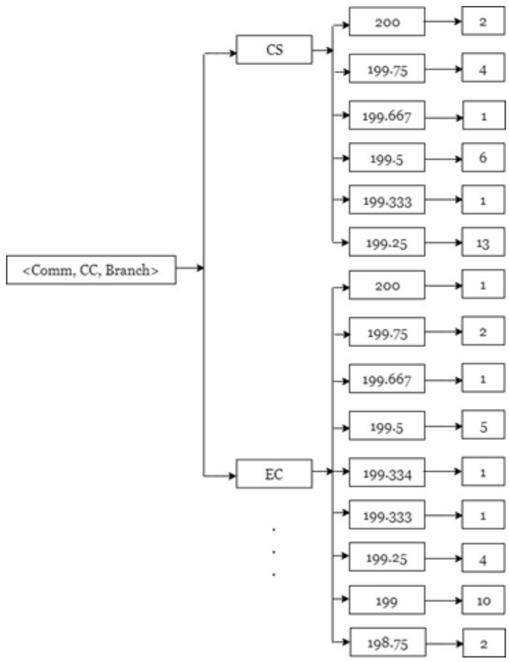

Fig. 5 Particulars of cutoff required to obtain a preferred branch at preferred institute

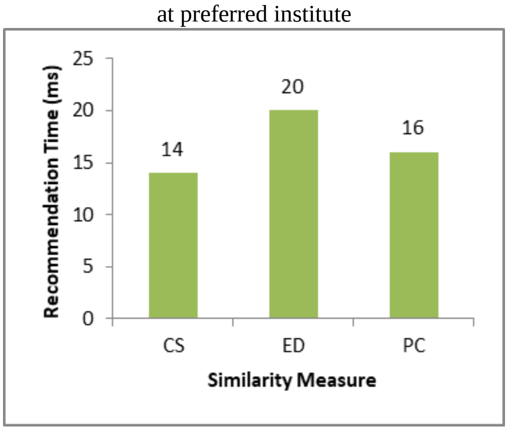

Fig. 6 Similarity measures vs. Recommendation time

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

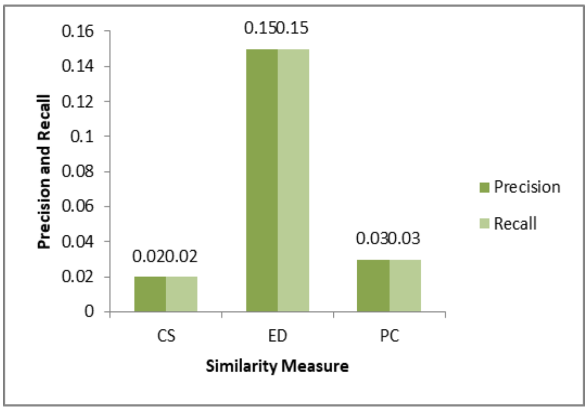

Fig. 7 Similarity measures vs. Precision and Recall

VI. CONCLUSION

This work aims at predicting the academic options to the students and recommending the possibilities too. This work initially aimed in minimizing the cold start problem and sparsity problems that normally is very challenging in collaborative filtering recommendation systems. The method demonstrated in this work has eliminated the cold start problem. The performance of the proposed method was also evaluated using the standard metrics similar to other recommendation systems. Three similarity measures were used and the performance of the proposed method was assessed. As an extension to this method, feedback can be collected from the students about their institutions and the courses so that the performance of this methodology would still be improved.

REFERENCES

1. E. Vozalis and K. G. Margaritis, “Analysis of recommender systems algorithms,” in The 6th Hellenic European Conference on Computer Mathematics & its Applications, 2003, pp. 732–745.
2. A. Felfernig, M. Jeran, G. Ninaus, F. Reinfrank, S. Reiterer, and M. Stettinger, “Basic approaches in recommendation systems,” in Recommendation Systems in Software Engineering, Springer, 2014, pp. 15–37.
3. S. Trewin, “Knowledge-based recommender systems,” Encycl. Libr. Inf. Sci., vol. 69, no. Supplement 32, p. 180, 2000.
4. B. Amini, R. Ibrahim, and M. S. Othman, “Discovering the impact of knowledge in recommender systems: A comparative study,” arXiv Prepr. arXiv1109.0166, 2011.
5. M. Zanker, “A collaborative constraint-based meta-level recommender,” in Proceedings of the 2008 ACM conference on Recommender systems, 2008, pp. 139–146.
6. N. Mirzadeh, F. Ricci, and M. Bansal, “Feature selection methods for conversational recommender systems,” in e-Technology, e-Commerce and e-Service, 2005. EEE’05. Proceedings. The 2005 IEEE International Conference on, 2005, pp. 772–777.
7. B. Smyth, L. McGinty, J. Reilly, and K. McCarthy, “Compound critiques for conversational recommender systems,” in Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, 2004, pp. 145–151.
8. A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, “Constraint-based recommender systems,” in Recommender Systems Handbook, Springer, 2015, pp. 161–190.
9. D. Bridge, M. H. Göker, L. McGinty, and B. Smyth, “Case-based recommender systems,” Knowl. Eng. Rev., vol. 20, no. 3, pp. 315–320, 2005.
10. R. Burke, “Knowledge-based recommender systems,” Encycl. Libr. Inf. Syst., vol. 69, no. Supplement 32, pp. 175–186, 2000.
11. F. Lorenzi and F. Ricci, “Case-based recommender systems: A unifying view,” in IJCAI Workshop on Intelligent Techniques for Web Personalization, 2003, pp. 89–113.
12. A. Felfernig, G. Friedrich, D. Jannach, and M. Zanker, “An integrated environment for the development of knowledge-based recommender applications,” Int. J. Electron. Commer., vol. 11, no. 2, pp. 11–34, 2006.
13. A. Felfernig, K. Isak, and T. Kruggel, “Testing knowledge-based recommender applications,” OEGAI Journal, Spec. Issue Recomm. Syst., vol. 24, no. 4, pp. 12–18, 2005.
14. E. Tsang, Foundations of constraint satisfaction: the classic text. BoD--Books on Demand, 2014.

AUTHORS PROFILE

Mrs. Selva Rani B is presently associated with Department of Information Technology as an Assistant Professor at Vellore Institute of Technology in Vellore, India. Currently she is working in Recommender Systems and Multi Criteria Decision Making. She is also interested in Network Security.

Dr. Ananda Kumar S is an Associate Professor of School of Computer Science and Engineering at Vellore Institute of Technology in Vellore, India. His research interests include IoT, Security, Recommender Systems and Sensor Networks. He has published good number of papers in International conferences and Journals of repute.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication