



Hybrid conversational recommender system using temporal difference

Poonam Thorat¹, Prof. R M Goudar², Prof. Sunita Barve³

Department of Computer Engineering

MIT Academy of Engineering, Alandi

University of Pune, Pune, India

Abstract: Recommender systems or recommendation systems are a subset of information filtering system that used to anticipate the 'evaluation' or 'preference' that user would feed to an item. In recent years E-commerce applications are widely using Recommender system. Generally the most popular E-commerce sites are probably music, news, books, research articles, and products. A conversational recommender system uses critiquing as a feedback to efficiently retrieve a product. Critiquing is common and powerful form of feedback, where a user can express their feature preferences. The expectation is that in each cycle the system retrieves the product that best satisfy the user's preferences from a minimal information input. The Reinforcement Learning (RL) approaches will be used to improve retrieval quality based on combination of compatibility and similarity scores. Reinforcement Learning (RL) reward function approach will be used with existing critiquing strategies to provide compatibility that will take into account for both, the moment at which the user makes a review and the number of satisfied review. This work focuses on new and existing strategies for conversational recommender system that will have potential to improve retrieval quality.

Keywords: Reinforcement learning, temporal difference, critique, conversational recommender system.

I. Introduction

Recommender systems have become extremely common in recent years and are applied in a variety of applications. Recommender Systems (RSs) are software tools that are used to provide suggestions to user according to their requirement. The suggestions associate with various decision-making processes, such as which items to buy, what music to listen to. "Item" is the general term used to denote what the system recommends to users. A RS normally focuses on a specific type of item, its design, its graphical user interface and the core recommendation technique used to generate the recommendations are all customized to provide useful and effective suggestions for that specific type of item. Due to the increasing importance of recommendation, it has become an autonomous research field since the mid 1990s [1].

Broadly speaking, a RS suggests to a user those items that might be of user's interest. Recommender systems are designed to assist users to navigate through complex information spaces. They combine ideas from information retrieval, user profiling and machine learning, among others, to deliver the user a more efficient search environment, one that is better adapted to their needs and preferences, and one that, ultimately, helps them to locate what they are looking for more quickly and more easily. Conversational recommender systems engage the user in an extended dialog, making suggestions based on the user's initial query and refining these suggestions based on their feedback. The type of feedback used depends on the recommender system. Sometimes the user is asked to provide very specific information.

Critiquing-based recommender systems generally adopt a fixed style of interaction with the user; during each cycle a case and a standard set of critiques are presented [2, 4]. This work combine content based filtering with collaborative filtering to form new hybrid recommender system. This hybrid recommender system will overcome the imitation of existing system. Also this work describe a more dynamic strategy that uses Reinforcement learning methods to automatically generate compound critiques, during each recommendation cycle [3, 6]. In this way the user is presented with a set of feedback options that are tuned to their particular recommendation session, the past choices that they have made, and the products that are remaining for consideration. Furthermore, we show that this approach can have a significant impact on recommender performance.

II. Related work

The following are the main filtering technique are used while designing the recommender system;

A. collaborative-filtering

Collaborative Filtering (CF) methods play an important role in the recommendation process and because of that Collaborative filtering is most extensively used approach to design recommender system [1,2]. In this approach recommendation for each active user is received by comparing with the preferences of other users who have rated the product in similar way to the active user.

B. Content-Based filtering

In content-Based filtering recommendations depends on users former choices. Item description and a profile of the user's orientation play an important role in Content-based filtering. Content-based filtering algorithms try to recommend items based on similarity count.

C. Demographic filtering

In demographic filtering recommendations is based on a demographic profile of the user. Here recommendation is based on the information provided by the user is considered to be similar according to demographic parameter such as nationality, age, gender etc.

The existing system work on either one of the above filtering technique so the existing system suffer from data sparsity , scalability, cold start problem. The existing system does not consider user feedback(critique) while recommending product .

D. Limitation of exiting system

1. Cold Start: CF systems often require a huge amount of existing data on which user can make exact recommendations [18].
2. Scalability: CF makes recommendations for various environments where billions of users and products exist. Therefore, a huge amount of computation power is often essential to compute recommendations.
3. Sparsity: On major e-commerce site the number of items sold are enormously large. Because of that only a small subset of the entire database is rated by most active users. Hence very few ratings are given to the most popular items[1].

III. Proposed work

The proposed work focus on hybrid conversational recommender system which will overcome the drawbacks of collaborative filtering and content based filtering. The reinforcement learning approach will be used to improve the prediction accuracy so that the preference list provided to the user will best satisfy their requirement.

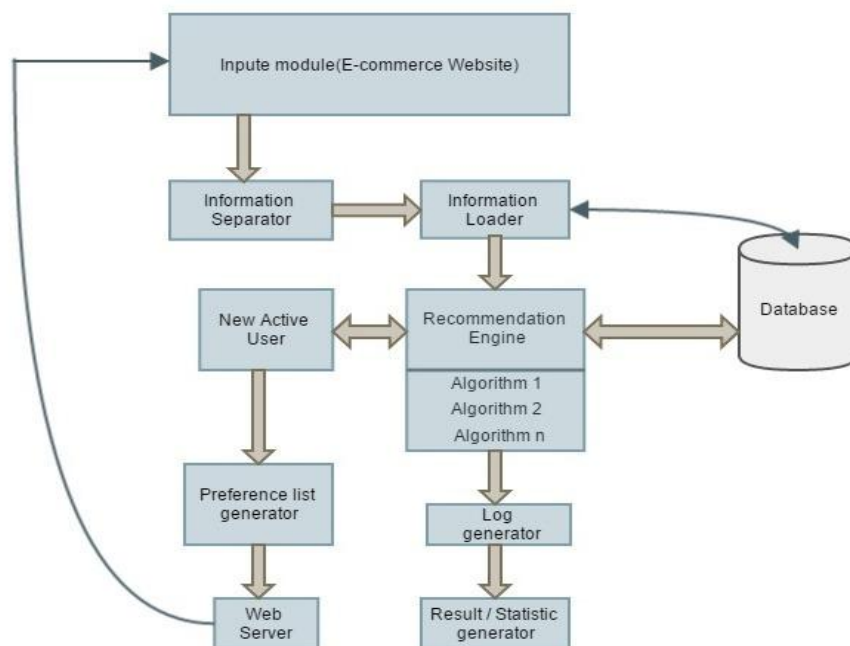


Figure.1 system architecture

The above Figure.1 shows the system architecture of the conversational recommendation system. System consists of following main modules.

A. Input Module / Ecommerce Website:

This is the input for the system: Here we are going to facilitate the user to map their ecommerce website contents. we are going to provide JDBC connector which will extract the information from their database tables. Usually Information contents, Product Database, Feedbacks provided by their users, User detailed information, ongoing Offers and Discounts .

B. Information Separator:

This module is responsible to get data from website database. Here JDBC connector will connect with database and extract the data (Read the data) and also it know the required table format for our system so it will transform the data accordingly to make it simplified. While inserting it also gives some statistics to Logger module (Like No of Records fetched, Time taken to convert the records).

C. Active Users Information:

This module will collect the user information who all are in active state, what they are searching for and what criteria they have currently selected. This all information will be the input of our system.

D. Information Loader:

This module takes the input from Information extractor and do the insert operation in respected tables. While inserting it also gives some statistics to Logger module (Like No of Records inserted, Time taken to insert the record).

E. Web server:

Web server will be the important component of this system as this project is developing web based application this server will responsible to manage all the Request/ResponDs and also provide session time limits to users. Typically apache tomcat will be the host who is responsible to do all these functionality.

F. Recommendation Engine:

This is the core module of our system which takes users previous critiquing , user current state , users selected criteria and product as a input , Another set of input it will take is users feedback. With the help of algorithm it will generate recommended product Ids as a output.

G. Preference List /Dynamic List generator:

This module will be responsible to generate product list based on recommendation generated by Recommendation engine.

H. Log Generator:

It will collect the data from various module and write the logs in to the text file or .html file.

I. Computational result Generator:

It will receive the data from Recommendation engine like algorithm selected ,Number of attributes ,Search time and plot the graph against it.

IV. Methodology

Methodology Using RL the agent learns how to achieve the given goal by trial and error interacting with environment. That improves performance of the agent from the received reward or punishment as a result of the performed action [15,16]. Temporal difference (TD) method learns directly from environment dynamics [17] without any knowledge of its model. It uses generalized policy iteration to update estimates i.e. value score, based in the parts on other learned estimates, without waiting for the final outcome. TD method wait only until the next time step, at $t+1$ it immediately form a target and make a useful update using observed reward r_{t+1} and the estimate $V(s_{t+1})$, So estimated value score $V(s_t)$ is updated by

$$V(s_t) \leftarrow V(s_t) + \alpha_{BRN} [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (1)$$

Where α_{BRN} , is a step-size parameter that influenced by BRN i.e. Best relay node among willing neighbor.

Markov decision process the task that satisfies the markov Property, i.e. all decisions and values are functions of the current state only, is called Markov Decision Process (MDP) [24]. MDP is defined by its state and action sets and by the one-step dynamics of the environment. Given the current state and action, one-step dynamics enable us to predict the next state, expected next reward and iteratively all future states and rewards.

A. Algorithm

1. Initial Recommendation

define **ItemRecommend**(q , CB, cq, U)

In this step the function is written to find out similarity using content based recommendation.

Here new case or product recommended to the user based on the current query q and previous critiques.

2. Create user profile

$CB' \leftarrow \{ p \in CB \mid \text{Satisfies}(p, cq) \}$

Let $CB = \{ \rho_1, \dots, \rho_n \}$ defined as set of products or cases for recommendation.

Where,

$CB = \text{Total number of product}$

$P_i = \text{ith product}$

In this step user reviews recommendation and then give feature critique, cq.

3. Calculation of compatibility score

$CB' \leftarrow \text{sort cases in } CB' \text{ in decreasing compatibility score}$

$CB'' \leftarrow \text{select from those cases } CB' \text{ with highest compatibility}$

$CB'' \leftarrow \text{sort cases in } CB' \text{ in decreasing order.}$

In this step compatibility score is calculated by reviewing query in each next cycle.

4. Calculation of similarity score

$pr \leftarrow \text{most similar case in } CB''$

In this step case based recommender system use a similarity function(based on nearest neighbour rules) to recommend the most similar product in each cycle.

5. Combining compatibility score and similarity score

In this step compatibility score and similarity score of initial case to the current recommendation are combined to obtain overall quality.

6. Updation of user model

The user model $U=\{U_1,...,U_k\}$ is updated by adding the last critique and deleting all the critiques that are inconsistent with it.

7. Termination of recommendation process

Finally, recommendation process terminates either when the user retrieves a suitable product or when user explicitly finishes the process.

• **Highest compatibility**

In this algorithm for calculation of highest compatibility the existing strategy is used with the Reinforcement Learning .Reinforcement Learning based on Markov Decision Process (MDP) is used.MDP consist of state ,action ,reward.

Here,

State : Each case or product.

Action : critique. Critique is one of the common form of feedback. Their are two types of critique:

1.Unit critique

2.Compound Critique.

Reward : satisfaction of current critique / user preference.

V. Future work and conclusion

Recommender systems are turning out to be a useful tool that will provide suggestion to user according to their requirement. Filtering is used to improved recommendation accuracy but each filtering technique associated with some disadvantages with it to overcome this drawbacks here we proposed hybrid conversational recommender system using temporal difference with the help of user feedback (critique) and reinforcement learning we will improve the accuracy of recommendation process. To improve the quality of recommender systems anticipations future research will concentrate on progressing the existing methods and algorithms. Novel lines of research will be formulated for following fields, such as on: (1) The existing recommendation methods that uses different types of available information will be combine in good order, (2) For recommender systems processes enable security and privacy, (3) Flexible frameworks are design for machine-controlled analysis of heterogeneous data.

References

- [1] J. Bobadilla, F. Serradilla, "The effect of sparsity on collaborative filtering metrics", in: *Australian Database Conference*, 2009, pp. 9–17.
- [2] R. Burke, Interactive Critiquing for Catalog Navigation in E-Commerce, *Artificial Intelligence Review*, 18(3-4):245-267, 2002.
- [3] K. McCarthy, J. Reilly, L. McGinty, and B. Smyth, On the Dynamic Generation of Compound Critiques in Conversational Recommender Systems ,In P. De Bra, editor, *Proceedings of the Third International Conference on Adaptive Hypermedia and Web-Based Systems (AH-04)*. Springer, 2004. Eindhoven, The Netherlands.
- [4] R. Burke, K. Hammond, and B.C. Young, The FindMe Approach to Assisted Browsing, *Journal of IEEE Expert*, 12(4):32-40, 1997.
- [5] J.B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), *The Adaptive Web*, 2007, pp. 291-324.
- [6] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth, Dynamic Critiquing, In P.A. Gonzalez Calero and P. Funk, editors, *Proceedings of the European Conference on Case-Based Reasoning (ECCBR-04)*. Springer, 2004. Madrid, Spain.
- [7] M. Pazzani, A framework for collaborative, content-based, and demographic filtering, *Artificial Intelligence Review-Special Issue on Data Mining on the Internet* 13 (5-6) (1999) 393-408.
- [8] X. Luo, Y. Xia, Q. Zhu, Applying the learning rate adaptation to the matrix factorization based collaborative filtering , *Knowledge Based Systems* 37 (2013) 154-164.
- [9] M. Balabanovic, Y. Shoham, Content-based, collaborative recommendation, *Communications of the ACM* 40 (3) (1997) 66-72.
- [10] Lops, Pasquale, Marco De Gemmis, and Giovanni Semeraro, Content- based recommender systems: State of the art and trends, *Recommender systems handbook*. Springer US, 2011. 73-105.
- [11] Sutton, R.S., Barto, A.G., 1998, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA.
- [12] Salam, Maria, et al, Improving incremental critiquing, *16th Artificial Intelligence and Cognitive Science* (2005): 379-388.
- [13] R. Burke, Hybrid recommender systems: survey and experiments, *User Modeling and User-Adapted Interaction* 12 (4) (2002) 331-370.
- [14] Doya, Kenji, Reinforcement Learning: Computational Theory and Biological Mechanisms, *HFSP Journal* 1.1 (2007): 30-40. PMC. Web. 19 Feb. 2015.
- [15] Duana Yet al (2007) Application of reinforcement learning in robot soccer. *Eng Appl Artif Intell* 20:936–950
- [16] Nie J, Haykin S (1999) A dynamic channel assignment policy through Q-learning. *IEEE Trans Neural Netw* 10(6):1443–1455
- [17] Kaelbling LP et al (1996) Reinforcement learning: a survey. *J Artif Intell Res* 4:237–285
- [18] K. Heung-Nam, E.S. Abdulmotaleb, J. Geun-Sik, "Collaborative error-reflected models for cold-start recommender systems", *Decision Support Systems* 51 (3) (2011) 519–531.