Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 45 (2024), No. 4, pp. 935–945
DOI : 10.47974/JIOS-1617

MRBUP : MapReduce based scalable user profiling in recommender system

Anjali Gautam *
School of Computer Applications
Manav Rachna International Institute of Research & Studies
Faridabad
Haryana
India

Veenu Bhasin †
Department of Computer Science
P.G.D.A.V College
University of Delhi
New Delhi
Delhi
India

Abstract

The speed at which data is generated has accelerated over the years and is predicted to grow exponentially in the future. It is preferable to pique the user's interest by building a user profile, then filter the products based on their preferences. User profiling becomes necessary as it enables the system to understand the user's tastes and likes from voluminous data. To build a scalable system, this paper proposes MRBUP which focuses on creating scalable user profile by using the model of MapReduce. Users in a recommender system interacts with products, which are humungous in nature. To build user profile, MRBUP practices term frequency-inverse document frequency technique. For experimental analysis, this work uses the Amazon Product Reviews dataset which defines product reviews given by users to different products in the catalog. MRBUP utilizes the product reviews given by the user to the products which are in pure textual form to generate user profile. MRBUP is executed on a Hadoop cluster of varying size and computational time is observed. Scalability MRBUP is evaluated using the usual metric of Speedup factor.

Subject Classification: *68T99.*

Keywords: *User profile, MapReduce, Information retrieval, TF-IDF.*

* *E-mail:* **anjali.gautam09@gmail.com (Corresponding Author)**
† *E-mail:* **veenu.bhasin@pgdav.du.ac.in**

© Taru Publications

1. Introduction

Users have been exposed to a vast amount of data since the Internet's inception. Users can now navigate through this enormous pool of data with ease due to tailored solutions, which are in high demand courtesy, data explosion. Identifying the user's interests is the biggest problem these tailored systems have to deal with. The process then entails identifying the user, removing any unnecessary information, and displaying the information that interests the user in a Recommender System (RS).

This work presents the first step in building any recommender system which is identifying the user's interest from humungous data. A user profile captures the users's interest in the form of feedbacks (both explicit and implicit) given by the user to the products. It is detailed information about his/her likes and tastes. For our proposed approach, MRBUP, we have exploited the product reviews given by the user to build a user profile. Various techniques of building user profiles exists in literature based on the application area [1]. In MRBUP, Term Frequency-Inverse Document Frequency (TF-IDF) is used to generate user profile. Scalability of the system is ensured by implementing TF-IDF using MapReduce programming on Apache Hadoop [2] [3] [4].

The entire work is structured into different sections. Section 2 presents the literature which led to the emergence of the proposed scalable technique. A detailed methodology of the proposed approach, MRBUP is presented in section 3. Section 4 presents the experimental analysis of MRBUP. It gives an insight of the dataset used to evaluate the proposed approach, the results and the analysis of the same. Section 5 concludes the paper and presents the direction for future work.

2. Related Work

The concept of user profiling becomes critical in a recommendation scenario data is in enormous volume. This section presents the contemporary state of literature relevant to our work.

A user profile is an essential part of a recommender system that records a specific user's preferences, actions, and traits. In essence, it functions as a digital representation of the user's demographic data, historical interactions, and interests. User-provided explicit feedback (ratings, reviews) [5], implicit feedback (click-through rates, past purchases), and demographic information (age, gender, location) are some of the methods used to construct user profiles. The recommender system

can comprehend the user's preferences and make precise predictions about their future preferences by examining these components [6]. The significance of user profile cannot be overstated. With the creation of user profile, the system is able to sift through a voluminous data and suggest items that are most likely to strike a chord with the user, increasing user happiness and engagement. Second, user profiles allow ongoing adaptation and learning. Users' profiles change as they engage with the system and offer input, guaranteeing that the suggestions are current and pertinent. In the end, user profiles are the foundation of efficient recommendation systems, enabling a more customized and interesting user experience [7].

[8] proposed a travel recommender system where user profiling was done by interpreting the activities carried out by the user or by incorporating the preference of the user. This helped in identifying diverse geographical interest of users. To build a user's profile, history of the user was captured and collaborative data was exploited. Bocanegra et al. [9] in their work has emphasized the importance of user profile creation and its usefulness. Reviews were regarded as a primary source for creating user profile [10]. The significance of understanding a user in terms of his likes and tastes helped the system generate quality recommendations. Liang [11] defined user profiling as the process of learning decision making for the users in a heterogeneous information networks. Explicit rating acted as a catalyst in improving the user profile. The importance of user profiling cannot be ignored in the cross-domain recommender systems as well [12] [13] [14]. Transfer learning via user profiling acted as a starting point for all the cold start users in the system thereby increasing the reliability of the system.

Literature has suggested various techniques to create user profiles when generating recommendations. Lu Zhongqi et al. [15] proposed a collaborative evolution model for creating user profile which focused on collaborative learning and evolutionary learning of user profiles. Hierarchical user profiling is proposed by Gu et al. [16] & Peterson [6] where real time interests of the user was identified by mapping the interests to different levels of granularity. Another interesting technique of frequent sub graph mining was adopted to generate semantic user profiling for better recommendations [17]. The researcher in their work trained a deep neural network to understand the association of users and items in order to create user profiles.

In order to generate user profiles, this study utilizes the idea of Term Frequency –Inverse Document Frequency (TF-IDF). For every user, the

user profile will be a set of terms which will include the terms (captured from user reviews) filtered on the tf-idf score depicting the importance of that term for a particular user. To ensure scalability of the system, we have used Hadoop MapReduce to implement TF-IDF.

3. MRBUP: Proposed Methodology

This work proposes MRBUP, a scalable technique of user profiling based on Hadoop MapReduce programming paradigm. We have consider amazon product reviews as our dataset. The reviews of products for every user is collected. These reviews are then preprocessed. To generate a user profile, the preprocessed reviews are used to generate a set of terms for each user using the principle of TF-IDF. Term vectors using TF-IDF is implemented in Hadoop MapReduce.

We take into account every user review while building their user profile. The terms that appear in the reviews are words. Every review is represented as a word vector. The term-document class weighs the terms in TF-IDF, a mathematical procedure, by utilizing MapReduce. The system creates a word vector by applying TF-IDF to determine the relevance of each word in a review that the user has provided. Let R = set of user review given by user

$$R = \{r_1, r_2, \ldots\ldots r_i\} \tag{1}$$

Every review $r_i \in R$ will be represented as:

$$r_i = [t_0, w_{i_0}; t_1, w_{i_1}; \ldots\ldots; t_t, w_{i_t}] \tag{2}$$

where t_t, w_{i_t} corresponds to term $t$and its associated weight w_{i_t} for a every review r_i.

Step 1: Preprocessing the reviews

The reviews provided by the users are not ready to use as an input to generate user profile. It is necessary to preprocess this purely textual data by using standard techniques of text preprocessing. In this work, tokenization, stop word filtering, punctuation removal, stemming and lemmatization is used. Tokenization involves the process of splitting text into smaller units such as words, symbols or other meaningful elements which are known as tokens. By eliminating commonly occurring stop words such as an, the, a, and on, stop word filtering increases the frequency of words. The practice of minimizing word variations that often have semantic explanations comparable a stem is known as stemming. For example, let us consider that the words “run,” “running,” and “ran” are

present in different user reviews. When stemming is applied to these words, the root word, run, will be stemmed. MRBUP uses Porter Stemmer, which provides a list of stem terms for every review. When processing reviews, it becomes important to understand and represent the meaning of the words present in the reviews which enhances the accuracy. For this, the proposed work implements the process of lemmatization which reduces the words occurring in the reviews into its base or dictionary form which is known as lemma. This process takes into account parts of speech and its context to deliver the resulting lemma a valid word. WordNetLemmatizer is used in this work.

After preprocessing the reviews, each review r_i will now be a list of stemmed words as:

$$r_i = \{t_1, t_2, \ldots t_i\} \tag{3}$$

In order to generate user profile using TF-IDF implemented on MapReduce it is vital to convert the data into a <key,value> pair. In this case, the <key, value> pair corresponds to <review_id$_i$, r_i>. The Hadoop Distributed File System (HDFS) is loaded with a sequence file in the <key,value> pair.

Computing TF-IDF score for Reviews

MRBUP defines TF-IDF for a term 't' in review $r_i \in R$ using equation (4) & (5) as:

$$TF(t, r_i) = \frac{No.\ of\ times\ t\ appears\ in\ r_i}{Total\ no.\ of\ terms\ in\ r_i} \tag{4}$$

$$INIF(t, R) = \log\left(\frac{|R|}{r_i \in R : t \in r_i}\right) \tag{5}$$

The weight of every important term in the review is given by:

$$Wt\ (r_i) = [t_0, TF(t_0, r_i) \times INIF(t_0, R);\ t_1, TF(t_1, r_i) \times INIF(t_1, R); \ldots\ldots\ ] \tag{6}$$

All those terms whose TF-IDF score is above the user defined threshold are considered important and are included as a part of user profile.

The MapReduce based algorithm for generating user profile using TF-IDF is executed in three successive MapReduce stages. In the first phase, term frequency is computed using equation (4). ReviewID along with review is provided as an input in the first phase. In the second phase,

Inverse Document Frequency (IDF) is computed using equation (5) and returns the weight formulated in equation (6). The pseudo-code for the three MapReduce phases are shown in Algorithm 1.

Algorithm 1 Phase 1 of MapReduce Task.

Input: <review_ID: review>
Output: <term: review_ID, count>

```
for each review_ID, review:
    for each term in review:
        emit(term, review_ID, 1)

for each term:
    total_count = sum(counts)
    emit(term, review_ID, total_count)
```

Algorithm 1 Phase 2 of MapReduce Task.

Input: <term: review_ID, count>
Output: <term: review_ID, tf_idf>

```
for each term, review_ID, count in input:
    emit(term, (review_ID, count))

for each term:
    review_count = count(distinct (review_IDs)
    idf = log(total_reviews / (1 + review_count))
        for each review_ID, count in values:
            tf_idf = count * idf
            emit(term, (review_ID, tf_idf))
```

Algorithm 1 Phase 3 of MapReduce Task.

Input: < term: review_ID, tf_idf >
Output: < review_ID: (term, tf_idf)>

```
for each term, review_ID, tf_idf in input:
        emit(term, (review_ID, (term, tf_idf)
for each review_ID:
        emit(review_ID, list of (term, tf_idf))
    emit(review_ID, list of (term, tf_idf))
```

For every user in the system, user profile includes the list of terms for which the TF-IDF score is above a threshold value. The relevance of TF-IDF for user profiling comes from the fact that this technique evaluates the importance of a term in a review relative to the entire pool of reviews given by the user.

4. Experiments & Results

This section evaluates the proposed methodology of creating user profiles from the explicit feedbacks given in the form of reviews. (Speed up) data set description.

4.1 *Dataset*

The dataset on which the experiments are performed is the Amazon Product Reviews [18]. In this dataset, all the products are categorized into various categories such as Arts, Automotive, Electronics to name a few. This dataset comprises of reviews given by the user to the different products available on amazon. The total number of reviews across all the categories are 34,686,770. The number of users are products are 6,643,669 and 2,441,053 respectively.

4.2 *Experimental Setup & Evaluation Metrics*

Experiments are carried out to determine the scalability of MRBUP by measuring the amount of time needed to generate a user profile. For this, Hadoop 0.20.2 is made to run in a standalone mode (with single node) and in a cluster set up with three and five nodes. To evaluate the scalability, Speedup factor is computed as defined in equation (7).

$$S_t = \frac{T_1}{T_t} \tag{7}$$

In equation (7), the numerator refers to the amount of time the system takes to generate user profile when there is single node in the cluster and T_t is the amount of time taken to generate user profile with 't' number of nodes in the cluster.

4.3 *Results & Discussions*

The computational time with different cluster size and the number of reviews is shown is Table 1 and is pictorially shown in Figure 1. Table 2

reports the speedup factor for MRBUP which is also pictorially represented in Figure 2.

Figure 1 indicates that as the size of the Hadoop cluster increases, so does the computing time needed to generate a user profile (with a given dataset size). Computational time increases very little when the size of the Hadoop cluster is fixed. The scalability of MRBUP is ensured by the speedup factor, which grows significantly with the size of the Hadoop cluster, as demonstrated in Figure 2.

Table 1

Computational time (seconds) for MRBUP

No. of Reviews	Single Node Cluster	Cluster with 3 Nodes	Cluster with 5 Nodes
27,980	162	140	125
58,621	216	167	145
137,042	405	195	173

Table 2

Speedup Factor for proposed approach

No. of Reviews	S_5	S_3
27,980	1.2	1.1
58,621	1.4	1.2
137,042	2.3	2.0

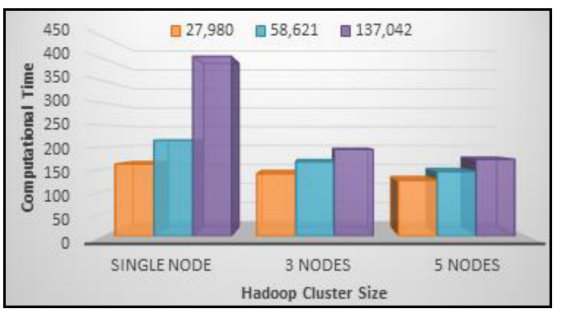


Figure 1

Computational Time (seconds) for MRBUP. Impact of changing cluster and dataset sizes on computing time

5. Conclusion and Future Work

Presenting the appropriate data to the user who is looking for it becomes crucial in order to solve the issue of information overload. Understanding the user's preferences and what he would want to view from the large amount of information is a crucial first step in this approach. Creating a user profile for the user provides an understanding of their preferences and likes. This is the first step in understanding the user. This study introduced MRBUP, a scalable Hadoop MapReduce-based user profile creation method. To generate user profiles, MRBUP took advantage of the concept of term frequency-inverse document frequency. The suggested method made advantage of the user-provided product reviews to generate user profile. Terms that the user finds interesting were included in their user profile. MRBUP is used to Hadoop clusters of different sizes as well as dataset sizes. To assess the system's scalability, the computing time of MRBUP was tracked, and a speedup factor was calculated. The findings showed that MRBUP scaled efficiently to accommodate the increasing amount of data items as the size of the Hadoop cluster increased. On the basis of the created user profile, MRBUP can be further expanded to offer recommendations to the user.

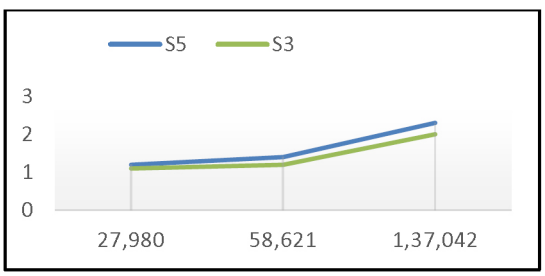


Figure 2

Speedup Factor for MRBUP

References

[1] Mekouar, Loubna, Youssef Iraqi, and Issam Damaj. "A global user profile framework for effective recommender systems." *Multimedia Tools and Applications*, 1-21, (2023). DOI: https://doi.org/10.1007/s11042-023-17436-w.

[2] Gupta, Shaurya, et al. "Tier application in multi-cloud databases to improve security and service availability." Handbook of research on

cloud computing and Big Data applications in IoT. IGI Global, 82-93 (2019).

[3] Purohit, Neha, Joshi, Shubhalaxmi, Pande, Milind & Lincke, Susan. "Pragmatic analysis of ECC based security models from an empirical perspective." *Journal of Discrete Mathematical Sciences and Cryptography,* 26:3, 739-758. (2023). DOI: 10.47974/JDMSC-1749.

[4] Jena, Biswaranjan, Mishra, Debahuti & Mishra, Smita Prava. "RDBMS to distributed in-memory data grid: A paradigm shift for improved performance." *Journal of Statistics and Management Systems*, 26:1, 169-175, (2023). DOI: 10.47974/JSMS-956.

[5] Betancourt, Yanelys, and Sergio Ilarri. "Use of Text Mining Techniques for Recommender Systems." In ICEIS (1), pp. 780-787. (2020). DOI: 10.5220/0009576507800787.

[6] Peterson, Jacob. "User Profiling in Recommender Systems." (2021). DOI:https://umm-csci.github.io/senior-seminar/seminars/spring2021/peterson.pdf.

[7] Sharma, S., Raja, L., Bhatnagar, V., Sharma, D., Bhagirath, S. N., & Poonia, R. C. Hybrid HOG-SVM encrypted face detection and recognition model. *Journal of Discrete Mathematical Sciences and Cryptography*, 25(1), 205–218 (2022). https://doi.org/10.1080/09720529.2021.2014141.

[8] Anjali, Anjali, Jasminder Kaur Sandhu, and Deepam Goyal. "User profiling in travel recommender system using hybridization and collaborative method." In 2021 international conference on computing, communication, and intelligent systems (ICCCIS), pp. 143-148. IEEE, (2021). DOI:10.1109/ICCCIS51004.2021.9397099.

[9] Hernandez-Bocanegra, D.C. and Ziegler, J., Explaining review-based recommendations: Effects of profile transparency, presentation style and user characteristics. i-com, 19(3), pp.181-200 (2021). DOI: https://doi.org/10.1515/icom-2020-0021.

[10] Bilal, Muhammad, et al. "Profiling users' behavior, and identifying important features of review "helpfulness"." IEEE Access 8 (2020): 77227-77244. DOI: 10.1109/ACCESS.2020.2989463.

[11] Liang, Huizhi. "DRprofiling: Deep reinforcement user profiling for recommendations in heterogenous information networks." *IEEE Transactions on Knowledge and Data Engineering* 34.4 (2020): 1723-1734. DOI: 10.1109/TKDE.2020.2998695.

[12] Sahu, Ashish Kumar, and Pragya Dwivedi. "User profile as a bridge in cross-domain recommender systems for sparsity reduction." *Applied Intelligence* 49 : 2461-2481 (2019). DOI: https://doi.org/10.1007/s10489-018-01402-3.

[13] Zhu, Feng, et al. "Cross-domain recommendation: challenges, progress, and prospects." arXiv preprint arXiv:2103.01696 (2021). DOI: https://doi.org/10.48550/arXiv.2103.01696.

[14] Zang, Tianzi, et al. "A survey on cross-domain recommendation: taxonomies, methods, and future directions." *ACM Transactions on Information Systems* 41.2 : 1-39 (2022). DOI: https://doi.org/10.1145/3548455.

[15] Lu, Zhongqi, Sinno Jialin Pan, Yong Li, Jie Jiang, and Qiang Yang. "Collaborative evolution for user profiling in recommender systems." In IJCAI, pp. 3804-3810 (2016). DOI: https://www.ijcai.org/Proceedings/16/Papers/535.pdf.

[16] Gu, Yulong, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. "Hierarchical user profiling for e-commerce recommender systems." In Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 223-231 (2020). DOI: https://doi.org/10.1145/3336191.3371827.

[17] Jung, Haemin, Heesung Park, and Kwangyon Lee. "Enhancing Recommender Systems with Semantic User Profiling through Frequent Subgraph Mining on Knowledge Graphs." Applied Sciences 13, no. 18 : 10041 (2023). DOI: https://doi.org/10.3390/app131810041.

[18] McAuley, Julian, and Jure Leskovec. "Hidden factors and hidden topics: understanding rating dimensions with review text." In Proceedings of the 7th ACM conference on Recommender systems, pp. 165-172 (2013). DOI: https://doi.org/10.1145/2507157.2507163.