ClubCF Approach for Big Data Application using Hadoop and Comparative Study with Item-Based CF
D. V. Chandran¹, Pranali Mahadik², Anuja Malvankar³, Ajay Gathadi⁴
Professor¹, BE Student² ³ ⁴
Department of Computer Science & Engineering
Smt. Indira Gandhi college of Engineering, Navi, Mumbai, India

Abstract:
Inspired from service computing and cloud computing, an increasing number of services are increasing on the Internet. Because of this, service-related data become too big to be effectively processed by existing approaches. In view of this challenge, a clustering-based collaborative filtering approach is proposed in this paper, which aims at extracting similar services in the same clusters to recommend services collaboratively. Technically, this approach is enacted around two stages. In the first stage, the available services are divided into small-scale clusters, in logic, for further processing. At the second stage, a collaborative filtering algorithm is applied on one of the clusters. Since the number of the services in a cluster is much less than the total number of the services available on the web, it is expected to reduce the online execution time of collaborative filtering.

Keywords: Hadoop, Big data, ClubCf, Item-Based

IINTRODUCTION

The unique dimension for big data is: Volume, Velocity, Variety and Veracity. Generally speaking, Big Data concerns large-volume, complex, growing data sets with multiple, autonomous sources. Big Data applications where data collection has grown tremendously and is beyond the ability of commonly used software tools to capture, manage, and process within a “tolerable elapsed time” is on the rise. The most fundamental challenge for the Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions. With the spread of service computing and cloud computing, more and more services are deployed in cloud infrastructures to provide rich functionalities. Service users have nowadays encountered unprecedented difficulties in finding ideal ones from the overwhelming services. Recommender systems (RSs) are techniques and intelligent applications to assist users in a decision making process where they want to choose some items among a potentially overwhelming set of alternative products or services. Collaborative filtering (CF) such as item- and user-based methods are the dominant techniques applied in RSs. The basic assumption of user-based CF is that people who agree in the past tend to agree again in the future. Different with user-based CF, the item-based CF algorithm recommends a user the items that are similar to what he/she has preferred before. Although traditional CF techniques are sound and have been successfully applied in many e-commerce RSs, they encounter two main challenges for big data application:

1) To make decision within acceptable time;
2) To generate ideal recommendations from so many services.

Concretely, as a critical step in traditional CF algorithms, to compute similarity between every pair of users or services may take too much time, even exceed the processing capability of current RSs. Consequently, service recommendation based on the similar users or similar services would either lose its timeliness or couldn't be done at all. In addition, all services are considered when computing services' rating similarities in traditional CF algorithms while most of them are different to the target service. The ratings of these dissimilar ones may affect the accuracy of predicted rating.

II.PROBLEM STATEMENT

Now Days, services useful to peoples are increasing in great extent on internet. As a result, service-relevant data become too big to be effectively handled by existing approaches. At a time, many people use the same service through different location on internet. Services on the internet contains large amount of data size of around 10¹² Giga Bytes. Existing approaches are not able to handle and control this data effectively

2.1 EXISTING SYSTEM

In the existing system with the prevalence of service computing and cloud computing, more and more services are deployed in cloud infrastructures to provide rich functionalities. Service users have nowadays encountered unprecedented difficulties in finding ideal ones from the overwhelming services. Recommender systems (RSs) are techniques and intelligent applications to assist users in a decision making process where they want to choose some items among a potentially overwhelming set of alternative products or services. Collaborative filtering (CF) such as item- and user-based methods are the dominant techniques applied in RSs. The most fundamental challenge for the Big Data applications is to explore the large volumes of data and extract useful information or knowledge for future actions. The basic assumption of user-based CF is that people who agree in the past tend to agree again in the future. Different with user-based CF, the item-based CF algorithm recommends a user the items that are similar to what he/she has preferred. Consequently, service recommendation based on the similar users or similar services would either lose its timeliness or could not be done at all.

International Journal of Engineering Science and Computing, April 2017
Disadvantages of Existing System:
- Much Time to Search and data Clustering Management is poor performance.
- In this system use a specific one type of recommendation technique so the recommendation is poor.

2.2 PROPOSED SYSTEM
We propose a Clustering-based Collaborative Filtering approach (ClubCF), which consists of two stages: clustering and collaborative filtering. Clustering is a pre-processing step to separate big data into manageable parts. A cluster contains some similar services just like a club contains some like-minded users. This is another reason besides abbreviation that we call this approach ClubCF. Since the number of services in a cluster is much less than the total number of services, the computation time of CF algorithm can be reduced significantly. Besides, since the ratings of similar services within a cluster are more relevant than that of dissimilar services, the recommendation accuracy based on users’ ratings may be enhanced.

Advantages of Proposed system
- Get Recommendation from overwhelming candidates within an acceptable time.
- Avoid Reduplication Data’s and improve performance the cluster Filtering.
- Recommender Service System is Provide our knowledge provide Based from the users preferences.
- The time of rating similarity computation between every pair of services will be greatly reduced.
- The recommendation accuracy based on users’ ratings may be enhanced.

The time complexity of ClubCF can be divided into two parts:
1) The offline cluster building; and
2) The online collaborative filtering.

III.WORKFLOW OF SYSTEM

IV.DEPLOYMENT OF CLUSTERING STAGE
A. Stem Words

Figure.2. Stem Words
Different developers may use different-form words to describe similar services. Using these words directly may influence the measurement of description similarity. Therefore, description words should be uniformed before further usage. In fact, morphological similar words are clubbed together under the assumption that they are also semantically similar. For example, “map”, “maps”, and “mapping” are forms of the equivalent lexeme, with “map” as the morphological root form. To transform variant word forms to their common root called stem, various kinds of stemming algorithms, such as Lovins stemmer, Dawson Stemmer, Paice/Husk Stemmer, and Porter stemmer, have been proposed [13]. Among them, Porter stemmer is one of the most widely used stemming algorithms. It applies cascaded rewrite rules that can be run very quickly and do not require the use of a lexicon. In ClubCF approach, the words in $D_t$ are gotten from service Big table where row key D “st” and column family D “Description”. The words in $D_j$ are gotten from service Big table where row key D “sj” and column family D “Description”. Then these words are stemmed by Porter Stemmer and put into $D_0 t$ and $D_0 j$, respectively.

B: Compute Description Similarity And Functionality Similarity
Description similarity and functionality similarity are both computed by Jaccard similarity coefficient (JSC) which is a statistical measure of similarity between samples sets. For two sets, JSC is defined as the cardinality of their intersection divided by the cardinality of their union. Concretely, description similarity between stand sj is computed by formula

C: Compute Characteristic Similarity
Characteristic similarity between stand sj is computed by weighted sum of description similarity and functionality similarity, which is computed as follow:

$$C_{sim}(st; sj)=\alpha XD_{sim}(st; sj)+\beta XF_{sim}(st; sj) \quad (3)$$

In this formula, $\alpha \in [0; 1]$ is the weight of description similarity, $\beta \in [0; 1]$ is the weight of functionality similarity and $\alpha+\beta=1$. The weights express relative importance between these two. Provided the number of services in the recommender system is n, characteristic similarities of every pair of services are calculated and form a n X n characteristic similarity matrix D. An entry $d_{rj}$ in D represents the characteristic similarity between stand sj.
D: Cluster Services
Clustering is a critical step in our approach. Clustering methods partition a set of objects into clusters such that objects in the same cluster are more similar to each other than objects in different clusters according to some defined criteria. Generally, cluster analysis algorithms have been utilized where the huge data are stored. Clustering algorithms can be either hierarchical or partitioned. Some standard partitional approaches (e.g., K-means) suffer from several limitations:

1) Results depend strongly on the choice of number of clusters \( K \), and the correct value of \( K \) is initially unknown;
2) Cluster size is not monitored during execution of the \( K \)-means algorithm, some clusters may become empty (“collapse”), and this will cause premature termination of the algorithm;
3) Algorithms converge to a local minimum.
Hierarchical clustering methods can be further classified into agglomerative or divisive, depending on whether the clustering hierarchy is formed in a bottom-up or top-down fashion. Many current state-of-the-art clustering systems exploit agglomerative hierarchical clustering (AHC) as their clustering strategy, due to its simple processing structure and acceptable level of performance. Furthermore, it does not require the number of clusters as input. Therefore, we use an AHC algorithm, for service clustering as follow. Assume there are \( n \) services. Each service is initialized to be a cluster of its own. At each reduction step, the two most similar clusters are merged until only \( K \) \((K < n)\) clusters remains.

Algorithm 1: AHC algorithm for service clustering

**Input:** A set of services \( S = \{s_1, \ldots, s_n\} \)

- a characteristic similarity matrix \( D = [d_{ij}]_{n \times n} \)
- the number of required clusters \( K \)

**Output:** Dendrogram, for \( k = 1 \) to \( |S| \)

1. \( C_i = \{s_i\} \), \( \forall i \)
2. \( d_{C_i,C_j} = d_{ij}, \forall i,j \)
3. for \( k = |S| \) down to \( K \)
4. Dendrogram \( = \{ C_1, \ldots, C_k \} \)
5. \( l_m = d_{C_m,C_m} \)
6. \( C_1 = \text{Join}(C_1, C_m) \)
7. for each \( C_k \in S \)
   8. if \( C_k \neq C_1 \) and \( C_k \neq C_m \)
   9. \( d_{C_1,C_k} = \text{Average}(d_{C_1,C_4}, d_{C_m,C_k}) \)
10. end if
11. \( S = S - \{C_m\} \)
12. end for

Algorithm: Agglomerative Hierarchical Algorithm
Comparison study between ClubCF and Item based CF: ClubCF spends less computation time than Item based CF.
Since the number of services in a cluster is fewer than the total number of services, the time of rating similarity computation between every pair of services will be greatly reduced.

V. IMPLEMENTATION

5.1 Hadoop Based Result for product: This page shows time required to data connection in Hadoop in product website & search result & time is 472ms which is less than item based.

5.2 Item based result for product: This page shows time required to data connection in item based in product website & search result & time is 1072ms which is more than Hadoop.

VII. FUTURE WORK

Future research can be done in two areas. First, in the respect of service similarity, semantic analysis may be performed on the description text of service. In this way, more semantic-similar services may be clustered together, which will increase the coverage of recommendations. Second, with respect to users, mining their implicit interests from usage records or reviews may be a complement to the explicit interests (ratings). By this means, recommendations can be generated even if there are only few ratings. This will solve the sparsity problem to some extent.
VIII. REFERENCES


