ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

Privacy Enhanced Clustering Based Collaborative Filtering of Services for Big Data Application

[1]Dr. Ramakrishna V Moni, [2]Akshay M Nandagaon

[1]Professor, Dept. of CSE, [2]PG Student, Dept. of CSE Sambhram Institute Of Technology Bengaluru, Karnataka, India

***Abstract*: Web services provide new types of services leading toan increasing number of services emerging on internet in service and cloud computing. As a result, service-relevant data has become too huge to effectivelyprocess and users are finding difficultyto utilizeservices that match their mash ups. To overcome this challenge technique used is to filter and recognize the similar services under clusters and recommend by calculating their similarities, which is referred fromClustering Based Collaborative Filtering Approach for Recommendation System (ClubRS) technique. User's data are always privacy sensitive and can be misused by the service provider while generating recommendations. In this paper we aim to protect the private data from the service provider while preserving the functionality of the system.Wepropose a cryptographic solution for preserving privacy of customer's data in recommender system. In short, private information of customer is protected and service provider generates recommendation by processing encrypted data.The proposed method is based on homomorphic encryption schemes. Later in the first stage,the encrypted services are divided into small clusters of similar services using agglomerative clustering algorithm (AHC). In the second stage, collaborative filtering algorithm is applied on one of the clusters and recommendations are made using decryption algorithm. Since the number of services in a cluster is much less than the total number of the services provided, it is expected to alsoenhance the execution time of collaborative filtering.**

***Keywords:* big data application, cluster, collaborative filtering, mashup, homomorphic encryption, privacy.**

I. INTRODUCTION

Big data has risento become known as a commonly perceived trend, attracting consideration from wide areas such as government institutions, industrial areas and academic research. Technically speaking, Big Data concerns in exploring larger-volume, complex, increasing data sets with multiple, autonomous sources [1]. Big Data applications where data collection has grown immense and is beyond the capability of commonly used software tools to capture, manage, modify, and process within a tolerable elapsed time [2]. Most common fundamental challenge is to explore the large volumes of data and extract and process useful knowledge or information for future use and perform actions for Big Data applications[3].

The basic yet simple statement of user-based Collaborative Filtering is that people who have the same opinion in the past is likely to agree again in the future. Unlike with user-based CF, the item-based CF algorithm recommends the items to the user that is similar to what he/she has chosen in earlier period.[6]. Even though traditional CF techniques are perfect and have been effectively applied in many e-commerce RSs applications, they encounter two major challenges for big data application: 1) making the decisions within tolerable time; and 2) to provide best recommendations from various services. Concretely, as a significant step in traditional CF algorithms, to calculate similarity between each and every pair of users/services may take much larger time, even surpassing the processing ability of current Recommender Systems (RSs). Accordingly, service recommendation based on the parallel users/services would whichever lose its timeliness or couldn't be completed at all. In accumulation, all services are considered while computing service's rating similarities in

traditional Collaborative Filtering (CF) algorithms while the majority of them are dissimilar to the target service[1]. The accuracy of predicted rating is affected by ratings of these dissimilar ones[5].

A naive solution must decrease the number of services that require to be processed in valid time. Clustering is technique that can decrease the size of data by a large feature by grouping of number similar services together[4]. As a result, we propose Cluster-based Recommendation of Services using Collaborative Filtering with enhancing privacy of the user, which consists of two stages: clustering stage and collaborative filtering stage. Clustering is a pre-processing step to cluster or separate big data into convenient parts to handle. A cluster contains some similar services just like a like-minded user in a club. This is another reason that why we call this approach ClubRS. Since the cluster contains the services much less than the total number of services, the computation time can be reduced significantly for Collaborative Filtering (CF) algorithm. Moreover, since the cluster contains similar services that have similar ratings are more related than that of dissimilar services, the enhancement of recommendation accuracy is based on user's ratings[7].

More information on the user helps the system to improve the accuracy of the recommendations. On the other hand, the information on the users creates a severe privacy risk since there is no solid guarantee for the service provider not to misuse the user's data. The personal information on the users creates a severe privacy risk since there is no solid guarantee for the service provider not to misuse the users' data[16]. It is often seen that whenever a user login in the system, service providers statements the ownership of data provided from user and approves itself to allocate the data to third party. In ClubRS, we propose a cryptographic solution for preserving the privacy of User's in a recommender system. In particular, the privacy-sensitive data of the users are kept encrypted and the service provider generates recommendations by processing encrypted data. The cryptographic protocol developed for this purpose is based on homomorphic encryption[17]. We aim at protecting the privacy of the users against the Service provider by means of encrypting the private data and generate recommendations in the encrypted domain by running cryptographic protocols. The personal information on the users creates a severe privacy risk since there is no solid guarantee for the service provider not to misuse the users' data. It is often seen that whenever a user login in the system, service providers statements the ownership of data provided from user and approves itself to allocate the data to third party[18].

II. RELATED WORK

Clustering methods for CF have been extensively studied by some researchers. Mai et al. [9] designed a neural networks-based clustering collaborative filtering algorithm in e-commerce recommendation system. The cluster analysis gathers users with similar characteristics according to the web visiting message data. However, it is hard to say that a user's preference on web visiting is relevant to preference on purchasing. Mittal et al. [10] proposed to achieve the predictions for a user by first minimizing the size of item set the user needed to explore. *K*-means clustering algorithm was applied to partition movies based on the genre requested by the user. However, it requires users to provide some extra information. Li et al. [11] proposed to incorporate multidimensional clustering into a collaborative filtering recommendation model. Background data in the form of user and item profiles was collected and clustered using the proposed algorithm in the first stage. Then the poor clusters with similar features were deleted while the appropriate clusters were further selected based on cluster pruning. At the third stage, an item prediction was made by performing a weighted average of deviations from the neighbour's mean. Such an approach was likely to trade-off on increasing the diversity of recommendations while maintaining the accuracy of recommendations. Zhou et al. [12] represented Data-Providing (DP) service in terms of vectors by considering the composite relation between input, output, and semantic relations between them. The vectors were clustered using a refined fuzzy C-means algorithm. Through merging similar services into a same cluster, the capability of service search engine was improved significantly, especially in large Internet-based service repositories. However, in this approach, it is assumed that domain ontology exists for facilitating semantic interoperability. Besides, this approach is not suitable for some services which are lack of parameters. Pham et al. [13] proposed to use network clustering technique on social network of users to identify their neighbourhood, and then use the traditional CF algorithms to generate the recommendations. This work depends on social relationships between users. Simon et al. [14] used a high-dimensional parameter-free, divisive hierarchical clustering algorithm that requires only implicit feedback on past user purchases to discover the relationships within the users. Based on the clustering results, products of high interest were recommended to the users. However, implicit feedback does not always provide sure information about the user's preference.

Paillier's scheme in which the expansion factor is reduced and which allows to adjust the block length of the scheme even after the public key has been fixed, without losing the homomorphic property[15]. It shows that the generalization is as secure as Paillier's original system and proposes several ways to optimize implementations of both the generalized. We

build a threshold of the generalized as well as zero-knowledge protocols to show that a given cipher text encrypts one of a set of given plain texts, and algorithms are to verify multiplicative on plaintexts. Then it also shows how these building blocks can be used for applying the scheme to efficient process. This reduces dramatically the work needed to compute the final result of a process, compared to the previously best known schemes. It shows how the basic scheme for process can be easily adapted to casting a vote for up to timeout of L candidates. The same basic building blocks can also be adapted to provide receipt-free elections, under appropriate physical assumptions. The scheme for 1 out of L process can be optimized.

III. PROPOSED SYSTEM

A ClubRS approach for Big Data application is offered, which aims to recommend services from irresistible candidates within an acceptable time. Technically, ClubRS aims on two inter-dependable stages, i.e., clustering stage and collaborative filtering stage. In the first stage, services are clustered according to calculated characteristic similarities. In the second stage, a collaborative filtering algorithm is applied within a cluster that an intended service belongs to the assumption that services are also semantically similar.

ClubRS approach for big data applications are appropriate torecommend service. Prior to applying CF technique, services are merged into several clusters via an AHC algorithm. Then rating similarities between services within the same cluster are calculated. While the number of services in a cluster is much lesser than that of in the entire system on web, ClubRS costs lesser online elapse time. Furthermore, as the ratings of services in the same cluster are more significant with each other than with the ones in other clusters, prediction based on the ratings of the services in the same cluster will be more accurate than based on the ratings of all similar or dissimilar services in all clusters.

In ClubRS, we consider a scenario where users of an online service receive personalized recommendations, which are generated using collaborative filtering techniques [1]. In this scenario, we aim at protecting the privacy of the users against the service provider by means of encrypting the private data, that is users' ratings, and to generate recommendations in the encrypted domain by running cryptographic protocols, which is an approach similar to [16]. The output of the cryptographic protocol, as well as the intermediate values in the algorithm, is also private and not accessible to the service provider. It is important to note that while generating recommendations by processing encrypted data is possible, the difficulty lies in realizing efficientprivacy-preserving protocols. Our goal is to provide a more efficient privacy-preserving recommender system by improving the state-of-the-art further.

IV. IMPLEMENTATION

HOMOMORPHIC ENCRYPTION:

A number of public-key cryptosystems are additively Homomorphic, meaning that there exists an operation on cipher texts such that the result of that operation corresponds to a new cipher text whose decryption yields the sum of the messages. Paillier and DGK scheme are two additively Cryptosystems used.

Algorithm 1: Encryption algorithm

INPUT: Mashup Services.
OUTPUT: Encrypted Message
1. Begin
2. Pick random big Integer P and Q value (i.e that must be probable prime number)
3. Compute N
4. Pick random bigInteger K value (i.e the value between (1-n-1))
5. Compute B1=K*P;
6. Then compute M value (i.e get the plain text in form of bytes means plaintext getBytes)
7. Compute B2=M+B1
8. for end
9. Encrypt message = B1, B2
10. End

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

Algorithm 2: Decryption algorithm

INPUT: Encrypted Message
OUTPUT: Decrypted Message
1. Begin
2. Pick the encrypted msg
3. Then split the msg by (,)
4. Get the split [0] value as B1 value and Split [1] value as B2 value
5. Then
6. Compute M value (i.e M=B2-B1)
7. Convert m value into byte Array
8. for end
9. Get the decrypted msg
10. End

Paillier cryptosystem is used to encrypt privacy sensitive data of user. DGK cryptosystem is used for sub-protocol. DGK is more efficient in terms of encryption and decryption compared to paillier due to its smaller message space of a few bits.

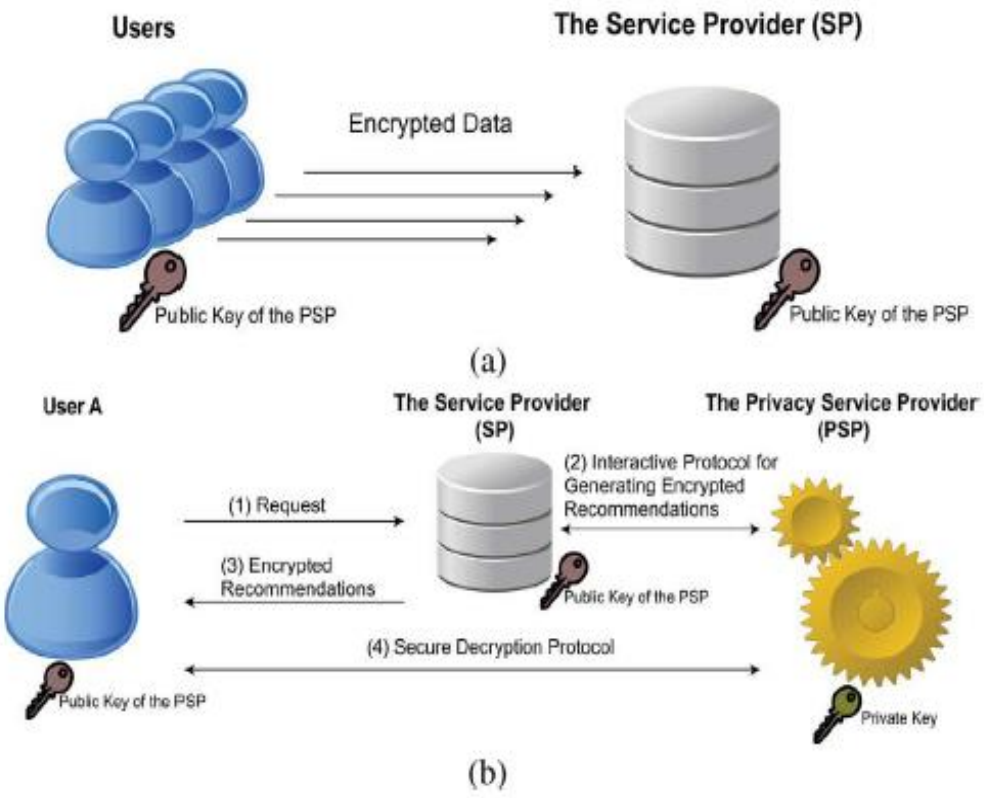


Fig. 1: System model for generating private recommendations.

(a) Encrypted database construction (b) generating private recommendation

The Service Provider (SP): has a business interest in generating recommendations for his customers. He has resources for storage and processing.

The Privacy Service Provider (PSP): is a semi trusted third party who has a business interest in providing processing power and privacy functionality. The PSP has private keys for the Paillier and the DGK cryptosystems.

Users: are the customers of the service provider. Based on their preferences, in the form of ratings, the service provider generates recommendations for them.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

ClubRS focuses on two inter dependable stages, i.e., clustering stage and collaborative filtering stage. In first stage, services are clustered by calculating their characteristic similarities and applying the Agglomerative hierarchical algorithm (AHC). In the second stage, inside a cluster a collaborative filtering algorithm is applied that a target service belongs to recommend.

A. CLUSTERING STAGE:

Step 1: Stem Words:

To describe similar services different developers use different kind of words to name them. By using these words to similar services may affect directly to the measurement of description similarity. For that reason description words should be uniformed before they are used. In fact, morphological similar words are clustered mutually under the assumption that they are semantically similar. For example, "map", "maps", and "mapping" are forms of the corresponding lexeme, with "map" as the morphological root form. To convert variation word forms to their universal root called stem, a variety of stemming algorithms, such as Lovins stemmer, Dawson Stemmer, Paice/Husk Stemmer, and Porter Stemmer, has been proposed and Porter Stemmer is one of the most popular and frequently used stemming algorithms. It applies cascaded rewrite rules that can be used to run very quickly and use of lexicon is not required. In ClubRS approach, the words in Dt are extracted from service table where row key = "St" and column family = "$Description$". The words in Dj are extracted from service table where row key = "Sj" and column family = "$Description$". Then Porter Stemmer is used to stem these words and put into Dt' and Dj', respectively [1].

Step 2: Computes Description Similarity and Functionality Similarity:

Description similarity and functionality similarity are both calculated by Jaccard similarity coefficient (JSC) which is a statistical measure of similarity between samples sets that are referred. For two sets, JSC is distinct as the cardinality of their intersection divided by the cardinality of their combination. Approximate, Description similarity between St and Sj is computed by equation (1).

$$D\_sim(s_t, s_j) = \frac{|D_t' \cap D_j'|}{|D_t' \cup D_j'|} \quad (1)$$

From the above equation that the larger $|Dt' \bigcap Dj'|$ is, the more similar these two services are with each other. Dividing by $|Dt' \bigcup Dj'|$ is the scaling factor that proves the description similarity ranges from 0 and 1.

The functionalities in Ft are extracted from services where row key = "St" and column family = "$Functionality$". The functionalities in Fj are extracted from services where row key = "Sj" and column family = "$Functionality$". Then, functionality similarity between St and Sj is computed using JSC as follows [1]:

$$F\_sim(s_t, s_j) = \frac{|F_t' \cap F_j'|}{|F_t' \cup F_j'|} \quad (2)$$

Step 3: Compute Characteristic Similarity:

Characteristic similarity between St and Sj is computed by weighted sum of description similarity and functionality similarity, which is calculated as follows:

$$C\_sim(s_{t,}s_j) = \alpha \times D\_sim(s_{t,}s_j) + \beta \times F\_sim(s_{t,}s_j) \quad (3)$$

From the above equation, $\alpha \in [0,1]$ is the weight of description similarity, $\beta \in [0,1]$ is the weight of functionality similarity and $\alpha+\beta=1$. The weights state relative magnitude between these two. Given that the number of services in the recommender system is n, characteristic similarities of every pair of services are computed and form a $n \times n$ characteristic similarity matrix D. A entry dt, in D represents the characteristic similarity between St and Sj [1].

Step 4: Cluster Services:

Clustering is a vital step in this approach. Set of objects in a cluster that are similar to each other are separated with set of objects that are dissimilar by means of clustering methods according to some defined criterion.

In general, clustering analysis algorithms have been utilized where the large amount data are stored. Clustering algorithms can be either hierarchical or partitioned. Some typical partitioned approaches (e.g., K-means) suffer from several

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

boundaries: 1) results strictly depend on the selection of number of clusters *K*, and the correct value of *K* is primarily unknown; 2) during the execution cluster size is not monitored while executing the *K*-means algorithm, some clusters may turn out to be empty ("collapse"), and this will cause untimely termination of the algorithm; 3) algorithms meet to a local minimum. Further hierarchical clustering methods can be separated into agglomerative or divisive, depends on either the clustering hierarchy is formed in a bottom-up or top-down fashion. Many clustering systems which are currently been used acquire agglomerative hierarchical clustering (AHC) for clustering strategy, because of its simple processing structure and tolerable level of performance [1]. Furthermore, it does not necessitate the number of clusters as input. As a result, use of AHC algorithm for service clustering as follow. Assuming there are n services. Each service is initialized to be an own cluster. At every reduction step, the two most similar clusters are combined until only K (K<n) clusters remains.

Algorithm 3: AHC algorithm for service clustering

Input: A set of services $S=$ $s1,\ldots,$, a characteristic similarity matrix $D=$ di,j $n\times n$, the number of required clusters K.
Output: Dendrogram_k for k=1 to S.
1. $Ci=$ S$i,\forall i$;
2. $dCi, Cj=di,,\forall i,j$;
3. **for**$k=S$ **down to** K
4. $\text{Dendrogram}_k=C1,\ldots,$;
5. $l,=argmaxi,jdCi,Cj$;
6. $Cl=Join\ Cl,$;
7. **for each** Ch$\in S$
8. **if**Ch$\neq Cl$ and Ch$\neq Cm$
9. $dCl,$h=av$erage$ $(dCl,C$h$,dCm,C$h$)$;
10: **end if**
11: **end for**
12. $S=S-Cm$;
13. **end for**

B. COLLABORATIVE FILTERING STAGE:

Up till now, item-based collaborative filtering algorithms have been extensively used in many real world applications such as at Amazon.com. It is divided into three main steps, i.e., compute rating similarities; select neighbours and recommend services [1].

Step 1: Compute Rating Similarity:

Rating similarity computation between items is a time-consuming but vital step in item-based CF algorithms. Ordinary rating similarity measures comprise the Pearson correlation coefficient (PCC) and cosine similarity between ratings vectors. The essential perception behind PCC measure is to give a larger similarity score for two items that likely to be rated the equal by many users. PCC which is the ideal choice in most key systems was found to perform improved than the cosine vector similarity. As a result, PCC is applied to compute rating similarity between each and every pair of services in ClubRS. Given that service St and Sj are both belong to the same cluster, PCC-based rating similarity between St and Sj is computed by formula (4):

$$R\_sim(s_t, s_j) \frac{\sum u_i \epsilon u_t \cap u_j \, (r_{u_i,s_t} - \overline{r_{s_t}})(r_{u_i,s_j} - \overline{r_{s_j}})}{\sqrt{\sum u_i \epsilon u_t \cap u_j \, (r_{u_i,s_t} - \overline{r_{s_t}})^2} \sqrt{\sum u_i \epsilon u_t \cap u_j (r_{u_i,s_j} - \overline{r_{s_j}})^2}} \quad (4)$$

Here, Ut are number of users who rated St whereas Uj are number of users who rated Sj, ui is a user who has rated both St and Sj, rui,st is the rating of St given by ui which is derived from services where row key = "St" and column key = "$Rating$: ui", rui,sj is the rating of Sj given by ui which is derived from service where row key = "Sj" and column key = "$Rating$:ui", $\overline{r_{s_t}}$ is the average rating of St, and Sj is the average rating of Sj. It must be well-known that if the denominator of formula (6.4) is zero, the result will also be 0, in order to avoid division by 0 [1].

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

Even though PCC can offer precise similarity computation, it may overrate the rating similarities when there is a minimum amount of co-rated services. To overcome this problem, the improved rating similarity between St and Sj is computed by formula (5):

$$R\_sim'(s_t, s_j) = \frac{2 \times |U_t \cap U_j|}{|U_t| + |U_j|} \times R\_sim(s_t, s_j) \quad (5)$$

In this formula, $|Ut \cap Uj|$ is the number of users who has rated both the services St and Sj, $|Ut|$ and $|Uj|$ are the number of users who has rated service St and Sj, separately. While the number of co-rated services is small, for example, the value $2\times|Ut\cap Uj|/|Ut|+|Uj|$ will reduce the rating similarity estimation between these two users. Given that the value of $2\times|Ut\cap Uj|/|Ut|+|Uj|$ is among the interval of [0,1] and the value of R_sim(St,Sj) is in the interval of [-1,1] and also the value of R_sim'(St,Sj) is also in the interval of [-1,1].

V. EXPERIMENTAL ENVIRONMENTS

1) Deployment of Clustering Stage:

Step 1.1: Stem Words:

Generally, a mashup service si is described with some tags and functionalize with some APIs [35]. As an experimental case, seven concrete mashup services (i.e.,1,s2,s3,s4,s5,s6 and s7) the corresponding tags and APIs are listed in TABLE I. APIs of si are put into Fi, tags of si are put into Di. Tags in Di are stemmed using Porter stemmer and put into Di'.

Step 1.2: Compute Description Similarity and Functionality Similarity

Description similarities between mashup services are computed using formula (1). For instance, there are one same stemmed tag (i.e., "book") among the six different stemmed tags in $D2$ and $D5$, therefore, $D\_sim(s_2, s_5) = \frac{|D_2' \cap D_5'|}{|D_2' \cup D_5'|} = \frac{1}{6}$

Functionality similarities between mashup services are computed using formula (2). Since there is only one API (i.e., "Amazon Product Advertising") in $F2$ and $F5$,$F\_sim(s_2, s_5) = \frac{|F_2' \cap F_5'|}{|F_2' \cup F_5'|} = 1$

TABLE I: CASE OF MAHSUP SERVICES

CASE OF MAHSUP SERVICES No.	Name	APIs (Fi)	Tags (Di)	Stemmed Tags (Di')
s1	4Wheelz RouteMate	Google Maps	driving, google, maps, streetview	drive, google, map, streetview
s2	GuruLib	Amazon Product Advertising	books, library, videos	book, library, video
s3	100 Destinations	Google Maps + Twitter	fun, mapping, photo, social, travel	fun, map, photo, social, travel
s4	Anuncios Total	Google Maps + Twitter	ads, deadpool, shopping	ads, deadpool, shop
s5	22books	Amazon Product Advertising	books, lists, shopping, social	book, list, shop, social
s6	Favmvs	Google Search + MTV	deadpool, MTV, music, video	deadpool, MTV, music, video
s7	FlickrCash	Flickr	photos, shopping	photo, shop

Step 1.3: Compute Characteristic Similarity

Characteristic similarity is the weight sum of the description similarity and functionality similarity, which is computed using formula (3). Without loss of generality, the weight of description similarity α is set to 0.5. Then the characteristic similarity between s2 and s5 is computed as C_sims2, s5 =$\alpha\times D_sim$s2,5 + 1−α ×F_sims2,s5 =0.5×16+0.5×1≅0.583. It should be noted that all the computation results retain 3 digits after the decimal point, thereafter. Characteristic similarities between the seven mashup services are all computed by the same way, and the results are shown in TABLE II.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

TABLE II: CHARACTERISTIC SIMILARITY MATRIX (KEEPING THREE DECIMAL PLACES)

	$s1$	$s2$	$s3$	$s4$	$s5$	$s6$	$s7$
$s1$	/	0	0.063	0	0	0	0
$s2$	0	/	0	0	0.583	0.083	0
$s3$	0.063	0	/	0	0.063	0	0.083
$s4$	0	0	0	/	0.083	0.083	0.125
$s5$	0	0.583	0.063	0.083	/	0	0.1
$s6$	0	0.083	0	0.083	0	/	0
$s7$	0	0	0.083	0.125	0.1	0	/

Step 1.4: Cluster Services

In this step, Algorithm 1 is processed in the specified order. Initially, the seven services $s1$~$s7$ are put into seven clusters $C1$~$C7$ one by one and the characteristic similarities between each pair of services in TABLE II are assigned to similarity of the corresponding clusters. The highlighted data in TABLE III is the maximum similarity in the similarity matrix.

TABLE III: INITIAL SIMILARITY MATRIX (k=7)

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$
$C1$	/	0	0.063	0	0	0	0
$C2$	0	/	0	0	**0.583**	0.083	0
$C3$	0.63	0	/	0	0.063	0	0.083
$C4$	0	0	0	/	0.083	0.083	0.125
$C5$	0	**0.583**	0.063	0.083	/	0	0.100
$C6$	0	0.083	0	0.083	0	/	0
$C7$	0	0	0.083	0.125	0.1	0	/

The reduction step of Algorithm 1 is described as follows.

Step1. Search for the pair in the similarity matrix with the maximum similarity and merge them.

Step2. Create a new similarity matrix where similarities between clusters are calculated by their average value.

Step3. Save the similarities and cluster partitions for later visualization.

Step4. Proceed with 1 until the matrix is of size K, which means that only K clusters remains.

Let K=3 as the termination condition of Algorithm 1, the reduction steps are illustrated in TABLE IV~TABLE VII. As for reduction Step 1 as shown in TABLE IV, since the maximum similarity in the similarity matrix is $d(C2,C5,C2)$ and $C5$ are merged into ($C2$, $C5$). And the similarity between ($C2$, $C5$) and other clusters are calculated by their average value. For example, $(C2,5),C3=(dC2,C3+dC5,C3)/2=(0+0.063)/2\cong0.032$.

As for reduction Step 2 as shown in TABLE VII, since the maximum similarity in the similarity matrix is $dC3$, $C4$, $C3$ and $C4$ are merged into ($C3$, $C4$). And the similarity between ($C3$, $C4$) and other clusters is calculated by their average value. For example, d ($C2$, $C5$), ($C3$, $C4$) = (($C2$, $C5$), $C3$+d ($C2$,5) ,$C4$)/ 2= (0.032+0.042)/2=0.037.

As for reduction Step 3 as shown in TABLE VIII, since the maximum similarity in the similarity matrix is $dC1$,($C3$,$C4$),$C1$ and ($C3$,$C4$) are merged into ($C1$,$C3$,$C4$) . And the similarity between ($C1$,3,$C4$) and other clusters is calculated by their average value. For example, $d(C1,C3,C4),C6=(d(C1,C6)+d(C3,C4),C6))/2=$ (0+0.042)/2=0.021.

As for reduction Step 4 as shown in TABLE IX, since the maximum similarity in the similarity matrix is $d(C1,C3,C4$),$C7$,($C1$,$C3$,$C4$) and $C7$ are merged into ($C1$,$C3$,$C4$,$C7$). And the similarity between ($C1$,3,$C4$,$C7$) and other clusters is calculated by their average value. For example,

$d(C1,C3,C4,C7),(C2,C5)=d(C1,C3,C4),(C2,C5)+d(C7,C2),C5)/2=$(0.019 +0.050)2≅0.035.

Now, there are only 3 clusters remaining and the algorithm is terminated.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

TABLE IV: ALGORITHM 1: REDUCTION STEP 1 (*k*=6)

	$C1$	(C_2,C_5)	(C_3,C_4)	$C6$	$C7$
$C1$	/	0	0.282	0	0
(C_2,C_5)	0	/	0.037	0.042	0.050
(C_3,C_4)	0.282	0.037	/	0.042	0.104
$C6$	0	0.042	0.042	/	0
$C7$	0	0.050	0.104	0	/

TABLE V: ALGORITHM 1: REDUCTION STEP 2 (*k*=5)

	$C1$	(C_2,C_5)	(C_3,C_4)	$C6$	$C7$
$C1$	/	0	0.282	0	0
(C_2,C_5)	0	/	0.037	0.042	0.050
(C_3,C_4)	0.282	0.037	/	0.042	0.104
$C6$	0	0.042	0.042	/	0
$C7$	0	0.050	0.104	0	/

TABLE VI: ALGORITHM 1: REDUCTION STEP 3 (*k*=4)

	(C_1,C_3,C_4)	(C_2,C_5)	$C6$	$C7$
(C_1,C_3,C_4)	/	0.019	0.021	0.052
(C_2,C_5)	0.019	/	0.042	0.050
$C6$	0.021	0.042	/	0
$C7$	0.052	0.050	0	/

TABLE VII: ALGORITHM 1: REDUCTION STEP 4 (*k*=3)

	(C_1,C_3,C_4,C_7)	(C_2,C_5)	$C6$
(C_1,C_3,C_4,C_7)	/	0.035	0.011
C_2,C_5	0.035	/	0.042
$C6$	0.011	0.042	/

By using Algorithm 1, the seven mashup services are merged into three clusters, where $s2$ and $s5$ are merged into a cluster named $C1$, $s1$, $s3$,4 $ands7$ are merged into a cluster named $C2$, and $s6$ is separately merged into a cluster named $C3$.

2) Deployment of Collaborative Filtering Stage

Step 2.1: Compute Rating Similarity

Suppose there are four users (i.e., $u1$,2,$u3$,$u4$) who rated the seven mashup services. A rating matrix is established as TABLE VIII. The ratings are on 5-point scales and 0 means the user did not rate the mashup. As $u3$ does not rate $s4$ (a not-yet-experienced item), $u3$ is regarded as an active user and $s4$ is looked as a target mashup. By computing the predicted rating of $s4$, it can be determined whether $s4$ is a recommendable service for $u3$. Furthermore, $s1$ is also chosen as another target mashup. Through comparing the predicted rating and real rating of $s1$, the accuracy of ClubRS will be verified in such case.

Since $s4$ and $s1$ are both belong to the cluster $C2$, rating similarity and enhanced rating similarity are computed between mashup services within $C2$ by using formula (4) and (5). The rating similarities and enhanced rating similarities between $s4$ and every other mashup service in $C2$ are listed in TABLE IX while such two kinds of similarities between $s1$ and every other mashup service in $C2$ are listed in TABLE X.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com

TABLE VIII: RATING MATRIX

Mashup service pair	Rating similarity	Enhanced rating similarity
($s1$,$s3$)	0.839	0.839
($s1$,$s4$)	0.544	0.467
($s1$,$s7$)	-0.187	-0.187

TABLE IX: RATING SIMILARITIES AND ENHANCED RATING SIMILARITIES WITH S4

Mashup service pair	Rating similarity	Enhanced rating similarity
($s4$,$s1$)	0.544	0.467
($s4$,$s3$)	0.736	0.631
($s4$,$s7$)	0	0

TABLE X: RATING SIMILARITIES AND ENHANCED RATING SIMILARITIES WITH S1

	C1		*C2*				*C3*
	$s2$	$s5$	$s1$	$s3$	$s4$	$s7$	$s6$
$u1$	5	4	4	3	3	1	0
$u2$	1	1	4	5	4	4	2
$u3$	4	4	1	2	0	2	3
$u4$	5	4	5	5	5	1	5

Step 2.2: Compute Predicted Rating

According to the predicted rating of $s4$ for $u3$, i.e., $Pu3,4=1.97$ and the predicted rating of $s1$ for $u3$, i.e., $Pu3,s1=1.06$. Thus, $s4$ is not a good mashup service for $u3$ and will not be recommended to $u3$. In addition, as the real rating of $s1$ given by user $u3$ is 1 (see TABLE X) while its predicted rating is 1.06, it can be inferred that ClubRS may gain an accurate prediction.

VI. EXPERIMENTAL EVALUATION AND RESULT

To evaluate the accuracy of ClubRS, Mean Absolute Error (MAE), this is a measure of the deviation of recommendations from their true user-specified ratings. As Herlocker et al. [14] proposed, MAE is computed as follow:

$$MAE = \frac{\sum_{i=1}^{n} |r_{a,z} - P(u_a, s_t)|}{n} \qquad (7)$$

In this formula, n is the number of rating-prediction pairs, ra, is the rating that an active user ua gives to a mashup service st, (ua,t) denotes the predicted rating of st for ua.

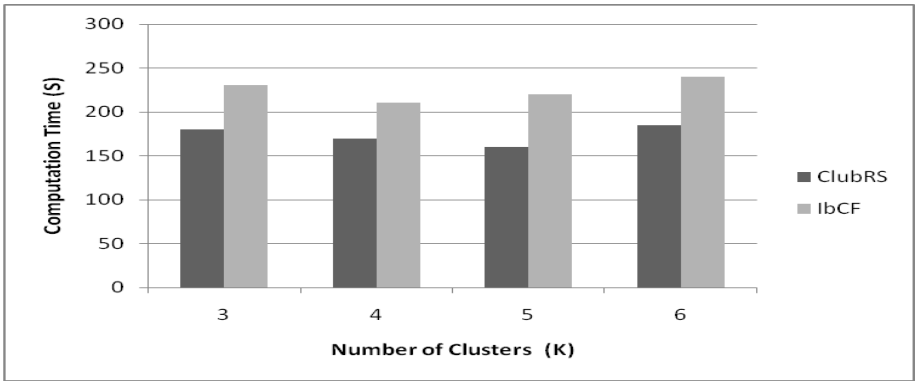
To evaluate the efficiency of ClubRS, the online computation time of ClubRS is compared with that of IbCF[1]. There are several discoveries as follows.

- In all, ClubRS 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.
- As the rating similarity threshold γ increase, the computation time of ClubRS decrease. It is due to the number of neighbors of the target service decreases when γ increase.
- As *K* increase, the computation time of ClubRS decrease obviously. Since a bigger *K* means fewer services in each cluster and a bigger γ makes lessneighbours, the computation time of predicted ratings based on less neighbours may decrease.

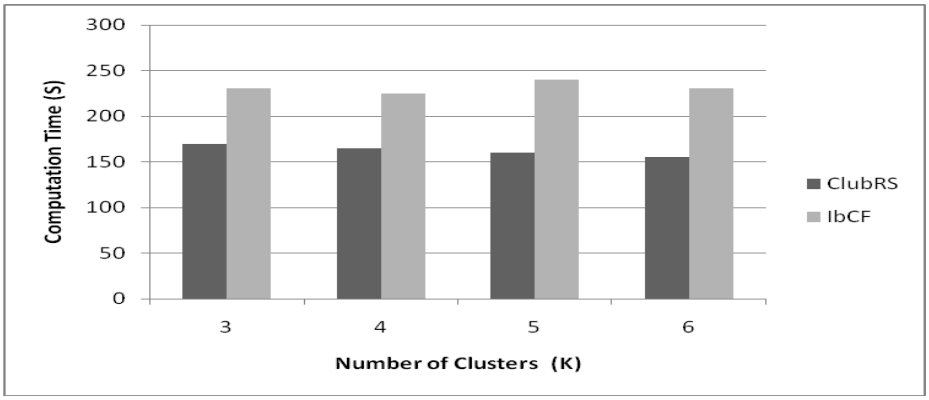
According to the computation complex analysis, it can draw a conclusion that ClubRS may gain good scalability via increase the parameter *K* appropriately. Along with adjustment of γ, recommendation precision is also improved.

A good encryption scheme should resist all kinds of known attacks, such as known plain text attack, cipher-text attack, statistical attack, differential attack, and various brute-force attacks.

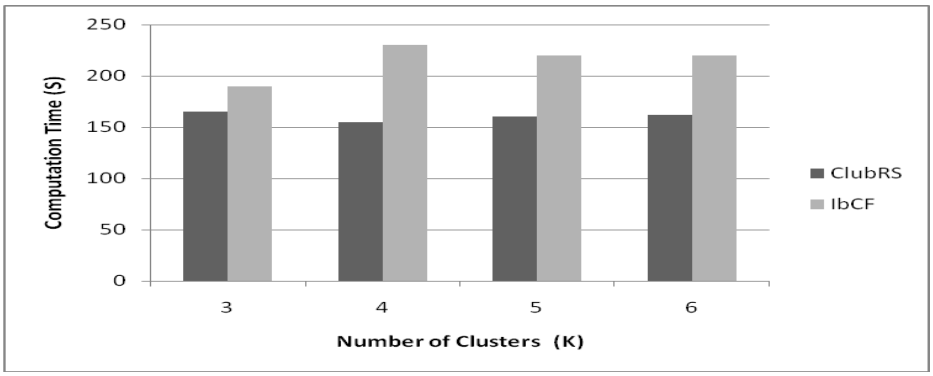
ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 4, pp: (284-296), Month: October - December 2015, Available at: www.researchpublish.com



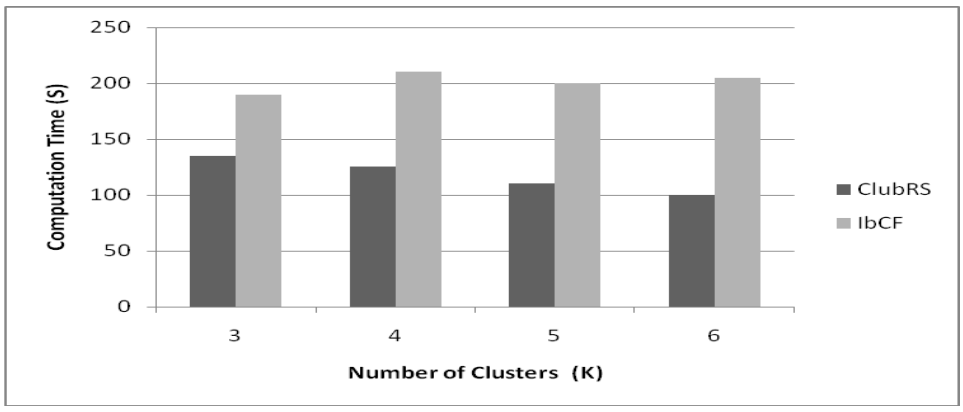
(a)γ=0.1



(b) γ=0.2



(c) γ=0.3

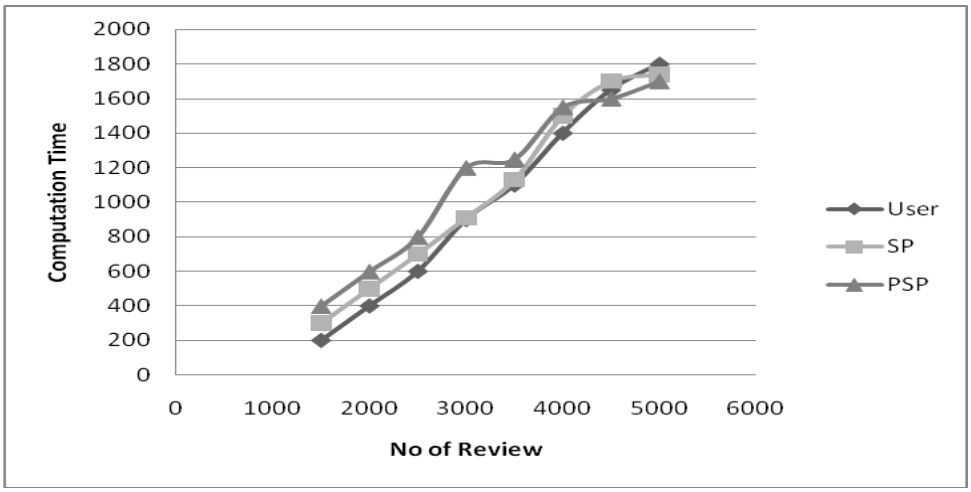


(d) γ=0.4

Fig. 2: Comparison of Computation Time with ClubRS and IbCF

The Homomorphic encryption is encryption on the already encrypted data rather than original data. It works on plain text. Complex mathematical operation is done on cipher text. Multiple Homomorphic algorithms designing the protocol that is used, provide the security of user's personal data. In the existing system, heavy computational and communication overload occurs. To remove this drawback multiple homomorphicalgorithm is used.

In the proposed system, user selects the list of mashup service and then algorithm is used to encrypt the data of user. Later, service provider computes similarities between services within the cluster to generate the recommendation. Then privacy service provider and service provider interact with protocol for generating the encrypted recommendation. This ensuresthat the personal information of user is hidden from other service providers enabling the data to be secured.



This is the Computation Time Graph which Shows the Computation time for three modules we are using for generating recommendation. The Computation time depends upon the no of reviews taken for generating the recommendation or the dataset size we used.

VII. CONCLUSION AND FUTURE WORK

In this paper, we present a ClubRS approach for big data applications related to service recommendation. The services are clustered using Agglomerative Hierarchical (AHC) algorithm. These services are clustered by their similarities. These service clusters take much lesser computation time than the whole system. The rating of these services is more relevant than services from other clusters. Prediction will be more accurate based on this similar cluster of services. These are the two advantages of ClubRS approach.

In our work, we aim to build a system that will generate recommendation privately using homomorphic Cryptography and we extend our work by designing new privacy preserving technique for recommendation generation by considering dynamic behaviour.

Semantic analysis can be applied on the description text of service. So, more semantic-similar services may be clustered mutually, which will increase the number of recommendations for a selected service.

In future attribute based encryption could be done where many users' attributes will be considered for recommendation generation.

REFERENCES

[1] Rong Hu, Wanchun Dou*,Jianxun Liu, Member, "ClubCF: A Clustering-based Collaborative Filtering Approach for Big Data Application," IEEE Translations and content mining, 2013.

[2] M. A. Beyer and D. Laney, "The importance of big data: A definition," Gartner, Tech. Rep., 2012.

[3] X. Wu, X. Zhu, G. Q. Wu, et al., "Data mining with big data," *IEEE Trans. on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97-107, January 2014.

[4] A. Rajaraman and J. D. Ullman, "Mining of massive datasets," *Cambridge University Press*, 2012.

[5] Z. Zheng, J. Zhu, M. R. Lyu. "Service-generated Big Data and Big Data-as-a-Service: An Overview," in *Proc. IEEE BigData*, pp. 403-410, October 2013.

[6] A. Bellogín, I. Cantador, F. Díez, et al., "An empirical comparison of social, collaborative filtering, and hybrid recommenders," *ACM Trans. on Intelligent Systems and Technology*, vol. 4, no. 1, pp. 1-37, January 2013.

[7] W. Zeng, M. S. Shang, Q. M. Zhang, et al., "Can Dissimilar Users Contribute to Accuracy and Diversity of Personalized Recommendation?," *International Journal of Modern Physics C*, vol. 21, no. 10, pp. 1217-1227, June 2010.

[8] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, "Fuzzy c-Means Algorithms for Very Large Data," *IEEE Trans. on Fuzzy Systems*, vol. 20, no. 6, pp. 1130-1146, December 2012.

[9] J. Mai, Y. Fan, and Y. Shen, "A Neural Networks-Based Clustering Collaborative Filtering Algorithm in E-Commerce Recommendation System," in *Proc. 2009 Int'l Conf. on Web Information Systems and Mining*, pp. 616-619, June 2009.

[10] N. Mittal, R. Nayak, M. C. Govil, et al., "Recommender System Framework using Clustering and Collaborative Filtering," in *Proc. 3rd Int'l Conf. on Emerging Trends in Engineering and Technology*, pp. 555-558, November 2010.

[11] X.Li, and T. Murata "Using Multidimensional Clustering Based Collaborative Filtering Approach Improving Recommendation Diversity," in *Proc. 2012 IEEE/WIC/ACM Int'l Joint Conf. on Web Intelligence and Intelligent Agent Technology*, pp. 169-174, December 2012.

[12] M. C. Pham, Y. Cao, R. Klamma, et al., "A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis," *Journal of Universal Computer Science*, vol. 17, no. 4, pp. 583-604, April 2011.

[13] R. D. Simon, X. Tengke, and W. Shengrui, "Combining collaborative filtering and clustering for implicit recommender system," in *Proc. 2013 IEEE 27th Int'l Conf. on. Advanced Information Networking and Applications*, pp. 748-755, March 2013.

[14] J. L. Herlocker, J. A. Konstan, L. G. Terveen, et al. "Evaluating collaborative filtering recommender systems," *ACM Trans. on Information Systems*, vol. 22, no. 1, pp. 5-53, January 2004.

[15] ZekeriyaErkin, ThijsVeugen, Tomas Toft, Reginald L. Lagendijk, "Generating Private Recommendations Efficiently using Homomorphic Encryption and Data Packing", IEEE Transactions on Information Forensics and Security, Vol 7, No. 3, June 2012.

[16] Casino, F. Domingo-Ferrer, J. ;Patsakis, C. ; Puig, D. ; Solanas, A. , "Privacy Preserving Collaborative Filtering with k-Anonymity through Microaggregation", e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference on 11-13 Sept.

[17] Erkin, M. Beye, T. Veugen, and R. L. Lagendijk, "Privacy enhanced recommendation system," in Proc. ThirtyFirstSymp. Information Theory in the Benelux, Rotterdam, 2010, pp. 35–42.

[18] F.McSherry and I. Mironov, "Differentially private recommendation systems: Building privacy into the net," in Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD'09), New York,NY, 2009, pp. 627–636, ACM.