ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (255-264), Month: July - September 2015, Available at: www.researchpublish.com

A Personal Recommendation Searcher Using Keywords

[1]Chiranjeevi T N, [2]Vishwanath R H

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

***Abstract*: Service recommender systems have been shown as valuable tools for providing appropriate recommendations to users. In the last decade, the amount of customers, services and online information has grown rapidly, yielding the big data analysis problem for service recommender systems. Consequently, traditional service recommender systems often suffer from scalability and inefficiency problems when processing or analysing such large-scale data. Moreover, most of existing service recommender systems present the same ratings and rankings of services to different users without considering diverse users' preferences, and therefore fails to meet users personalized requirements. In this paper, we propose a Keyword-Aware Service Recommendation method, named KASR, to address the above challenges. It aims at presenting a personalized service recommendation list and recommending the most appropriate services to the users effectively. Specifically, keywords are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations.**

Keywords: **Recommender system, preference, keyword, big data, MapReduce.**

I. INTRODUCTION

In recent years, the amount of data in our world has been increasing explosively, and analyzing large data sets so called "Big Data" becomes a key basis of competition underpinning new waves of productivity growth, innovation, and consumer surplus [1]. Big data refers to data sets whose size is beyond the ability of current technology, method and theory to capture, manage, and process the data within a tolerable elapsed time. Today, Big Data management stands out as a challenge for IT companies. The solution to such a challenge is shifting increasingly from providing hardware to provisioning more manageable software solutions [2]. Big data also brings new opportunities and critical challenges to industry and academia [3], [4].

Similar to most big data applications, the big data tendency also poses heavy impacts on service recommender systems. With the growing number of alternative services, effectively recommending services that users preferred has become an important research issue Service recommender systems have been shown as valuable tools to help users deal with services Overload and provide appropriate recommendations to them. Examples of such practical applications include CDs, books, web pages and various other products now use recommender systems [5], [6], [7]. Over the last decade, there has been much research done both in industry and academia on developing new approaches for service recommender systems [8], [9].

Service recommender systems have been shown as valuable tools for providing appropriate recommendations to users. In the last decade, the amount of customers, services and online information has grown rapidly, yielding the big data analysis problem for service recommender systems. Consequently, traditional service recommender systems often suffer from scalability and inefficiency problems when processing or analysing such large-scale data. Moreover, most of existing service recommender systems present the same ratings and rankings of services to different users without considering diverse users preferences, and therefore fails to meet users' personalized requirements.

Despite the fact that the above arrangements have gotten some positive results, they are a long way from accomplishing the expense efficient huge information preparing due to the accompanying shortcomings. In the first place, information territory may bring about a misuse of assets. Case in point, most reckoning asset of a server with less prevalent information may stay unmoving. The low asset utility further causes more servers to be initiated and consequently higher working cost.

In this paper, we propose a Keyword-Aware Service Recommendation method, named KASR, to address the above challenges. It aims at presenting a personalized service recommendation list and recommending the most appropriate services to the users effectively. Specifically, keywords are used to indicate users' preferences, and a user-based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. To improve its scalability and efficiency in big data environment. Finally, extensive experiments are conducted on real-world data sets, and results demonstrate that KASR significantly improves the accuracy and scalability of service recommender systems over existing approaches.

With the success of the Web technology, more and more companies capture large-scale information about their customers, providers, and operations. The rapid growth of the number of customers, services and other online information yields service recommender systems in "Big Data" environment, which poses critical challenges for service recommender systems. Moreover, in most existing service recommender systems, such as hotel reservation systems and restaurant guides, the ratings of services and the service recommendation lists presented to users are the same. They have not considered users different preferences, without meeting users personalized requirements.Recommender systems developed as an independent research area in the mid-1990s when recommendation problems started focusing on rating models [10], [11]. According to the definition of recommender system in [12], recommender system can be defined as system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful services in a large space of possible options. Current recommendation methods usually can be classified into three main categories: content-based, collaborative, and hybrid recommendation approaches [13]. Content-based approaches recommend services similar to those the user preferred in the past. Collaborative filtering approaches recommend services to the user that users with similar tastes preferred in the past. Hybrid approaches combine content-based and CF methods in several different ways.

CF algorithm is a classic personalized recommendation algorithm, which is widely used in many commercial recommender systems [13]. In CF based systems, users receive recommendations based on people who have similar tastes and preferences, which can be further classified into item-based CF and user-based CF. In item-based systems, the predicted rating depends on the ratings of other similar items by the same user. While in user-based systems, the prediction of the rating of an item for a user depends upon the ratings of the same item rated by similar users. And in this work, we will take advantage of a user based CF algorithm to deal with our problem.

II. RELATED WORK

There have been many recommender systems developed in both academia and industry. In [33], the authors propose a Bayesian-inference-based recommendation system for online social networks. They show that the proposed Bayesian-inference-based recommendation is better than the existing trust-based recommendations and is comparable to Collaborative Filtering recommendation. In [13], Adomavicius and Tuzhilin give an overview of the field of recommender systems and describe the current generation of recommendation methods. They also describe various limitations of current service recommendation methods, and discuss possible extensions that can improve recommendation capabilities and make recommender systems applicable to an even broader range of applications. Most existing service recommender systems are only based on a single numerical rating to represent a service's utility as a whole [34]. In fact, evaluating a service through multiple criteria and taking into account of user feedback can help to make more effective recommendations for the users.

With the development of cloud computing software tools such as Apache Hadoop, Map-Reduce, and Mahout, it becomes possible to design and implement scalable recommender systems in "Big Data" environment. The authors of [35] implement a CF algorithm on Hadoop. They solve the scalability problem by dividing data set. But their method doesn't have favourable scalability and efficiency if the amount of data grows. [36] presents a parallel user profiling approach based on folksonomy information and implements a scalable recommender system by using Map-Reduce and Cascading

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (255-264), Month: July - September 2015, Available at: www.researchpublish.com

techniques. Jin et al. [37] propose a large-scale video recommendation system based on an item-based CF algorithm. They implement their proposed approach in Qizmt, which is a .Net Map-Reduce framework, thus their system can work for large-scale video sites. in the information position strategy to support vitality productivity in server farms and propose a booking calculation.

Generally speaking, comparing with existing methods, KASR utilizes reviews of previous users to get both of user preferences and the quality of multiple criteria of candidate services, which makes recommendations more accurate. Moreover, KASR on MapReduce has favourable scalability and efficiency.

III. PROPOSED SYSTEM

In this paper, we propose a keyword-aware service recommendation method, named KASR. In this method, keywords are used to indicate both of users preferences and the quality of candidate services. A user-based CF algorithm is adopted to generate appropriate recommendations. KASR aims at calculating a personalized rating of each candidate service for a user, and then presenting a personalized service recommendation list and recommending the most appropriate services to him/her.

IV. SYSTEM MODEL

The main steps of KASR are depicted in Fig. 1, which are described in detail as follows:

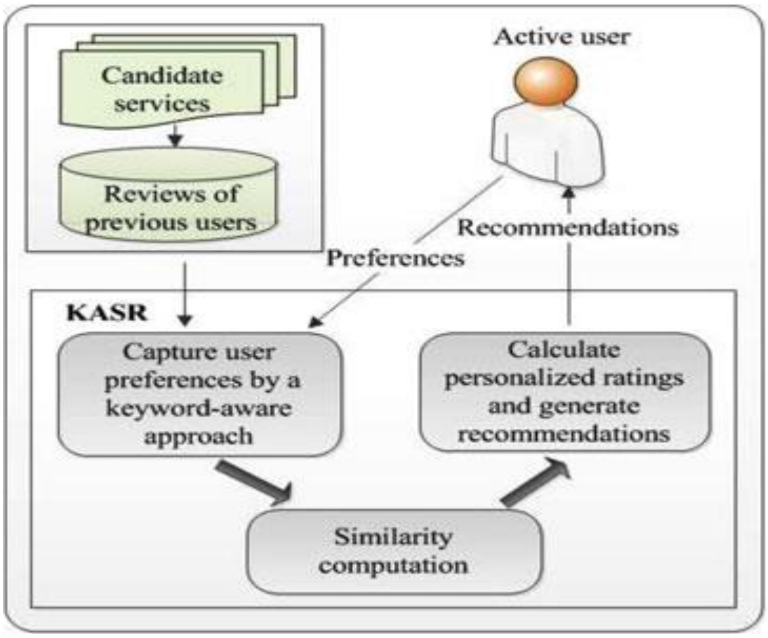

Fig 1: KASR'S MAIN STEPS

(1***) Capture user preferences by a keyword-aware approach*****:** In this step, the preferences of active users and previous users are formalized into their corresponding preference keyword sets respectively. In this paper, an active user refers to a current user needs recommendation.

(2) ***Similarity computation*****:** The second step is to identify the reviews of previous users who have similar tastes to an active user by finding neighbourhoods of the active user based on the similarity of their preferences. Before similarity computation, the reviews unrelated to the active user's preferences will be filtered out by the intersection concept in set theory. If the intersection of the preference keyword sets of the active user and a previous user is an empty set, then the preference keyword set of the previous user will be filtered out.

Two similarity computation methods are introduced in our recommendation method: an approximate similarity computation method and an exact similarity computation method. The approximate similarity computation method is for the case that the weights of the keywords in the preference keyword set are unavailable, while the exact similarity computation method is for the case that the weight of the keywords are available.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (255-264), Month: July - September 2015, Available at: www.researchpublish.com

(a) Approximate similarity computation. A frequently used method for comparing the similarity and diversity of sample sets, Jaccard coefficient, is applied in the approximate similarity computation.

Jaccard coefficient is measurement of asymmetric information on binary (and non-binary) variables, and it is useful when negative values give no information. The similarity between the preferences of the active user and a previous user based on Jaccard coefficient is described as follows:

$$sim(APK,\ PPK) = Jaccard(APK,\ PPK) = \frac{|APK \cap PPK|}{|APK \cap PPK|},$$

Where APK is the preference keyword set of the active user, PPK is the preference keyword set of a previous user. And the weight of the keywords is not considered in this approach.

Algorithm 1, SIM-ASC, illustrates the functionality of the approximate similarity computation method.

Algorithm 1: SIM-ASC (Approximate Similarity Computation)

Input: The preference keyword set of the active user APK

The preference keyword set of a previous user PPK_j

Output: The similarity of APK and PPK_j, $sim_{ASC}(APK,PPK_j)$

1: $sim_{ASC}(APK,PPK_j)=\dfrac{|APK \cap PPK_j|}{|APK \cup PPK_j|}$

2: return the similarity of APK and PPK_j, $sim_{ASC}(APK,UPK_j)$

(b) Exact similarity computation. A cosine-based approach is applied in the exact similarity computation, which is similar to the vector space model (VSM) in information retrieval [24], [25].

Then the similarity based on the cosine-based approach is defined as follows:

$$sim(APK, PPK) = \cos(\vec{W}_{AP}, \vec{W}_{PP}) = \frac{\vec{W}_{AP} \bullet \vec{W}_{PP}}{\|\vec{W}_{AP}\|_2 \times \|\vec{W}_{PP}\|_2} = \frac{\sum_{i=1}^{n} \vec{W}_{AP,i} \times \vec{W}_{PP,i}}{\sqrt{\sum_{i=1}^{n} \vec{W}_{AP,i}^2}\sqrt{\sum_{i=1}^{n} \vec{W}_{PP,i}^2}},$$

(3) Calculate personalized ratings and generate recommendations. Based on the similarity of the active user and previous users, further filtering will be conducted. Given a threshold ɗ, if sim(APK,PPK_j)< ɗ, the preference keyword set of a previous user PPKj will be filtered out, otherwise PPKj will be retained. The thresholds given in two similarity computation methods are different, which are both empirical values.

Once the set of most similar users are found, the personalized ratings of each candidate service for the active user can be calculated. Finally, a personalized service recommendation list will be presented to the user and the service(s) with the highest rating(s) will be recommended to him/her.

Algorithm 2, SIM-ESC, illustrates the functionality of the exact similarity computation method.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (255-264), Month: July - September 2015, Available at: www.researchpublish.com

Algorithm 2: SIM-ESC (Exact Similarity Computation)

Input: The preference keyword set of the active user APK
The preference keyword set of a previous user PPK_j
Output: The similarity of APK and PPK_j, $sim_{ESC}(APK, PPK_j)$
1: for each keyword k_i in the keyword-candidate list
2: if $k_i \in APK$ **then**
3: get $\vec{W}_{AP,i}$ by formula (2)
4: else $W_{AP,i} = 0$
5: end if
6: if $k_i \in PPK_j$ **then**
7: get $\vec{W}_{PP_j,i}$ by formula (5)
8: else $W_{PP_j,i} = 0$
9: end if
10: end for
11: get $sim_{ESC}(APK, UPK_j)$ by formula (6)
12: return the similarity of APK and PPK_j, $sim_{ESC}(APK, UPK_j)$

Algorithm 3 illustrates the basic algorithm of KASR. The input contains the preference keyword set of the active user APK, the candidate services WS = {ws_1,ws_2,....,ws_N}, the threshold ɗ in the filtering phase, and the number K. In line 2, R is used to store the remaining preference keyword sets of previous users, and sum is to record the number of the remaining preference keyword sets of previous users. Line 3 to line 8 is used to process each review of the previous users into the corresponding preference keyword sets, and then do a simple filtering to filter out the reviews unrelated with the active user's preferences. Line 9 to line 15 are to calculate the similarity of APK and PPKj, and then filter out the keyword set PPKj whose similarity with APK is less than the threshold ɗ. In this paper, there are two methods to calculate the similarity: approximate similarity computation (see Algorithm 1) and exact similarity computation (see Algorithm 2). Line 16 to line 18 is to calculate the personalized ratings of the candidate services for the active user. Finally, line 19 and line 20 is to sort the candidate services according to the personalized ratings and recommend the services with the Top-K highest ratings to the active user.

For convenience, KASR with the approximate and exact similarity computation methods are denoted as KASR-ASC and KASR-ESC, respectively. Suppose there are N candidate services and each service with R reviews on average. Moreover, suppose that there are n keywords in the keyword- candidate list. Then, the time complexity of KASR-ASC and KASR-ESC are O (NR) and O(NRn), respectively.

Algorithm 3: Basic Algorithm of KASR

Input: The preference keyword set of the active user APK
The candidate services $WS=\{ws_1, ws_2, \ldots, ws_N\}$
The threshold δ in the filtering phase
The number K
Output: The services with the Top-K highest ratings $\{tws_1, tws_2, \ldots, tws_K\}$
1: for each service $ws_i \in WS$
2: $\hat{R} = \Phi, sum = 0, r = 0$
3: for each review R_j of service ws_i
4: process the review into a preference keyword set PPK_j
5: if $PPK_j \cap APK \neq \Phi$ **then**
6: insert PPK_j into $\hat{R}$
7: end if
8: end for
9: for each keyword set $PPK_j \in \hat{R}$
10: $sim(APK, PPK_j) = SIM(APK, PPK_j)$
//$SIM(APK, PPK_j)$ can be $SIM\text{-}ASC(APK, PPK_j)$ or $SIM\text{-}ESC(APK, PPK_j)$
11: if $sim(APK, PPK_j) < \delta$ **then**
12: remove PPK_j from $\hat{R}$
13: else $sum=sum+1$, $r=r+r_j$
14: end if
15: end for
16: $\bar{r} = r / sum$
17: get pr_i by formula (7)
18: end for
19: sort the services according to the personalized ratings pr_i
20: return the services with the Top-K highest ratings $\{tws_1, tws_2, \ldots, tws_K\}$

V. EXPERIMENTAL EVALUATION AND RESULT

In this section, experiments are designed and analysed to evaluate the accuracy and scalability of KASR. To evaluate the performance of KASR in accuracy, we compare KASR with other two well-known recommendation methods: user-based algorithm using Pearson Correlation Coefficient (PCC) and item-based algorithm using PCC, which are called as UPCC [13] and IPCC [28] respectively. Three metrics are used to evaluate the accuracy: mean absolute error (MAE) [29], mean average precision (MAP) [30] and discounted cumulative gain (DCG) [31]. As to the scalability, a well-accepted scalability metric, Speedup [32], is adopted to measure the performance in the scalability of KASR.

Two groups of experiments are conducted to evaluate the accuracy and scalability of KASR. In the first one, we compare KASR with UPCC and IPCC in MAE, MAP and DCG to evaluate the accuracy of KASR. The other is to explore the scalability of KASR.

5.1 Accuracy Evaluation:

(1) Comparison of UPCC, IPCC, KASR-ASC and KASR-ESC in MAE. MAE is a statistical accuracy metric often used in CF methods to measure the prediction quality. And the normalized mean absolute error (NMAE) metric is also used to measure the prediction accuracy. The lower the MAE or NMAE presents the more accurate predictions.

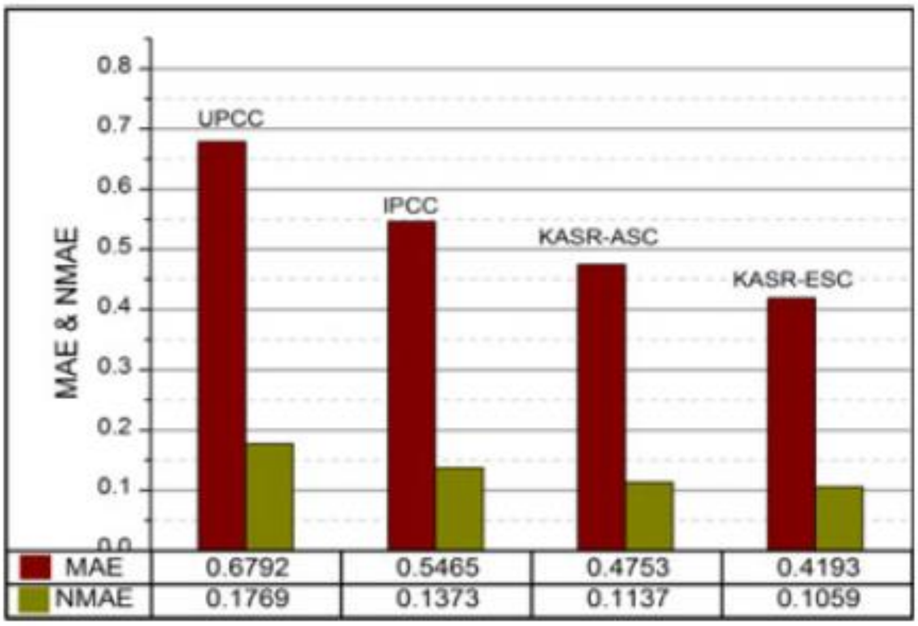

Fig 2: Comparison of UPCC, IPCC, KASR-ASC and KASR-ESC in MAE

Fig. 2 shows the MAE and NMAE values of UPCC, IPCC, KASR-ASC and KASR-ESC. It could be found that the MAE and NMAE values of KASR-ASC and KASR ESC are much lower than UPCC and IPCC (e.g., the MAE and NMAE values of KASR-ASC are respectively 30.02% ((0.6792–0.4753) / 0.6792 ¼ 30.02%) and 35.73% lower than UPCC. And the MAE and NMAE values of KASR-ESC are respectively 23.28 and 22.87 percent lower than IPCC). Thus our methods KASR-ASC and KASRESC can provide more accurate predictions than traditional methods UPCC and IPCC.

2) Comparison of UPCC, IPCC, KASR-ASC and KASR-ESC in MAP and DCG. In most service recommender systems, users tend to be recommended the top services of the returned result list. The services in higher position, especially the first position, should be more satisfying than the services in lower position of the returned result list. To evaluate the quality of Top-K service recommendation list, MAP and DCG are used as performance evaluation metrics. And the higher MAP or DCG presents the higher quality of the predicted service recommendation list. More tasks and their corresponding data chunks can be placed in the same data center, or even in the same server. Further increasing the number of servers will not affect the distributions of tasks or data chunks any more.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (255-264), Month: July - September 2015, Available at: www.researchpublish.com

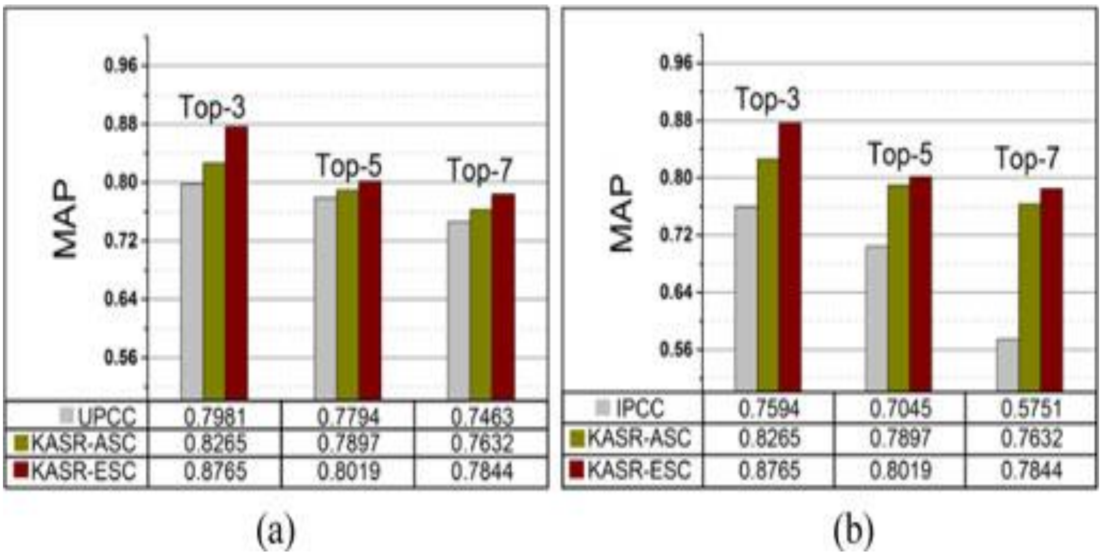

Fig 3: Comparison of UPCC, IPCC, KASR-ASC and KASR-ESC in the MAP values of Top-K (K ¼ 3, 5, 7) recommendation list. (a) Shows the comparison of UPCC, KASR-ASC and KASR-ESC in MAP. (b) Shows the comparison of IPCC, KASR-ASC and KASR-ESC in MAP

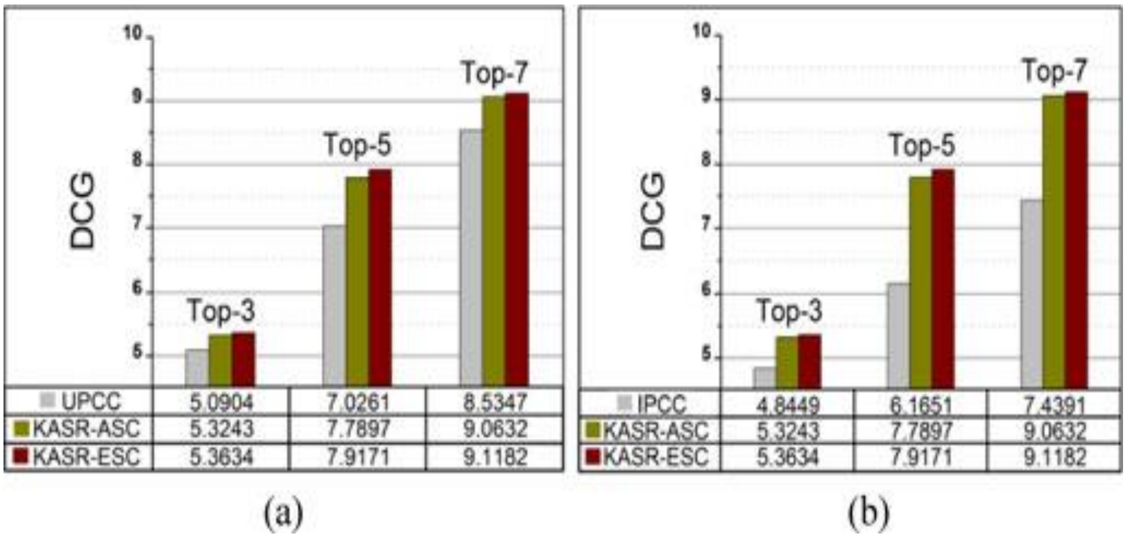

Fig 4: Comparison of UPCC, IPCC, KASR-ASC and KASR-ESC in the DCG values of Top-K (K ¼ 3, 5, 7) recommendation list. (a) Shows the comparison of UPCC, KASR-ASC and KASR-ESC in DCG. (b) Shows the comparison of IPCC, KASR-ASC and KASR-ESC in DCG.

Figs. 3 and 4 respectively show the MAP values and DCG values of Top-K (K ¼ 3, 5, 7) recommendation list of UPCC, IPCC, KASR-ASC and KASR-ESC. From Figs. 6 and 7, we can see that the MAP values and DCG values of KASR-ASC and KASR-ESC are comparatively higher than UPCC and IPCC. It also could be found that the MAP values decrease when K increases, while the DCG values increase when K increases.

5.2 Scalability Evaluation:

A well-accepted scalability metric, Speedup [33], is adopted to measure the performance in the scalability of KASR. Speedup refers to how much a parallel algorithm is faster than a corresponding sequential algorithm, which can be defined as follows:

$$S_P = T_1/T_P$$

Where p is the number of processors, T1 is the sequential execution time, Tp is the parallel execution time with p processors. If the speedup has a linear relation with the numbers of nodes with the data size fixed, the algorithm will have good scalability.

ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 3, pp: (255-264), Month: July - September 2015, Available at: www.researchpublish.com

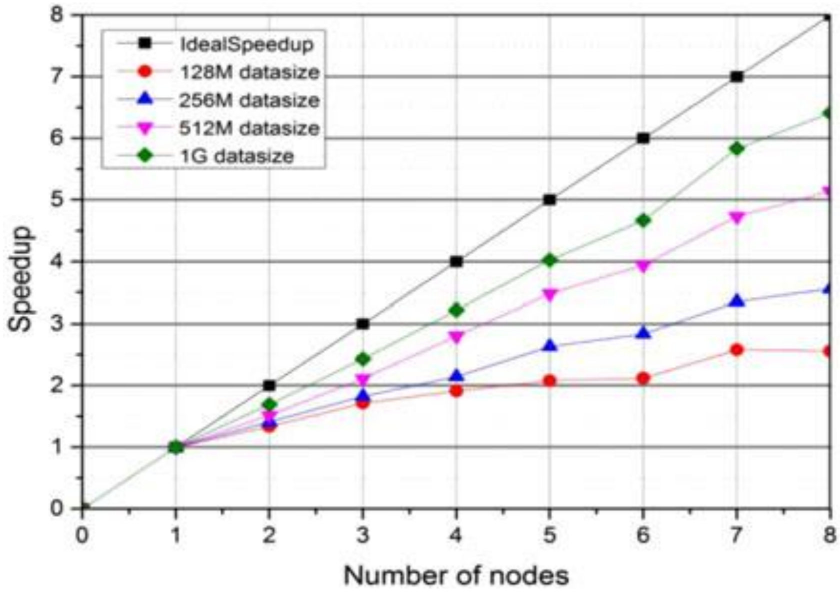

Fig 5: Speedup of KASR

To verify the scalability of KASR, experiment is conducted respectively in a cluster of nodes ranging from 1 to 8. There are four synthetic data sets used in the experiments (128M, 256M, 512M and 1 G data size). Fig. 3 shows the speedup of KASR (Here, KASR-ESC method is adopted in the scalability experiment). From Fig. 5, we can see that the speedup of KASR increases relative linearly with the growth of the number of nodes. Meanwhile, larger data set obtained a better speedup. When the data size is 1 G and the number of nodes is 8, the speedup value reaches 6.412, which is 80.15 percent (6.412/8 ¼ 80.15%) of the ideal speedup.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a keyword-aware service recommendation method, named KASR. In KASR, keywords are used to indicate users' preferences, and a user based Collaborative Filtering algorithm is adopted to generate appropriate recommendations. More specifically, a keyword- candidate list and domain thesaurus are provided to help obtain users preferences. The active user gives his/her preferences by selecting the keywords from the keyword candidate list, and the preferences of the previous users can be extracted from their reviews for services according to the keyword-candidate list and domain thesaurus. Our method aims at presenting a personalized service recommendation list and recommending the most appropriate service(s) to the users.

In our future work, we will do further research in how to deal with the case where term appears in different categories of a domain thesaurus from context and how to distinguish the positive and negative preferences of the users from their reviews to make the predictions more accurate.

REFERENCES

[1] J. Manyika et al., "Big Data: The Next Frontier for Innovation, Competition, and Productivity," 2011.

[2] C. Lynch, "Big Data: How Do Your Data Grow?" Nature, vol. 455, no. 7209, pp. 28-29, 2008.

[3] F. Chang, J. Dean, S. Ghemawat, and W.C. Hsieh, "Bigtable: A Distributed Storage System for Structured Data," ACM Trans. Computer Systems, vol. 26, no. 2, article 4, 2008.

[4] W. Dou, X. Zhang, J. Liu, and J. Chen, "HireSome-II: Towards Privacy-Aware Cross-Cloud Service Composition for Big Data Applications," IEEE Trans. Parallel and Distributed Systems, 2013.

[5] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, Jan. 2003.

[6] M. Bjelica, "Towards TV Recommender System Experiments with User Modeling," IEEE Trans. Consumer Electronics, vol. 56, no. 3, pp. 1763-1769, Aug. 2010.

[7] M. Alduan, F. Alvarez, J. Menendez, and O. Baez, "Recommender System for Sport Videos Based on User Audiovisual Consumption," IEEE Trans. Multimedia, vol. 14, no. 6, pp. 1546-1557, Dec. 2012.

[8] Y. Chen, A. Cheng, and W. Hsu, "Travel Recommendation by Mining People Attributes and Travel Group Types from Community- Contributed Photos," IEEE Trans. Multimedia, vol. 25, no. 6, pp. 1283-1295, Oct. 2013.

[9] Z. Zheng, X. Wu, Y. Zhang, M. Lyu, and J. Wang, "QoS Ranking Prediction for Cloud Services," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1213-1222, June 2013.

[10] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and Evaluating Choices in a Virtual Community of Use," Proc. SIGCHI Conf. Human Factors in Computing System (CHI '95), pp. 194-201, 1995.

[11] P. Resnick, N. Iakovou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," Proc. ACM Conf. Computer Supported Cooperative Work (CSCW '94), pp. 175-186, 1994.

[12] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," User Modeling and User-Adapted Interaction, vol. 12, no. 4, pp. 331-370, 2002.

[13] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the Stateof-the-Art and Possible Extensions," IEEE Trans. Knowledge and Data Eng., vol. 17, no. 6, pp. 734-749, June 2005.

[14] D. Agrawal, S. Das, and A. El Abbadi, "Big Data and Cloud Computing: New Wine or Just New Bottles?" Proc. VLDB Endowment, vol. 3, no. 1, pp. 1647-1648, 2010.

[15] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Comm. ACM, vol. 51, no. 1, pp. 107-113, 2005.

[16] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazons Highly Available Key-Value Store," In Proc. 21st ACM Symp. Operating Systems Principles, pp. 205-220, 2007.

[17] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly, "Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks," Proc. European Conf. Computer Systems, pp. 59-72, 2007.

[18] S. Ghemawat, H. Gobioff, and S. T. Leung, "The Google File System," Proc. 19th ACM Symp. Operating Systems Principles, pp. 29- 43, 2003.

[19] L. Zhang, "Editorial: Big Services Era: Global Trends of Cloud Computing and Big Data," IEEE Trans. Services Computing, vol. 5, no. 4, pp. 467-468, Fourth Quarter, 2012.

[20] Z. Luo, Y. Li, and J. Yin, "Location: A Feature for Service Selection in the Era of Big Data," Proc. IEEE 20th Int'l Conf. Web Service, pp. 515-522, 2013.

[21] H. Schutze and J. O. Pedersen, "A Cooccurrence-Based Thesaurus and Two Applications to Information Retrieval," Information Processing & Management, vol. 33, no. 3, pp. 307-318, 1997.

[22] Y. Jing and W. Croft, "An Association Thesaurus for Information Retrieval," Proc. Intelligent Multimedia Retrieval Systems and Management Conf. (RIAO), vol. 94, pp. 146-160, 1994.

[23] B. Issac and W.J. Jap, "Implementing Spam Detection Using Bayesian and Porter Stemmer Keyword Stripping Approaches," Proc. IEEE Region 10 Conf. (TENCON '09), pp. 1-5, 2009.

[24] P. Castells, M. Fernandez, and D. Vallet, "An Adaptation of the Vector-Space Model for Ontology-Based Information Retrieval," IEEE Trans. Knowledge and Data Eng., vol. 19, no.2, pp. 261-272, Feb. 2007.

[25] Y. Zhu and Y. Hu, "Enhancing Search Performance on Gnutellalike P2P Systems," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 12, pp. 1482-1495, Dec. 2006.

[26] Ff A. Chu, R. Kalaba, and K. Spingarn, "A Comparison of two Methods for Determining the Weights of Belonging to Fuzzy Sets," J. Optimization Theory and Applications, vol. 27, no. 4, pp. 531-538, 1979.

[27] G. Salton, Automatic Text Processing. Addison-Wesley, 1989.

[28] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-Based Collaborative Filtering Recommendation Algorithms,” Proc. 10th Int’l Conf. World Wide Web, pp. 285-295, 2001.

[29] K. Lakiotaki, N.F. Matsatsinis, and A. Tsoukis, “Multi-Criteria User Modeling in Recommender Systems,” IEEE Intelligent Systems, vol. 26, no. 2, pp. 64-76, Mar./Apr. 2011.

[30] Y. Pan and L. Lee, “Performance Analysis for Lattice-Based Speech Indexing Approaches Using Words and Subword Units,” IEEE Trans. Audio, Speech, and Language Processing, vol. 18, no. 6, pp. 1562-1574, Aug. 2010.

[31] G. Kang, J. Liu, M. Tang, X. Liu, and B. cao, “AWSR: Active Web Service Recommendation Based on Usage History,” Proc. IEEE 19th Int’l Conf. Web Services (ICWS), pp. 186-193, 2012.

[32] G.M. Amdahl, “Validity of the Single-Processor Approach to Achieving Large Scale Computing Capabilities,” Proc. Spring Joint Computer Conf., pp. 483-485, 1967.

[33] X. Yang, Y. Guo, and Y. Liu, “Bayesian-Inference Based Recommendation in Online Social Networks,” IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 4, pp. 642-651, Apr. 2013.

[34] G. Adomavicius and Y. Kwon, “New Recommendation Techniques for Multi criteria Rating Systems,” IEEE Intelligent Systems, vol. 22, no. 3, pp. 48-55, May/June 2007.

[35] Z.D. Zhao and M.S. Shang, “User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop,” Proc. Third Int’l Workshop Knowledge Discovery and Data Mining, pp. 478-481, 2010.

[36] H. Liang, J. Hogan, and Y. Xu, “Parallel User Profiling Based on Folksonomy for Large Scaled Recommender Systems: An Implementation of Cascading MapReduce,” Proc. IEEE Int’l Conf. Data Mining Workshops, pp. 156-161, 2010.

[37] Y. Jin, M. Hu, H. Singh, D. Rule, M. Berlyant, and Z. Xie, “MySpace Video Recommendation with Map-Reduce on Qizmt,” Proc. IEEE Fourth Int’l Conf. Semantic Computing, pp. 126-133, 2010.