IJCSEC- International Journal of Computer Science and Engineering Communications.
Vol.3, Issue 2, 2015, Page.866-869, ISSN: 2347–8586

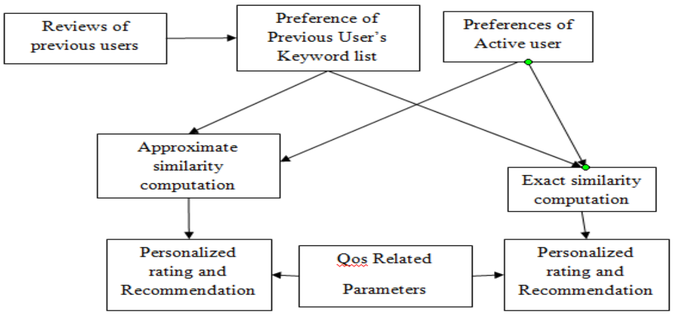
Qos and Keyword-Aware Service Recommendation Method on Map Reduce for Big Data Applications

T Sasikala[1] S Jaya Sundar[2]
[1]PG Student, [2]HOD/IT
[1&2]Dept. of Computer Science and Engineering
Idhaya Engineering College for Women
susikala28@gmail.com

Abstract: 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 became an important research issue. Service recommender systems have been exposed as valuable tools to help users deal with services overload and provide appropriate recommendations to them. 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. The proposed system proposes methods it aims at presenting a personalized service recommendation list and recommending the most appropriate service(s) to the users. To improve the scalability and efficiency of KASR in "Big Data" environment, the proposed system proposes techniques that have been implemented it on a Map Reduce framework in Hadoop platform. It improves the recommendation accuracy by considering the location of the user while recommend the service.
Index Terms: recommender system, preference, keyword, Big Data, MapReduce, Hadoop

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. Handling of large scale data, to provide recommendation, consequently, traditional service recommender systems often suffer from scalability and inefficiency problems when processing or analyzing 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. The proposed system provides technique to personalize recommended services. There have been many recommender systems developed in both academia and industry. The authors propose a Bayesian-inference-based recommendation system for on-line 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 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 discussed 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. 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.

Architecture diagram of proposed system



Proposed System
Techniques
Capture user preferences by a keyword-aware approach
1.Preference of an active user:

An active user can give his/her preferences about candidate services by selecting keywords from a keyword-candidate list, which reflect the quality criteria of the services he/she is concerned about. The preference keyword set of the active user can be de- noted as, where is the *i*th keyword selected from the keyword-candidate list by the active user.

2.Preference of an previous User:

The preferences of a previous user for a candidate service are extracted from his/her reviews for the service according to the keyword-candidate list and domain thesaurus. And a review of the previous user will be formalized into the preference key-word set of him/her, which can be denoted as, where is the *i*th keyword extracted from the review, *h* is the number of extracted keywords.

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 PPKj
Output: The similarity of APK and PPKj, $\text{sim}_{ASC}(APK,PPK_J)$

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 *PPKj*
Output: The similarity of *APK* and *PPKj*, $\text{sim}_{ESC}(APK,PPK_J)$

Algorithm 3: Basic Algorithm of KASR
Input: The preference keyword set of the active user *APK* The candidate services *WS*={*ws1*, *ws2*, …, *wsN*} The threshold in δ in the filtering phase The number *K*
Output: The services with the Top-K highest ratings {t*ws1*, t*ws2*, …, t*wsK*}

Similarity computation
1.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**.** And the weight of the keywords is not considered in this approach Similarity value calculated using the Jaccard coefficient technique.

Sim(APK,PPK)= Jaccard(APK, PPK)= $\left|\frac{APK \cap PPK}{APK \cup PPK}\right|$

2.Exact Similarity computation:

The weight vector of the preference keyword set of a previous user can be decided by the term frequency/inverse document frequency (*TF-IDF*) measure, which is one of the best-known measures for specifying the weight of keywords in Information Retrieval. In the *TF-IDF* approach, to calculate the preference weight vector of a previous user u', "all reviews" by user u' should be collected. Here, "all reviews" contain the reviews by user u' for the candidate services and similar services not in the candidate services. The reviews should also be transformed into keyword sets respectively according to the keyword-candidate list and the domain thesaurus. Weight of the keyword considered in the exact similarity calculation.

$$Wi=\frac{1}{m}\sum_{j=1}^{m}\frac{ajj}{\sum_{k=1}^{m}akj}$$

Generation of Recommendation and calculation of personalized rating:

Based on the similarity of the active user and previous users, further filtering will be conducted. Given a threshold, if, the preference keyword set of a previous user *PPKj* will be filtered out, otherwise *PPKj* will be retained. The thresholds given in two similarity computa-tion 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.

Qos Aware Recommendation and rating

System that provides the recommended services with the quality of services. This proposed system contributes one method called, QoS aware recommendation system, in this, Qos and keyword aware recommendation system, location of the similar user will be considered as one of the parameters to recommend the services. It will increase recommendation accuracy of the system. Because a same location or region user prefers or likes same services to access and use It may improve accuracy of the recommendation system**.**

KASR on Map Reduce

To improve the scalability and efficiency of KASR in "Big Data" environment, The proposed system implements it in a Map Reduce framework on Hadoop platform. The whole computation will be performed in the KASR-ASC and KASR-ESC on Map Reduce are respectively.

KASR Using Approximate similarity computation, First step is to process the reviews for candidate services by previous users into their preference keyword sets and compute the average ratings for each candidate service. Using Map Reduce. The second step is to compute the similarity between the active user and previous user Using Map Reduce. Third step aims to calculate the personalized rating of each candidate service and present a personalized recommendation list to the active user Using Map Reduce.KASR Using Exact similarity computation performed the service are recommended for the active user using MapReduce to handle the large data parallel.

Related work

Xiwang Yang[6] used a Recommendation is playing an increasingly important role in our life. Accurate recommendations enable users quickly locate desirable items without being overwhelmed by irrelevant information. It does not achieve scalability. **Zheng[8]** CLOUD computing is Internet-based computing, whereby shared configurable resources (e.g., infrastructure, platform, and software) are provided to computers and other devices as services. It does not support scalability**. Zhao[7]**Collaborative Filtering (CF) algorithm is a widely used personalized recommendation technique in commercial recommendation systems, and many works have been down in this field to improve the performance. It does not support scalability. **Yan-Ying Chen[2]**Leveraging community-contributed data (e.g., blogs, GPS logs, and geo-tagged photos) for personalized recommendation is one of the active research problems since there are rich contexts and human activities in such explosively growing data. It does not concentrate on scalability.

Conclusion

The proposed system proposes 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. The proposed system method aims at presenting a personalized service recommendation list and recommending the most appropriate service(s) to the users. Moreover, to improve the scalability and efficiency of KASR in “Big Data” environment, the proposed system has implemented it on a Map Reduce framework in Hadoop platform. It also considers the QoS parameters of the Active user recommendation service.

Future Work:

In future, the proposed system enhances the recommendation accuracy by consider another parameters as to improve the recommendation system performance.

Reference:

[1] Adomavicius.G and Kwon.Y (2007), “New Recommendation Techniques for Multi criteria Rating Systems,” IEEE Intelligent Systems, Vol. 22, No. 3, pp. 48-55.

[2] Chen.Y et al (2012),“Travel Recommendation by Mining People Attributes and Travel Group Types From Community-Contributed Photos”. IEEE Transactions on Multimedia, Vol. 25, No.6, pp. 1283-1295.

[3] Dean.J and Ghemawat.S (2005),“MapReduce: Simplified data processing on large clusters,” Communications of the ACM, Vol. 51, No.1, pp. 107-113.

[4] Sarwar.B et al (2001), “Item-based collaborative filtering recommendation algorithms,” Proceedings of the 10th international conference on World Wide Web, pp. 285-295

[5] Shunmei Meng et al (2013),**“**KASR: A Keyword-Aware Service Recommendation Method on Map Reduce for Big Data Applications,” IEEE Transactions on parallel and distributed systems, TPDS.12-1141.

[6] Yang.X and Guo.Y (2013), “Bayesian-inference based recommendation in online social networks,” IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 4, pp. 642-651.

[7] Zhao.Z.D and Shang.M.S (2010), "User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop," In the third International Workshop on Knowledge Discovery and Data Mining, pp. 478-481.

[8] Zheng.Z et al (2013), “QoS Ranking Prediction for Cloud Services,” IEEE Transactions on Parallel and Distributed Systems, Vol. 24, No. 6, pp. 1213-1222.