International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 02, Issue – 02, February – 2017, PP – 45-48

RATING SYSTEM REGULARIZED WITH UNINSTALLATION COUNT PROCEDURE

K.Rajamani, ME, (PHD),
Assistant Professor,
New Prince Shri Bhavani College of
Engineering and Technology,

R. Ajitha, L. Backiyalakshmi,
New Prince Shri Bhavani College of
Engineering and Technology,

Abstract: A rating of the mobile application should depend on the number of installation of the application. The TrustSVD integrates multiple information sources into the recommendation model to reduce cold start problems and data sparsity, but it does not recommend top-N applications. The uninstallation procedure fully avoids these problems and increase the recommendation performance of applications. It maintains the real application user's count. If an installed application is deleted by the user then the count of installation should be deduced by one. The backend algorithm maintain the number of real installed user through the internet. If a user enter in the market, the application used by the user should be updated in the market for rating. Then the number of downloads and uninstallation count will displayed on the app description with the user ratings. Thus the real value of the applications will be obtained to the user.

1. INTRODUCTION

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouse. Data mining tools predict future trends and behaviours, allowing businesses to make proactive, knowledge driven decisions. Data mining tools can answer business questions that traditionally were too time consuming to resolve.

Recommender Systems have been widely used to provide users with high quality personalized recommendations from a large volume of choices. Robust and accurate recommendations are important in e-commerce operations (e.g., navigating product offerings, personalization, improving customer satisfaction), and in marketing (e.g. tailored advertising, segmentation, cross-selling).

Collaborative filtering (CF) is one of the most popular techniques to implement a recommender system. The idea of CF is that users with similar preferences in the past are likely to favour the same items in the future. CF has also been applied to tasks besides item recommendations in domains such as image processing and bioinformatics. However, CF suffers from two well-known issues: data sparsity and cold start. The former issue refers to the fact that users usually rate only a small portion of items while the latter indicates that new users only give a few ratings. Both issues severely degrade the efficiency of a recommender system in, modeling user preferences and thus the accuracy of predicting a user's rating for an unknown item.

In the uninstallation count procedure, the data sparsity and cold start problems are fully reduced and it gives accurate results for recommending the performance of mobile applications. Now a days, mobile application plays vital role in day to day life. In the mobile application market many apps are developed for same process. Users download a application based on their reviews but the review may be false and it degrades the performance of recommendation system.so the false rating can be avoided by the uninstallation count procedure. It increases the overall recommendation system performance.

2. RELATED WORK

Trust SVD model builds on top of SVD++ through which both the explicit and implicit influence of user-item ratings are involved to generate predictions. It recommend an item based on user trust and ratings. In addition, it consider the influence of trust users (including trustees and trusters) on the rating prediction for an active user.

It conduct an empirical trust analysis and observe that trust and ratings, these observations helps to solve recommendations problems. However, the literature has shown that models for rating prediction cannot suit the task of top-N recommendation.
[1] G. Guo, J. Zhang, and D. Thalmann, "A simple but effective method to incorporate trusted neighbors in recommender systems, "in proceedings of the 20th International Conference on User Modeling, Adaptation and Personalization (UMAP), 2012, pp.144-125.
Guo et al. complement a user's rating profile by merging those of trusted users through which better recommendations can be generated, and the cold start and data sparsity problems can be better handled. However, memory based approaches have difficulty in adopting to large scale.

[2] Optimizing User Experience in Choosing Android Applications
We present a recommendation system aimed at helping users and developers alike. We help users to choose optimal sets of applications belonging to different categories (e. g. browsers, e-mail, cameras) while minimizing energy consumption, transmitted data, and maximizing application rating. We also help developers by showing the relative placement of their application's efficiency with respect to selected others. When the optimal set of applications is computed, it is leveraged to position a given application with respect to the optimal, median and worst application in its category.

[3] UI Ripping in Android: Reverse Engineering of Graphical User Interfaces and its Application
Android applications are user interaction intensive programs that makes UI become an indispensable part of mobile applications. UI also reflect information, such as the functions of applications, that makes the study of android UI very significant. We design a UI modeling method based on attribute graph by the technology of reverse engineering and program analysis for application on Android. The proposed method is also applied to malware repackage detection and evaluation of the application family resemblance.

[4] User review matter Tracking Crowdsourced reviews to Support Evolution of Successful Applications
Nowadays software applications, and especially mobile apps, undergo frequent releases updates through app stores. After installing/updating apps, user can post reviews and provide ratings, expressing their level of satisfaction with apps, and possibly pointing out bugs or desired features. In this paper we show by performing a study on 100 Android apps-how developers addressing user reviews increase their app's success in terms of rating. Specifically, we devise an approach, named cristal, for tracing informative reviews onto source code changes, and for monitoring the extent to which developers accommodate crowd request and follow-up user as reflected in their rating.

[5] Detecting Clones in Android Application through Analyzing User Interfaces
The blooming mobile smart phone device industry has attracted a large number of application developers. However, due to the availability of reverse engineering tools for Android applications, it also caught the attention of plagiarist and malware writers. In recent years, application cloning has become a serious threat to the Android market. In previous work, mobile application clone detection mainly focuses on code-based analysis. Such an approaches lacks resilient to advance **obfuscation techniques.**

3. PROPOSED SYSTEM:

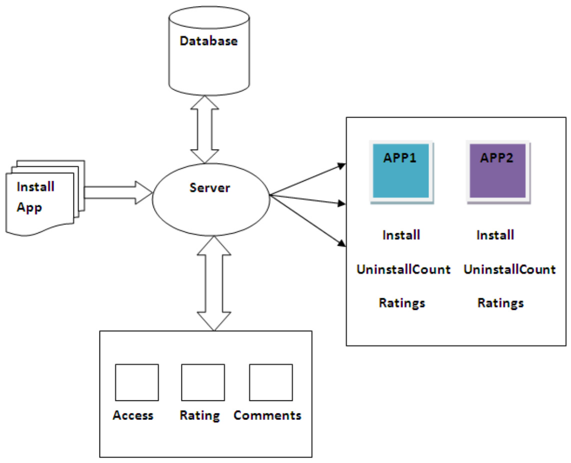
In now a days, the applications are rated by number of downloads, the application with maximum downloads are considered as a successful application. The review and ratings will be updated by the user input. By this malicious ratings cannot be find, thus the user cannot get a real application feature.

The rating of the application should be based on the number of users who are really using the application. If a user download application it will add to download count and if the user uninstall the application, it will be shown in addition to the number of downloads, so the real rating of the user can be produced and the abnormal rating for app will be analysed.

The uninstallation count procedure will maintain a server to update the real usage count of the applications. The server will show both the installation and uninstallation count, thus the user will know the real usage of the applications. This procedure also helps to top-N applications and to scale up large systems.

This approach sort the real applications easily and filter's the false application. Application max level is examined and no need for analysis of the application by the user. User feedback decides the success of application.

ARCHITECTURE DIAGRAM



APP CREATION

- The application for various function are collected.
- The app will be unique.
- The app function will be analyzed for further purpose.
- It will maintain the count and ratings.

MARKET FORMATION

- The application will collect and download link will be generated.
- The download count will be maintained in the database for the rating calculation.
- The previous for user comments will be provided.
- The average rating for the app will be displayed.

APP RATING

- Apprating depends on the number of real user of the application.
- The sudden raise of the rating will be analyzed in the backend of the market.
- False rating of the application will be identified.
- The the average rating for the application will be displayed.

UNINSTALL COUNT

- If the user uninstall an application, the download count will be deduced for the corresponding app.
- Whenever user connect to the market the app will be checked if not found found it will consider as uninstalled.
- The uninstallation count will be displayed on the application description.

4. CONCLUSION

The Uninstall Count Recommendation System avoids Data Sparsity and Cold Start problems and it can be scale up to large number of systems. It ensures the value of the application and reduce the false rating to make the application market with real application and rating. This procedure sorts the application easily and it can recommend top-N applications.

5. REFERENCES

[1]. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, “IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 17, no. pp. 734-749,2005.

[2]. Y. Huang, X. Chen, J. Zhang, D. Zeng, D. Zhang, and X. D-ing, “Single-trial fERPsg denoising via collaborative filtering on fERPsg images,” Neurocomputing, vol. 149, Part B, pp.914-923.

[3]. X. Luo, Z. Ming, Z. You, S. Li, Y. Xia, and H. Leung, “Improving network topology –based protein interactome mapping via collaborative filtering,” Knowledge-Based Systems(KBS), vol. 90,pp.23-32,2015.

[4]. G. Guo, J. Zhang, and D. Thalmann, “A simple but effective method to incorporate trusted neighbors in recommender systems,” in proceedings of the 20th International Conference on User Modeling,Adaptation and Personalization (UMAP), 2012, pp.114-125.

[5]. -----, “Factorization meets the neighborhood: a multifaceted col-laborative filtering model,” in Proceeding of 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD),2008,pp.426-434**.**

[6]. Junnan Chen, Courtney University and Gaby G. Dagher, “product recommendation system for small online retailers using association rules mining,” in Proceeding of the 2014 International Conference on Innovation Design and Manufacturing (ICIDM).

[7]. Suhasini Parvatikar and Bharti Joshi, “Online Book recommendation system by using collaborative filtering and association mining,” in Proceeding of the 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC).