User Preference Based Mobile Application Recommendation

Vipinkatara, Prashant K. Gupta, Madhurima Hooda, Saurabh Gupta

Abstract: Application recommendation is one of the larger scales and sophisticated recommendation system currently exists. In this research work, we devised an approach which will deal with suggesting application based on users click on a particular application. The approach described in this paper is efficient and with less memory requirement than other traditional methods. This paper also includes the details about the implementation of the approach with User Interface. The paper also provides the details that how it can be implemented on a large scale. The approach is implemented in a mobile-based platform with react native support. The main objective of this paper is to describe an approach, which will be efficient and completely based on users data. The main objective of an Application Recommendation to recommend applications to increase user experience and recommend application based on their needs. Companies like Google, Apple, Samsung etc. are implementing it also.

Keywords: Mobile application, User preference, Recommender Systems

I. INTRODUCTION

Mobile devices have permeated daily life of human beings in a seamless manner. These have become like a personal advisor or like a personal assistant and contain greater user information than even a close friend. The reason for this standardchange is mainly due to the capability of these devices to support an ever-increasing number of mobile applications. Mobile applications or the mobile apps, as these are commonly referred, keep the user experience in which. These mobile apps have a number of properties or advantages such that a user uses them even without having any technical information about them.

The most commonly used apps are categorized into messaging like WhatsApp or entertainment like Netflix [22], [23]. A lot of hoursgoes in the designing of these apps so that a user can keep using them in a faster and easier manner, that too for long hours without getting tired. Furthermore, if a user uses an app repeatedly then this can be a sign of his/her affinity for that app. In the nutshell, authors of this paper believe that every user has a different preference for a different type of apps thus whenever a user unlocks his/her mobile device then he/she should be presented with the plethora of installed apps based on the app ranking.

The app ranking is generally employed by the app stores also. For example, the Apple Application Store recommends application based on the user's previous data and the same thing is done by Google Play Store but based on different algorithms. It is important to learn the way these app stores rank the mobile apps using the different ranking algorithm. This knowledge is also necessary because a large amount of research is currently going on for the design of algorithms that optimize mobile app rankings. The major part of the research consists of enhancing the previous method of recommendation engine and developing new and better recommendation engine algorithms [4], [5].

There are number of techniques employed for ranking these mobile apps such as collaborative filtering, content-based recommendation and Hybrid Recommendation.

However, these approaches suffer from the number of limitation. These methods ignore the relationship between data values. Therefore in this research work, we proposed graph-based methodology for app ranking.

Our novel graph-based mobile app ranking method captures the user's preferences and builds a graph by forming a connection between different nodes. Initially, the user preferences are captured by an app numbered 0 and as the user browses through the app, weighted connection are developed from this app to the other apps. Greater the weight of the link greater is the number of times a user browses the app(we will discuss the details in section 3).

The biggest advantage of our proposed novel algorithm is that no time is wasted in learning user preferences and aligning the apps [1-3]. User preferences are learned on the fly as the user continue to browse the apps. We strongly feel that our approach is novel because no one has used the graph-based algorithm for learning the user preferences and ranking the mobile apps till day.

The organization of the rest of the paper is as follows; section 2 discusses the mathematical preliminary that forms the base of the present work, section 3 discusses our proposed user preference aware novel graph-based mobile app ranking algorithm, section 4 discusses the result and finally, section 5 concludes the present work.

II. MATHEMATICAL PRELIMINARIES

In this section, we will discuss the mathematical details of the algorithm used to optimize the memory usage of the program [6-10].

In our previous method, we have added a constant(for example 1) in every case when the user has clicked on 1 and based on that we have ranked the applications.

In our new approach, we have selected a normalization method to prevent from going out of memory. This will prevent more memory usage at the cost of computation.

Revised Manuscript Received on November 22, 2019.

Corresponding author

VipinKatara, Amity University, Noida, India, Email: vipinkatara2@gmail.com

Prashant K. Gupta, Amity University, Noida, India, Email: pkgupta@amity.edu

MadhurimaHooda, Amity University, Noida, India, Email: mhooda@amity.edu

Saurabh Gupta, Dr. Akhilesh Das Gupta Institute of Technology & Management, GGSIP, India, Email: saurabh@adgitmdelhi.ac.in

$$z = \frac{x - min(x)}{(x) - min(x)} \qquad (1)$$

Where min and max are the minimum value in x and maximum value in x given its range. $X$is current value at which value is to be normalize [12-15].

Min-max normalization or equation (1) is often refer as feature scaling where the values of a integer range of a feature of data, i.e. a property, whichnormalize to a scale in range of 0 and 1. Therefore, in order to calculate z, i.e. the normalized value of a member of the observed values of x.

III. DESIGN

To suggest application based on similar kind of users previous choice, an algorithm can be suggested which connect all the applications by link or virtual link and contain score how frequently it had been visited irrespective of the user. One of the approach to do this is by following techniques.

A. Use weight graph [16], [17]

Connect all the application by assuming its one application as initial vertex and other as final vertex and one edge connecting them. a Edge should have initial weight as minimum as possible or it can be 0. To render suggestion of application, all the neighbors of current vertex need to be copied in dictionary with key being end vertex and value being the weight. That dictionary then sorted in decreasing order based on values and then it should be rendered sequentially.

B. Update [18]

It is necessary to update the data based on which the application user has clicked. This is done by adding a constant ‘c' to the value of the key that the user has select and then normalize it with using min max normalized equations.

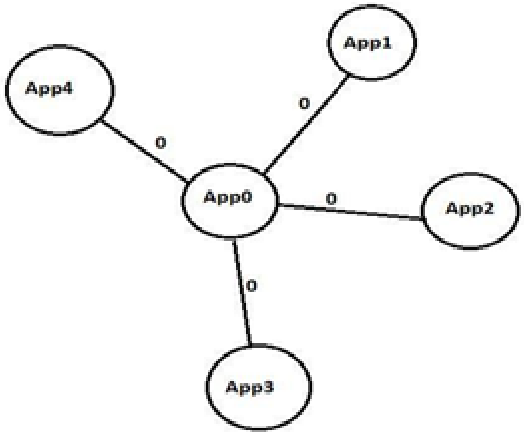

Fig 1 Graph representing the neighbors the app0

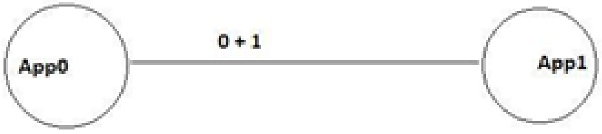

Fig 2 Update when app1 is selected on app0 page

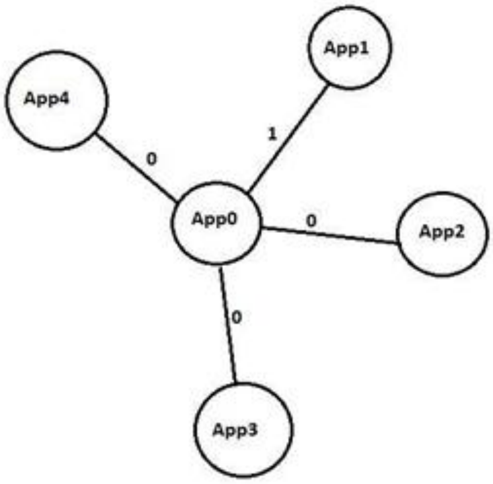

Fig 3 Updated graph showing the neighbors of App0

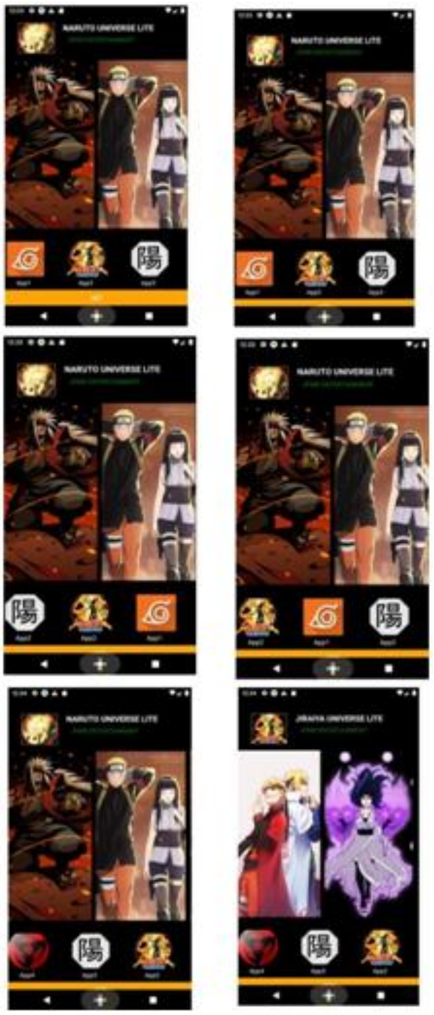

Fig 4 Screenshots of implemented algorithm

C. Adding back to graph[19-21]

It is required to add back all the updates that happened in the dictionary. After adding back graph will contain the values which as per the user's data.

The pseudo code for above implementation is stated below /* this pseudo-code will will increase the rank of the particular applications and display them based on their rank */

```
Rank_Inc(map, tar, start_vertex)
            value := map(tar) //this will extract value
 from map which was created to store the apps
 connected with all apps
           map(tar) := value + 1 // this will increase the
                        value of edge by 1
                      for i[key, val] to map.keys()
                     add back all the updates to original
                           graph normalize the all values
              Traverse(start_vertex, edge, map, app)

                         for i to edge[vertex]
                         map(i, edge[start_vertex][i])

                              sort map key wise
                     Rank_inc(map, app, start_vertex)

                       display the content of edge
```

Implementation of above implementation is done by using JavaScript based framework react native. Screenshot of the implementation are shown in Fig. 4.

IV. RESULTS AND DISCUSSION

This algorithm will suggest application based on user's data with less access time and memory than other classic recommendation algorithms like basket analysis. This algorithm will work best with user-user collaborative filtering in which users have the same kind of interest in the products. Not only this application can be used in application recommendation but in other recommendation also like video recommendation, product recommendation.

This method can also be implemented with item-item collaborative filtering by changing the vertex to applications. This method is most effective with integration in collaborative filtering.

This method consumes less memory and computation power than most of the recommendation model, which makes it more efficient and fast than the others. To overcome the high memory usage to store the frequency of the users preference, we have use normalization technique from data mining i.e. Min max Normalization. Which let the value of the frequency between 0 and 1, which in turn decrease memory usage and prevent the memory overflow.

V. CONCLUSION

Application recommendation is a subpart of the Recommendation system, so, it is very important to suggest applications to users, which will they like. So, as a developer, it is necessary to develop an algorithm with the utmost possible accuracy with less processing power and memory. So that each and every company can easily use it without dealing with cost overhead. We have developed a recommendation algorithm which takes much less power and good accuracy with less difficulty in implementation and with good accuracy if users are of the same mindset or behavior. This algorithm can be implemented as a part of a recommendation engine as a same group of users, which has the same taste in the application. As a part of the recommendation engine algorithm, this algorithm can greatly increase the accuracy of the recommendation engine as a whole.

REFERENCES

[1] http://www.google.com/technology/index.html, Our Search: Google Technology.

[2] A.Arasu, J.Novak, A.Tomkins, and John Tomlin,*PageRank Computation and theStructure of the Web: Experiments and Algorithms*,Technical Report, IBM Almaden Research Center (November 2001).

[3] Pavel Berkhin, A survey on PageRank computing,*Internet Mathematics***2** (2005), no. 1, 73–120.

[4] www.abcseo.com/seo-book/toolbar-google.htm Google Toolbar PageRank.

[5] Fig 2 Update when app1 is selected on app0 pagehttp://www.google.com/corporate/history.html, Google Corporate Information: Google Milestones.

[6] C.Priyal, A.Tiwari, and M.Hooda. "Fraudulent Loan Prediction using Machine Learning Algorithms." Indian Journal of Public Health Research & Development 10.5 (2019): 845-850. Indexed in Scopus.

[7] Srivastava, Ashutosh, Aashie Roy Saxena, Madhurima and SaruDhir. "Creating Convolutional Neural Network and Training it to Classify Images using Machine Learning." Indian Journal of Public Health Research & Development 10.5 (2019): 839-844.

[8] Sapra, Shradha, MadhurimaHooda, Anshul Chhabra, and SaruDhir. "Smart Med-Minder." Indian Journal of Public Health Research & Development 9, no. 11 (2018).

[9] A.Sood, M.Hooda, S.Dhir, S., andM.Bhatia, (2018). An Initiative to Identify Depression using Sentiment Analysis: A Machine Learning Approach. Indian Journal of Science and Technology, 11(4). doi:10.17485/ijst/2018/v11i4/119594.

[10] Aggarwal S., Goswami D., Hooda M., Chakravarty A., Kar A., Vasudha (2020) Recommendation Systems for Interactive Multimedia Entertainment. In: Hemanth J., Bhatia M., Geman O. (eds) Data Visualization and Knowledge Engineering. Lecture Notes on Data Engineering and Communications Technologies, vol 32. Springer, Cham

[11] S.Bhardwaj, S.Dhir and M.Hooda, "Automatic Plant Watering System using IoT". 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), IEEE Xplore: 04 July 2019.

[12] Khandelwal, Harshit, and SaruDhir and MadhurimaHooda. "An Analysis of Interactions Among Barriers on the Implementation of Green Computing: Using Multi-objective Decision Modelling ISM." International Conference on Advanced Informatics for Computing Research. Springer, Singapore, 2018.

[13] Vashishtha, Akash, SaruDhir, and MadhurimaHooda. "Smart Cities in India: Revamping the Street Lighting System Using IOT." International Conference on Advanced Informatics for Computing Research. Springer, Singapore, 2018.

[14] Bhatia M., Pandey M., Kumar N., Hooda M., Akriti (2017) Enacting Segmentation Algorithms for Classifying Fish Species. In: Saini H., Sayal R., Rawat S. (eds) Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol 8. Springer, Singapore.

[15] Kumar A., Bhatia M., Garg A., Madhurima (2017) Anatomization of Software Quality Factors: Measures and Metrics. In: Saini H., Sayal R., Rawat S. (eds) Innovations in Computer Science and Engineering. Lecture Notes in Networks and Systems, vol 8. Springer, Singapore.

[16] Sikka, Chandrika, SaruDhir, and MadhurimaHooda. "A survey on challenges in Software Development during the Adoption of Agile Environment." Innovations in computer science and engineering. Springer, Singapore, 2016. 219-224.

[17] Mishra, S., Dhir, S. and Hooda, M., 2016. A Study on Cyber Security, Its Issues and Cyber Crime Rates in India. In Innovations in Computer Science and Engineering (pp. 249-253). Springer Singapore.

[18] Agarwal S., Bhatia M., Hooda M. (2016) Emotion Recognition: A Step Ahead of Traditional Approaches. In: Satapathy S., Mandal J., Udgata S., Bhateja V. (eds) Information Systems Design and Intelligent Applications. Advances in Intelligent Systems and Computing, vol 434. Springer, New Delhi

[19] Madhurima, Anuj Kumar Chauhan, SaruDhir, Madhulika, 2015, October. Difficulties and challenges faced in testing AJAX applications. In Soft Computing Techniques and Implementations (ICSCTI), 2015 International Conference on (pp. 113-115). IEEE

[20] Nandan, V., Madhurima, Dhir, S. and Garg, A., 2015, October. Statistical analysis of Indian e-commerce market. In Soft Computing Techniques and Implementations (ICSCTI), 2015 International Conference on (pp. 146-151). IEEE.

[21] Rai, P., Hooda, M., Dhir, S., Bhatia, M. and Garg, A., 2015, March. A prologue of JENKINS with comparative scrutiny of various software integration tools. In Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on (pp. 201-205). IEEE.

[22] Saurabh Mishra, MadhurimaHooda, SaruDhir and Alisha Sharma. iCop: A System for Mitigation of Felonious Encroachment Using GCM Push Notification, CSI-2015 50th Golden Jubilee Annual Convention on “Digital Life”, 02nd – 05th December, 2015.

[23] Bhatia, Madhulika, Divakar Yadav, Madhurima, Pritee Gupta, Gurpreet Kaur, Jyoti Singh, Mallika Gandhi, and Ajeet Singh. "Implementing edge detection for medical diagnosis of a bone in Matlab." 5th International Conference on Computational Intelligence and Communication Networks (CICN), 2013, pp. 270-274. IEEE, 2013.

AUTHORS PROFILE

VipinKatarais currently pursuing his B. Tech in the field of Computer Science from Amity School of Science and Technology, Amity University, Uttar Pradesh, India. He is a Machine Learning Enthusiast. His research interest include Artificial Intelligence, Machine Learning, and Deep Learning. He has done his senior and senior secondary from St Mary's convent School and St Clare's Senior Secondary school, Agra, Uttar Pradesh, India in 2015 and 2017 respectively.

Prashant K Gupta (M’14) received the B.Tech and M.Tech degrees from the Guru Gobind Singh Indraprastha University, New Delhi, India in 2008 and 2012, respectively. He has completedhis Ph.D degree in Computer Science from South Asian University, New Delhi, India. His research interests include fuzzy logic, computing with words, linguistic optimization and energy management. Prashant has published in reputed journals/conferences such as IEEE Transactions on Fuzzy Systems, Fuzzy Sets and Systems, Applied Soft Computing, Granular Computing, Fuzz-IEEE and IEEE SMC.

Dr. Madhurima has done her Ph.D in IT, M.Tech in IT and MBA in HRM (from University of South Wales, United Kingdom) and B.E(Information Technology). She has published many research papers in national and international journals and conferences of repute. She has publsihed a book titled "Computer Networks" with laxmi publications and contributed as an editor in global edition of the book "Absolute Java", Pearson Education. Her area of research is video object tracking, software testing, IoT, AJAX Applications and Machine learning. Her M.Tech work has been published as a book titled “Video Object Tracking” by LAP LAMBERT Academic Publishing GmbH & Co. KG, Germany. She has currently three patents and two copyrights on her name.

Saurabh Gupta (M’14) received the B.Tech degree from IET, MJPRU, Bareilly in 2001 and M.Tech degrees in 2005. He has completed his Ph.D degree in Computer Science from MIT, BRABU, Muzaffarpur, India. His research interests include WSN and fuzzy logic. Saurabh has published in reputed journals/conferences such as IEEE on Data Mining, Fuzzy Sets. Saurabh is guiding the Ph.D students and reviewer of IGI Global.