

CAARD - Context Aware App Recommendation and Delivery using Decision Support Systems

V. Praveena * Adithya Raam Sankar, S. Jeya Balaji, R. Sreyas Naaraayanan and Srikrishnan Subramanian

Department of Computer Science and Engineering, SRM University, Ramapuram Campus, Chennai - 603203, Tamil Nadu, India; praveena.v@rmp.srmuniv.ac.in, dithya_raam@srmuniv.edu.in, jeyabalaji_s@srmuniv.edu.in, sreyas_naaraayanan@srmuniv.edu.in, srikrishnan_subramanian@srmuniv.edu.in

Abstract

Objectives: To deliver the affinity of user(s) as a service to prospective developers and help them in providing context aware content. **Methods:** First, the list of apps installed on the mobile device is retrieved using a special module attached to the host application and is then tagged according to its genre. The module then runs a background service to record the active times of each application. All these information is synchronized with the database on the cloud periodically. The CAARD engine analyses the information in the database to predict the affinity of the user(s). **Findings:** Currently the users are classified based on their location and categorized using search history, browsing pattern which are not so efficient. They sometimes aim to deliver contents based on the web searches. This again does not necessarily mirror the requirements of the user as all the search terms may be trivial and need not always be specific to the user. Most if the previous systems aim to modify web based techniques for the mobile ecosystem which makes it less efficient. This directly reflects on the revenue of the developers or results in the fall of user base. The proposed system understands the user based on the applications that he frequently uses. This makes it even more user centric and helps the developers in delivering content that is specific to each of the users. **Applications:** One possible application of this system could be to display more relevant advertisements. This persuades the user to read and click the advertisement thus generating more revenue. Another implementation could be to recommend products that the user might be interested to buy.

Keywords: Context Aware, Decision Support System, Mobile Application, Recommendations

1. Introduction

Present generation systems concentrate more on providing content for dominant media such as TV, radio and the Web. Recent studies have shown that providing context-aware content in apps shares said to reach significant milestone. Mobile App industry is expected to spend \$100 billion by the end of 2016, which is almost 430% increase from 2013. Traditional web system shows recommendation by using contextual advertising¹, providing context-aware content always has been a very big challenge in the App industry.

Giving irrelevant ads and recommendations in mobile devices tend to cause a revenue loss because it doesn't suit the users. They only annoy the users and not

drive them to click the ads and generate revenue. The contextual recommendations used in a browser cannot be implemented in mobile apps as the contents of app can't be crawled as in a web page. In order to overcome this shortcoming, we propose a system that can give recommendations based on user interest.

In this paper, we propose a context-aware recommendation system for the app industry. This system uses the list of apps installed on the user devices as the primary input to the Context-Aware App Recommendation and Delivery (CAARD) engine. The app list, along with active time of each app, is collected through a special component that is attached to the initial data provider's application. Each of the app from the app-list is tagged with a specific genre. The genre tag and the active time of

*Author for correspondence

each app forms the initial data set to CAARD engine. This initial data set is fed into the categorization module. Here the apps are categorized into different genres using the associated tag. Once the list of applications is divided into groups, it is then analyzed using strength computation module. From the calculated strengths we get the interest of specific users which can be utilized to provide context aware content. This decision support system is designed to be auto-didactic in nature²

This decision support system uses a mathematical procedure to calculate the strength. The mathematical model calculates the strength of a cluster of apps for a single user (S_0), which denotes the individual affinity of that user towards that particular genre, and the strength of a cluster of apps for a group of users (S_1) which denotes the affinity of the group of users under consideration.

The CAARD engine has two important components, viz the categorization and the analysis. Initially, the apps are clustered based on the genre. Once the clusters are formed, each of them is analyzed using the basic K-Means algorithm. The Affinity Genre is obtained as an output which forms the base for the self-learning part of the CAARD engine.

The CAARD Engine will attempt to recommend the best interest of a specific group of users, which will benefit the prospective developers to give context-aware content thereby increasing its popularity and revenue.

Web based Decision Support System uses a brute-force mechanism or cookie based learning mechanism³. Brute force mechanism involves past browsing history of a particular user and makes inferences from that. In case of Cookie based learning systems, the history is stored by a particular website for their own future use. This information is used by the learning algorithm to predict user interest patterns.

Common interest of a group of users is an important factor as the mobile devices are dominating the next generation user base. These inferred results are provided as a service to the mobile client devices⁴. Recommendation based on the above two mentioned criteria in most of the cases is found to be extraneous. The recommendation does not particularly reflect the need of the user in more than half of situations. As the previous system is not made for web based applications, the same methodology is adapted to fit for mobile systems. So this reflects in the core working part of the system and creates a depreciation in relevancy of recommended data⁵.

1.1 Drawbacks of Existing System

The existing strategies lead to a lot of shortcomings when they were adapted to the mobile system. Some of them are:

- It uses Brute-force mechanism or Cookie learning mechanism
- Satisfactory results are not achieved.
- Poor recommendations and distribution methodologies.
- It is not specifically tailored for the app ecosystem.
- Derived and mutated from the web and web based services.
- Does not use context aware criteria. Results were generalized.
- Lack of user centric approach leading to inefficient results.

2. Proposed Work

This system proposes a unique method in identifying the prospective users of a particular genre application from a specific group of users. This is a valuable insight for the developers to identify the target audience for a specific application. This system is enhanced by the context and thus the solution obtained promotes a more end user centric usage pattern. The output of this system will be a user specific interest pattern for a specific user group. When this system becomes fully functional, it automatically improves the accuracy of the usage pattern by iterative steps. The dataset for analyzing the usage pattern can be obtained by using a specific functionality in the mobile development platform. By using this function, the list of apps installed on the individual device and the active running time for individual app is uploaded to the CAARD Engine. The rules are obtained by performing an association rule learning algorithm that uses the list of applications as the input and obtains the rules of learning on basis of confidence parameters⁴. This can be further understood using an example. The list of application installed in the phone can be extracted using specific commands targeted at the system applications list. Once this list has been extracted, it is clustered into segments on basis of the genre of the application. This is achieved using the K-Means clustering algorithm. Once this app list has been classified, the strengths of the different categories are found. This is then sent back to the host application

to provide context-aware and user specific content. The strengths are also stored in the database to be used later to infer the affinity of a group of users.

In order to find the affinity of a group of users, the strengths of individual users are grouped on the basis of the genres. These genres are then classified into four pre-defined categories based on the number of users in the genre and the average active time of that genre. These four categories are sorted by priority. If there exists a clear affinity genre, viz there is only one genre in the highest priority category, then that is considered as the affinity of the group. If there is a conflict between two or more genres, then a larger group consisting of the current group is considered and the process is repeated. If there is no larger group, then the past results are analyzed and an affinity genre is identified.

3. Module Description

3.1 Module 1: Collection module

The collection module provides the source of the data for the CAARD Engine. The list of apps installed on the user device is captured through a special component that is attached to a host application installed on the user device. In major mobile operating systems like Android and iOS, every app will have an information tag about the genre to which it belongs. The genre information tag can be retrieved from the app repository where the app was hosted to be downloaded by users. This information tag is the basis to classify the apps into defined categories. The genre based classification will help in the underlying context aware approach⁶ in prediction. The app list along with its genre tag is collected for a large user base. This large database is uploaded on a cloud server for hassle free and location independent access of the data collected. Collection module gets the app list from discrete users. This captured data is used to predict the affinity of the user towards a particular genre and also that of a group of users.

3.2 Module 2: Categorization Module

Once the list of applications has been obtained from the user, this data is sent to the categorization module is shown in Figure 1. In this module, the applications are classified on basis of their genre. This genre is automatically updated along with the application list that is obtained during the data collection process. This clustering can be achieved

using a hashing procedure. The key - value pairs consist of the genre as the key and the list of application that fall into that genre as the values. This helps in enhanced mapping of the genres⁷. Once the list of applications has been sorted on basis of the genre, it is then stored into the database as well as a parallel channel is used in order to feed the data to further processes of the algorithm.

3.3 Module 3: Strength Computation Module

In order to compute the strength of each of the cluster of applications, a normalization mechanism is used. Two different types of strengths are calculated from the obtained data list. They are:

- Strength for a single user
- Strength for a group of users

The strength of a cluster of apps for a single user denotes the individual affinity of that user towards that particular cluster. This is different for each user. This value can be used by prospective developers in order to provide the users with specific content that could be useful for the user. This value is calculated using the time during which each application is active. This time is added for all applications in the cluster and divided by the number of applications in the cluster. This is named as S_0 value.

The strength of a cluster of apps for a group of users is calculated by the CAARD Engine as and when required. This part of the engine takes the parameter by which the users are to be grouped. The grouping can be done based on various parameters such as location, institution, area, age group etc. The S_0 values for the given set of users is

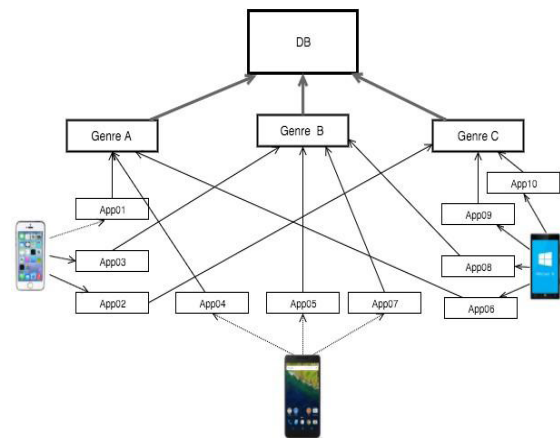


Figure 1. Categorization Module.

summed up and then divided by the number of users in the group. The formula is for calculating S_0 and S_1 is:

$$S_0 = \frac{\sum_{Applications} (\sum \text{Active duration of application})}{\text{Total number of applications in the cluster}}$$

$$S_1 = \frac{\sum_{Users} (S_0 \text{ Value of each user})}{\text{Total number of users in the group}}$$

3.4 Module 4: Cluster Analysis

Once the strengths have been calculated, it is up for the CAARD Engine to analyse the strengths and to predict the affinity of the user or a group of users. For this, the genres are classified into a pre-defined set of four categories based on the strengths and the number of users. These categories are then sorted according to priority. After sorting, there is a search for the affinity genre by following a specific procedure.

First, the categories are checked if there is a clear affinity genre. A clear affinity genre means that there is only one genre in the high priority category. In such a case, it is inferred that the user(s) have a greater affinity towards a particular genre than the others. Thus, there is no further processing to be done.

However, a single genre getting a clear affinity is a rare scenario as the characteristics of each and every user in a group is very different. Thus, there is a need for further analysis in the case of a conflict between multiple genres. In such a situation, the learning of the engine is relied on to find the affinity genre. The learning happens when similar conflicts have happened in the past. The system learns when a conflict occurs and a result is found by taking a larger group of users, that is a super set of the current group of users. During this time, the same procedure is repeated iteratively for the larger group until a clear affinity genre is obtained.

During the initial implementation of the CAARD Engine, there is no possibility of a super set being different from the set of users under consideration. When such a situation occurs, the engine temporarily chooses a random genre as the Affinity Genre. The engine then waits for the super set of users to become different than the existing set to provide an accurate analysis of the user base. When a genre is picked randomly, the engine does not learn from this result as this might not be an accurate result.

4. Work Flow of the System

The collection module of the system is attached to a host application that is present in the mobile device of the user. This module collects the list of apps that are installed on the user's device. It then calculates the active time of each application and sends it to the CAARD Engine. The CAARD Engine receives the list of apps along with the active times of each application⁸. This list is then categorized according to genres. For each of the identified genres, the strength of the genre is computed using the active times. This strength is sent back to the device to be used by the host application to provide user specific content. This content is also stored on the database for autodidactic learning and for the computation of strength for a group of users. In order to compute the strength for a group of users, the list of users who belong to the group is retrieved from the database. Then, the strength for this group is computed using the strength for each user. With the help of the strength of the genre for that group and the number of users for each genre, the genres are classified into a pre-defined set of four categories⁹. These categories are prioritized and the affinity genre is looked for.

To find the affinity genre a unique algorithm is employed which considers various factors to predict the affinity of the user(s) towards a particular genre of applications. If there exists a clear affinity genre, viz there is only one genre in the highest priority category, then that is considered as the affinity of the group. If there is a conflict between two or more genres, then the past results are analyzed and an affinity genre is identified. If there is no past conflict, then a larger group consisting of the current group is considered and the process is repeated¹⁰. This result is then used for the learning process of the engine, which is delivered as a service.

Algorithm 1: CAARD Engine Algorithm

input: List of apps along with Active Times

output: Affinity genre for user(s)

1. Genre tag for each application is found;
2. Applications with same genre tag are grouped together;
3. Parameters S_0 , S_1 are calculated for each genre;
4. Based on S_0 , S_1 & Number of users, genres are split into four pre-defined categories;
5. Categories are sorted based on priority;
6. **If single genre in high priority category then**
7. Return that genre as Affinity Genre of User(s);
8. **else**

9. **if** *same conflict hit in the past* **then**
10. Return the best of the past results;
11. **else**
12. **if** *dataset smaller than larger group's dataset* **then**
13. The enlarged dataset is passed to the CAARD Engine again;
14. Learn from result to use for future conflicts;
15. Return the Affinity Genre of the enlarged dataset;
16. **else**
17. Randomly pick a genre;
18. Return the picked genre as Affinity Genre;
19. **end**
20. **end**
21. **end**

5. Future Work

In the future, this method of predicting the affinity of the user can be implemented to web browsers and computer application by considering similar parameters. Other methods of clustering can be employed to get more accurate results. The learning mechanisms can be altered to make the engine more and more efficient. This can also be implemented as an individual application that entertains the user based on their needs and interests. It can also be used for delivering user specific advertisements or products that drives them to click on it there by generating revenue for the developers. Also its a prospective idea to concentrate more on providing context- aware content rather than contextual content to the mobile app industry as its growth will be exponential in the coming years¹¹.

6. Conclusion

The proposed system attempts to support the delivery of context-aware content in the mobile app industry. This CAARD engine optimally provides us with the user interest pattern as it initially tries to learn from the users' data. This engine uses a mathematical model to calculate the affinity of users towards particular genre or category of application. The algorithm is devised in such a way that the categories have specific priority based on the affinity of the users towards the app. This makes sure that efficiency of the system can be made better every time there is new category or genre of apps. In the mobile app

industry providing context-aware content is always been a daunting task, this system takes a step forward in providing content which satisfies their interest and also helping the development community to make apps which makes the user happy.

7. Acknowledgment

The authors would like to thank the institution for allowing us to carryout this research work. The staff and coordinators were very much supportive in bringing out this system. Their queries, comments and suggestions were helpful in shaping the system to yield more effective results.

8. References

1. Nath S, Lin SX, Ravindranath L, Padhye J. SmartAds: Bringing Contextual Ads to Mobile Apps, 2013.
2. Das TK. Intelligent Techniques in Decision Making: A Survey. Indian Journal of Science and Technology. 2016 Mar; 9(12):1–6.
3. Ayed EB, Ayed MB, Kolski C, Gargouri F, Ezzedine F. Context Aware Criteria for the Evaluation of Mobile Decision Support Systems, 2015.
4. Iberná A, Vehovar V. Using Social Network to Predict the Behavior of Active Members of Online Communities, ASONAM '09. International Conference on Advances in Social Network Analysis and Mining, 2009.
5. Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules, San Jose, CA. 1994.
6. Ko KE, Sim KB. Development of context aware system based on Bayesian network driven context reasoning method and ontology context modeling, ICCAS 2008. International Conference on Control, Automation and Systems, 2008.
7. Chen N. An Chen Integrating Context-Aware Computing in Decision Support System, Hong Kong, 2010.
8. Doukas C, Antonelli F. COMPOSE: Building Smart and Context-Aware Mobile Applications utilizing IoT Technologies, Trento, Italy, 2013.
9. Nada A, Nasr M, Salah M. Service Oriented Approach for Decision Support Systems, Egypt. 2015.
10. Sellami K, Kassa R, Dris D, Tiako PF. Taking advantage of semantic- social information in recommendation systems ISKO-Maghreb, 3rd International Symposium. 2013.
11. Alshareet OM. An Empirical Study to Develop a Decision Support System (DSS) for Measuring the Impact of Quality Measurements over Agile Software Development. Indian Journal of Science and Technology. 2015 Jul; 8(15):1–17.