Personalized App Service System Algorithm for Effective Classification of Mobile Applications

Suraj K. Shewale, Vivek V. Gayakee, Piyush D. Ugale, Hemant D. Sonawane

Abstract— **Mobile application is the important program that drives different function of mobile devices or we can say that it is the important entity of all major Smartphone, tab devices. Since, every human is different with regards to their likes and dislikes. Same is applied to their mobile application. There are different types of mobile Operating Systems and every mobile OS has their own AppStore. These Apps Store house "millions" of Apps form various Developers. Each of these Apps fits into a predefined category. With as much as millions of apps and their user, there is an immediate need to study and understand the behavior of apps with respect to its users. The study regarding the use of mobile apps plays an important role in understanding the user preference which in return helps to provide an intelligent personalized app based service. The important step to carry out such study is to classify Apps in a particular predefined category. Due to limited contextual information of an app on respective app store, it is difficult to carry out the proposed analysis. The current information is limited, incomplete and ambiguous. So, to enrich our analysis of apps there is a need to boost the current available contextual information by deep observation of different mobile apps, use of real world contents and application feedback to add more knowledge of apps from web services (Web Search Engine). Finally, combining all these collected information into one for an efficient mobile app classifier. This structured classifier in return will provide apps based on users preference.**

Index Terms—**A App, AppStore, MaxEnt, BP-Growth, L-BFGS, Snippets**

I. INTRODUCTION

The use of mobile devices is to such an extent that it is possessed by every educated or uneducated person. The important portal to these mobile devices is the "App". App as small is the name but as big in use. Be it office, home or travelling if you are using a mobile device means you are using an app, .There are different mobile OS such as Google's Android, Windows phone OS, Apple's iOS and many others. Each mobile OS has an AppStore such as Play Store, Windows Store, and Apple Store. These AppStore are house of millions of apps which are from various developers. Every app in an AppStore is categorized in a predefined manner.

The working of an AppStore is such that a user selects an app according to his/her convenience and downloads it. The selection of an app is by random/keyword searching or through recommendation by someone.With introduction to millions of app their download and uses we can say that app plays an important role in the daily lives of mobile users. Many of these apps provide us with similar kind of functionality, as a result having a classification of these apps will play an important role not only to the user in order to search the required app easily but also we can have the analysis of the user preferences which can help the intellectual services like app recommendation, user segmentation, target advertising etc. It is beneficial to understand user preferences by studying use of mobile applications. This motivates to model many intelligent, personalized app services such as app recommendation and user segmentation. To carry out user analysis and app study is not possible because of limited information on AppStore. The information directly collected from mobile AppStore is incomplete, limited and unambiguous. With such low flow of information we cannot model a new user app preference system. But with availability of high performance mobile devices and connection services a user expects an effective and automatic approach for personalized mobile app classification service.

To model the personalized mobile app classification on the basis of app-study and user analysis we need to boost the information related to app and users. We can add such information by elaborating what is available on AppStore, use of web services which is a search engine, get developers information, collect real world content also from important aspect "the users". Analyze the user app behavior and provide thorough rating system.

II. OBJECTIVES

We purpose to extract and use both web knowledge and real world context to enrich contextual information of app. An effective approach for enriching app classification information can be modeled by combining various works on text classification. To take advantage of a web search engine and obtain some snippets to describe a given app. This search engine can be Google or any other search engine. For new or rare apps we use real world context of mobile apps. More information of app can be made available by obtaining the context rich device logs of user who use them in mobile devices. We study and extract various features of apps through web knowledge and real world context. This extracted information is combined using Maximum Entropy model which is used to train a very effective and efficient app classifier.

Two kinds of textual features methods are used to capture the relevance between Apps and the corresponding category labels,

1. Explicit Feedback of Vector Space Model
2. Implicit Feedback of Semantic Topics

To extract effective contextual features of mobile Apps from real-world context logs we study three types of methods:

1.Pseudo Feedback of Context Vectors
2.Implicit Feedback of Context Topics
3.Frequent Context Patterns

After extracting both textual and contextual features, the remaining work is to train an efficient classifier, which can integrate multiple effective features for classifying Apps. For

Manuscript received December 31, 2014.

Suraj K. Shewale, IT Dept. BVCOERI, Nashik, India
Vivek V. Gayakee, IT Dept. BVCOERI, Nashik, India
Piyush D. Ugale, IT Dept. BVCOERI, Nashik, India
Hemant D. Sonawane, Comp. Dept. BVCOERI, Nashik, India.

this purpose various classification model are available such as Naive Bayes, SVMs, Decision Tree and Maximum Entropy. Among such model we choose the Maximum Entropy because it is proven to perform better than other alternative models in classifying insufficient and sparse data. Also compared with other classification approaches Maximum Entropy is more flexible to incorporate different types of features, such as the various features extracted from a Web search engine and real-world context logs.

III. PROBLEM STATEMENT

Classification of mobile apps is considered as a quite difficult task. This is because for having a proper or effective classification we need to have detailed information about the app. This is challenging task as very limited contextual information about the app is available. To be specific contextual information obtained from the apps name is very limited, as the words used for app name are very short and sparse. Hence there is an immediate need to provide an effective classification of the mobile apps by using the enriched information about the apps.

To achieve this goal, we will be exploiting not only the web knowledge but also the real world contextual features about the apps along with their word labels. This will automatically improve the contextual information of the apps, resulting into improved performance of the classification. Here the web knowledge is extracted from the general search engine like Google or from the app store, while the real world features will be extracted from the mobile usage record of the user.

IV. EXISTING SYSTEM

X. H. Phan et al [3] in their work have presented a general framework to process the short and sparse text documents on the web. They have focused mainly on data sparseness and synonyms/hyponyms by exploiting the hidden topics discovered from large scale external document collection i.e. universal data set. Here leveraging the hidden topics has improved the representation of the short and sparse text for classification. The semantic topics are the additional textual features integrated with the words to improve the classification. M. Sahami and T.D. Heilman [4], in their work they have presented a similarity kernel function based approach to find the similarity between the short text. They have found that the traditional cosine similarity measures like for example cosine coefficient produce inadequate results like suppose we for the two short texts like “AI” and “Artificial intelligence” it will give the similarity as zero though both the terms are actually related to each other. According to the results of their work, they have proved that there approach can effectively measure the similarity between short text snippets which by exploiting the web search engine and provide greater context for the short texts. Classifying the queries is an important task, as it is beneficial for a number of higher level tasks like web search and advertising matching. But search queries are usually short, thus carry insufficient information to provide accurate classification. A.Z. Broder et al [5] in their work have proposed a methodology for classifying these short queries using blind feedback technique. In which given a query its topic will be determined by the web searched results that will be returned for the query. The empirical evaluation performed by the authors proved that the methodology yields higher classification for the queries. H. Ma, H. Cao, O. Yang, E. Chen and J. Tian [6], in their work have proposed an approach which leverages search snippets to build vector space for both app usage and categories and classifies the app usage records using the cosine space distance.

V. PROPOSED SYSTEM

In our proposed system to have an effective classification of the mobile apps, we will be exploiting and collecting information from various methods like we search engine, real world contextual data, contextual log information of users etc. From this data, we obtain the features for the apps appearing in these logs. Then with the help of machine learning model available, we will train the classifier to give us the appropriate classification of the app. With drastic increase in use of mobile devices, millions of mobile apps are developed for mobile users. The large number of apps make searching and classification the immediate need. The major challenge for classification is that there are not many effective and explicit features available for classification models due to the limited contextual information of Apps available for the analysis. Current platforms do not allow developers to systematically filter, aggregate, and classify user feedback to derive requirements or prioritize development efforts. Our system also classify app which will take advantage of app Rating, Feedback, Web Search engine, the User and Developers point of view and all aspects related to the app. The greatest advantage of feedback study is that we can really examine the user comments about the app and discover its features alongside its performance, public presentation and classification. We study and extract several effective features from both Web knowledge and real-world contexts through the mining technologies and study all these aspects to form an enrich data which will provide the best quality app.

We first extracted several Web knowledge based textual features by taking advantage of a Web search engine. Then, we also used real-world context logs which record the usage of Apps and corresponding contexts to extract relevant contextual features. Finally, we integrated both types of features into a widely used MaxEnt model for training an App classifier. Our approach is both efficient and effective for solving the problem of automatic App classification. Mobile devices have very limited computing resources, it is necessary to design a more effective service framework which can reduce the load of mobile devices. We come with client server architecture such that processing will be done on server which will eventually reduce the load and increase performance of mobile devices. Developers would also benefit from enriching textual feedback with usage and context data. Feedback helps developers to understand user needs extending the application towards crowd sourcing requirements.

VI. ALGORITHM USED

A. Maximum Entropy Algorithm(MaxEnt)

Maximum entropy can be traced back along multiple threads to Biblical times. Recently, computers have become powerful enough to permit the wide scale application of this concept to real world problems in statistical estimation and pattern recognition. It describes a method for statistical modeling based on maximum entropy. Which present a maximum-likelihood approach for automatically constructing maximum entropy models and describe how to implement this approach efficiently, using as examples several problems in natural language processing.

B. *Stemming Algorithm*

Stemming is the term used in linguistic morphology and information retrieval to describe the process for reducing in selected (or sometimes derived) words to their word stem, base or root form-generally a written word form. The stem needs not to be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Many search engines treat words with the same stem as synonyms as a kind of query expansion. Preprocessing steps to save both space and time requirements by using improved Stemming Algorithm. Stemming algorithms are used to transform the words in texts into their grammatical root form. Several algorithms exist with different techniques. The most widely used stemming algorithm is Porter stemming algorithm.

C. *Stop words*

In computing, stop words are words which are filtered out before or after processing of natural language data (text). There is not one definite list of stop words which all tools use and such a filter is not always used. Any group of words can be chosen as the stop words for a given purpose. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as 'The Who', 'TheThe', or 'Take That'. Other search engines remove some of the most common words including lexical words, such as "want"-from a query in order to improve performance.

D. *Frequent Itemset Mining (FIM)*

The frequent itemset mining (FIM) is one of the most important techniques to extract knowledge from data in many real-world applications. YAFIM (Yet Another Frequent Itemset Mining), a parallel Apriori algorithm based on the Spark RDD framework-a specially-designed in-memory parallel computing model to support iterative algorithms and interactive data mining. Experimental results show that, compared with the algorithms implemented with MapReduce, YAFIM achieved speedup in average for various benchmarks. Especially, we apply YAFIM in a real-world medical application to explore the relationships in medicine.

E. *BP-Growth*

Used for mining frequent context patterns.The basic idea of the algorithm is partitioning the original context logs into smaller sub-context logs for reducing the mining space and mining frequent context patterns in these sub-context logs. BP-Growth combines two optimizing strategies for association rule mining and the experimental results on real context data clearly show that it significantly outperforms GCPM and other two baselines in terms of both running time and memory cost.

F. *L-BFGS*

Broyden Fletcher Goldfarb Shanno is an iterative method for solving unconstrained nonlinear optimization problems. BFGS methods approximate Newton's method, a class of hill-climbing optimization techniques that seeks a stationary point of a (preferably twice continuously differentiable) function. For such problem a necessary condition for optimality is that the gradient be zero.

VII. METHODOLOGY FOR DEVELOPMENT

A. *App Taxonomy*

To recognize the semantic meanings of Apps, we can classify each App into one or more categories according a predefined App taxonomy specifically, App taxonomy is a tree of categories where each node corresponds to a predefined App category. The semantic meaning of each App can be defined by the category labels along the path from the root to the corresponding nodes.

B. *Search Snippets*

We use the Web knowledge to enrich the textual information of Apps. To be specific, we first submit each App name to a Web search engine (e.g., Google or other App search engines), and then obtain the search snippets as the additional textual information of the corresponding App. search snippet is the abstract of the Web page which are returned as relevant to the submitted search query. The textual information in search snippets is brief but can effectively summarize the corresponding web pages. Thus, they are widely used for enriching the original textual information in the short text classification problem.

C. *Context Log*

Smart mobile devices can capture the historical context data and the corresponding App usage records of users through context-rich device logs or context logs for short. For example, a context log which contains several context records and each context record consists of a timestamp, the most detailed contextual information at that time, and the corresponding App usage record captured by the mobile device. The contextual information at a time point is represented by several contextual features (e.g., Day name, Time range, and Location) and their corresponding values (e.g., Saturday, AM8:00-9:00, and Home), which can be annotated as contextual feature-value pairs. Moreover, App usage records can be empty (denoted as “Null”) because users do not always use Apps, location related raw data in the context logs, such as GPS coordinates or cell IDs, have been transformed into semantic locations such as “Home ”and “Work Place” by a location mining approach. The basic idea of such approach is to find the clusters of user positions and recognize their semantic meanings through the time pattern analysis.

VIII. CONCLUSION

In our system, we have studied methods in which we will be extracting the contextual information from sources like information from the labels (app name), information from the web search engine (snippets) and the contextual usage history of the app collected from the users-usage record. This is will give us effective and secure classification of the apps as most of these apps are coming from an unknown vendors and so they have the higher possibility of being unclassified.

ACKNOWLEDGMENT

It is my pleasure to express my knowledge to my respected sir Mr. H. D. Sonawane, Computer Engineering, BVCOE&RI, Nashik for his valuable guidance, inspiration and continues support. This paper could not be success without apps analysis done which help to understand the necessity for this paper.

REFERENCES

[1] Hengshu Zhu, Enhong Chen, Hui Xiong, Huanhuan Cao, and Jilei Tian, “Mobile App Classification with Enriched Contextual Information ", IEEE Transactions on mobile computing (Volume: 13, Issue: 07), 7 July 2014.

[2] K. Nigam, “Using maximum entropy for text classification,” in Proc. IJCAI Workshop Machine Learning for Information Filtering, 1999, pp.61–67.

[3] X.-H. Phan et al., “A hidden topic-based framework toward building applications with short web documents,” IEEE Trans. Knowledge. Data Eng. vol. 23, no. 7, pp. 961–976, Jul. 2010

[4] M. Sahami and T. D. Heilman, “A web-based kernel function for measuring the similarity of short text snippets”, in Proc. WWW, Edinburgh, K., 2006, pp. 377–386.

[5] Z. Broder et al., “Robust classification of rare queries using web knowledge,” in Proc. SIGIR, Amsterdam, Netherlands, 2007, pp. 231–238.

[6] H. Ma, H. Cao, Q. Yang, E. Chen, and J. Tian, “A habit mining approach for discovering similar mobile users”, in Proc.WWW, Lyon, France, 2012, pp. 231–240.

[7] H. Cao, T. Bao, Q. Yang, E. Chen, and J. Tian.“Aneffective approach for mining mobile user habits”, in Proceedings of the 19th ACM Conference on Information and Knowledge Management