CLASSIFICATION OF MOBILE APPLICATIONS WITH RICH INFORMATION

Senthil kumar .B, Ravi .R

M.E (CSE) Second year, Assistant Professor

J.J College of Engineering and Technology, Trichy.

Abstract:

Mobile Application activates an important role in the daily lives of mobile users. Intuitively, the study of the use of mobile Apps can help to understand the user favorites, such as App recommendation, user segmentation and target advertising. The major challenge 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. From the CDMA to recent mobile applications more user activities have improved. Still now all mobile app have limited contextual information in their names, and the only available explicit features of mobile Apps are the semantic words contained in their names and proposing enriched contextual information of mobile Apps by exploiting the additional Web knowledge from the Web search engine. Then, inspired by the observation that different types of mobile Apps may be relevant to different real-world contexts, also extract some contextual features for mobile Apps from the context-rich device logs of mobile users. By collecting details about the user side information regarding the app usage to categorizing the mobile application on own. Specifically natural language processing has been driven for the classification work. Finally combining all the enriched contextual information into the classified framework model for training the mobile App classifier.

Index Terms— App classifier, real-world contexts, Web Knowledge, context information

1. INTRODUCTION:

Mobile apps were originally offered for general productivity and information retrieval, including email, calendar, contacts, and stock market and weather information. However, public demand and the availability of developer tools drove rapid expansion into other categories, such as mobile games, factory automation, GPS and location-based services, banking, order-tracking, ticket purchases and recently mobile medical apps. The explosion in number and variety of apps made discovery a challenge, which in turn led to the creation of a wide range of review, recommendation, and sources, including blogs, magazines, and dedicated online app-discovery services.

In recent years, the development of mobile devices progressed at an ever fastest pace to make possible supporting various applications and services beyond the traditional speech centric service, such as music, videos, web browsing, gaming and camera shooting, just to name a few. The rich user interaction information captured by the mobile device can be used to understand user habits, which can bring a great business value, such as data-driven user studies for marketing, targeted advertising and personalized recommendation. Consequently, studying the habits of mobile users through their interactions attracts many researchers' attention.

As part of the development process, Mobile User Interface (UI) Design is also an essential in the creation of mobile apps. Mobile UI considers constraints & contexts, screen, input and mobility as outlines for design. The user is often the focus of interaction with their device, and the interface entails components of both hardware and software. User input allows for the users to manipulate a system, and device's output allows the system to indicate the effects of the users' manipulation. Mobile UI design constraints include limited attention and form factors, such as a mobile device's screen size for a user's hand. So usage has to be depends on arrangement of app presentation.

Classification is considered an instance of supervised learning, which is learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance. Often, the individual observations are analyzed into a set of quantifiable properties, known variously explanatory variables, features, etc. These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a part word in an email) or real-valued (e.g. a measurement of blood pressure).

Classifiers work by comparing observations to previous observations by means of a similarity or distance function. An algorithm that implements classification, especially in a concrete implementation, is known as a classifier. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category. Implicit Contextual information is the identity of things named in the text such as people, places, books, etc. So Information about mobile software information usage named in the text as date that have used that application, geographical locations and date published. Interpretive information is themes, keywords and all. With respect to this data, classification can be processed. Prior implementations of language-processing tasks typically involved the direct hand coding of large sets of rules.

The machine-learning paradigm calls instead for using general learning algorithms often, although not always, grounded in statistical inference to automatically learn such rules through the analysis of large corpora of typical real-world examples. A corpus (plural, "corpora") is a set of documents (or sometimes, individual sentences) that have been hand-annotated with the correct values to be learned.

Many different classes of machine learning algorithms have been applied to NLP tasks. These algorithms take as input a large set of features that are generated from the input data. Some of the earliest-used algorithms, such as decision trees, produced systems of hard if-then rules similar to the systems of hand-written rules that were then common. Increasingly, however, research has focused on statistical models, which make soft, probabilistic decisions based on attaching real-valued weights to each input feature. Such models have the advantage that they can express the relative certainty of many different possible answers rather than only one, producing more reliable results when such a model is included as a component of a larger system. Simultaneously web knowledge about the mobile application can be gathered and using as explicit information. Each and every usage of mobile app entities has been

collected and using for the description identification.

On developing mobile application session a critical problem of this type of features is that the lengths of App names are usually too short and the contained words are very sparse. As a result, it is difficult to train an effective classifier by only taking advantage of the words in App names. Moreover, the available training data are usually with limited size and may not cover a sufficiently large set of words for reflecting the relevance between Apps and category labels. Therefore, a new App who's partial, or all words in name do not appear in the training data will not obtain accurate classification results if the classifier is only based on the words in App names. To this end, process will consider the effective features which can capture the relevance between Apps and category labels.

2. TAXONOMIES

2.1 EXTRACTING EFFECTIVE FEATURES FOR APP CLASSIFICATION

2.1.1 Web Based Textual Features

A new App whose partial, or all words in name do not appear in the training data will not obtain accurate classification result if the classifier is based on the word in App names. The effective features which can capture the relevance between Apps and category labels. The features extracted from both the relevant web knowledge and real-world contextual information for training an App classifier.

2.1.2 Real-World Contextual Features

Extract an effective contextual feature of mobile Apps from real-world context logs. Three types of contextual features are Pseudo Feedback of Context Vectors, Implicit Feedback of Context Topics and Frequent Context Patterns. The usage of a particular category of App is relevant to some contextual feature-value pairs. The Apps in the "*Game/Strategy Game*" category, they may be relevant to the contextual feature-pairs (Day period: Evening) and (Location: Home) respectively. Based on this assumption, similar to the VSM-based approach introduced to build a context vector for each App category as follows. First, for each pre-selected and labeled App *a*, we collect all context records which record the usage of App *a* from the context logs of many mobile users.

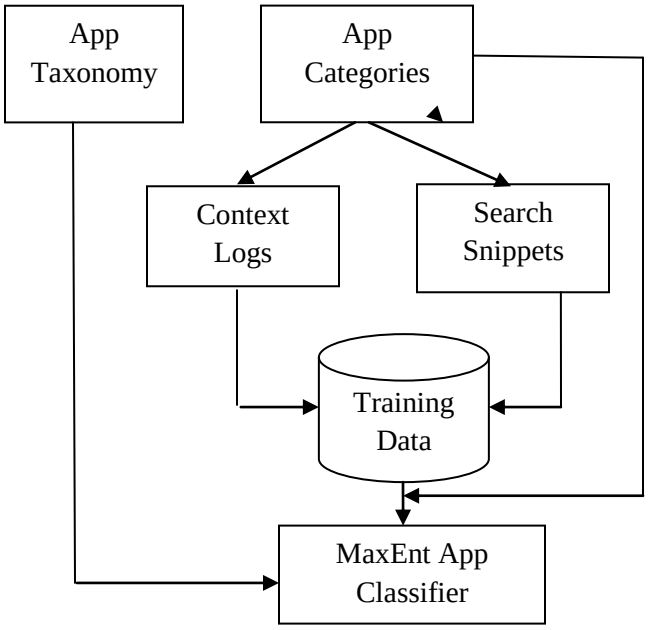

Fig 2.1 Framework of App classification approach

2.2 Search Snippets

The Web knowledge to enrich the textual information of Apps. A 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 an effectively summarize the corresponding web pages. 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.

2.3 Context Logs

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 an example of 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. In 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 [20]. The basic idea of such approach is to find the clusters of user positions and recognize their semantic meanings through the time pattern analysis.

3. LITERATURE SURVEY

3.1 Query Classification

In [1], H. Cao et al., studied for QC algorithms classify individual queries without considering their context information. Many Web queries are short and ambiguous, whose real meanings are uncertain without the context information. Mainly, context information into the problem of query classification by using conditional random field (CRF) models. Query classification has been by classifying user queries into a ranked list of predefined target categories.

Query classification is dramatically different from traditional text classification because of two issues. First, Web queries are usually very short, like most queries contain only 2-3 terms. Second, many queries are ambiguous, and it is common that a query belongs to multiple categories. Manually labels 800 randomly sampled queries from the public data set from ACM KDD Cup'051, and 682 queries have multiple category labels. QC can be classified into two categories depending on the types of taxonomy. In the first category, the taxonomy is defined by considering the Web query types.

3.1.1 Modeling search context by CRF

The Conditional Random Field (CRF) model is a discriminative graphical model, which focuses on modeling the conditional distribution of unobserved state sequences given an observation sequence. The strength of processing sequential data and incorporating rich features makes CRF model particularly suitable for context-aware query classification.

Huanhuan Cao (2009) categorized three important Query classification approaches. The first category tries to augment the queries with extra data, including the search results returned for a certain query, the information from an existing corpus, or an intermediate taxonomy. The second category leverages unlabeled data to help improve the accuracy of supervised learning. The third category of approaches expands the training data by automatically labeling some queries in some click through data via a self-training-like approach.

3.2 TEXT CLASSIFICATION

In [2], K. Nigam outline the text classification. Maximum entropy is a

probability distribution estimation technique widely used for a variety of natural language tasks, such as language modeling, part-of-speech tagging, and text segmentation. The principle of maximum entropy is that without external knowledge, one should prefer distributions that are uniform. Constraints on the distribution, derived from labeled training data, inform the technique where to be minimally non-uniform. The maximum entropy formulation has a unique solution which can be found by the improved iterative scaling algorithm.

The maximum entropy is used for text classification by estimating the conditional distribution of the class variable given the document. The underlying principle of maximum entropy is that without external knowledge one should prefer distributions that are uniform. Constraints on the distribution, derived from labeled training data, inform the technique where to be minimally non-uniform.

The techniques that would be more appropriate for maximum entropy. Over fitting is reduced, and performance improves. But, the results indicate that maximum entropy may be sensitive to poor feature selection. It is not for maximum entropy is a combination of a class and a word, there is no need to have features for all classes for a vocabulary word.

3.3 AN UNSUPERVISED APPROACH MODEL

In [3] T. Bao and H. Cao proposed the personalized contexts for mobile users. Mobile context modeling is a process of recognizing and reasoning about contexts and situations in a mobile environment, which is critical for the success of context aware mobile services. On mobile context modeling, the use of unsupervised learning techniques for mobile context modeling is used. An unsupervised technique has the ability to learn personalized contexts which are difficult to be predefined. An unsupervised approach to modeling personalized contexts of mobile users.

Segment the raw context data sequence of mobile users into context sessions where a context session contains a group of adjacent context records which are mutually similar and may reflect the similar contexts. They used an adaptive segmentation approach named the minimum entropy segmentation to address the challenges of context segmentation on determining the number of segments and the segmentation threshold. If an unsupervised approach can discover that the contextual feature-value pairs (Is a holiday?: Yes), (Time range: AM10:00-11:00), (Movement: Moving), and (Cell ID: 2552) are highly related.

They produced three important categories of about the context based unsupervised approach. In the first category, contexts are modeled manually based on domain knowledge. For example used key-value pairs to model the context by providing the value of context information (e.g. location information) to an application as an environment variable. In which prototypes of a mobile context-aware tour guide were built.

The context with ontology's and analyzed psychological studies on the difference between recall and recognition of several issues in combination with contextual information. Indeed, none of the above studies adopted machine learning approach for learning contexts from the raw context data automatically. As a result, they may work well in simple environments, such as guiding tourists in tourist attractions, but are not flexible for applying to more complex environments where it is difficult to build context models.

The second category includes the mobile context modeling though supervised learning approaches. An individual's

transportation routine given the user raw GPS data. By leveraging a dynamic Bayesian network, the system learns and infers the person's transportation routines between the significant places. Then exploited to use several supervised learning approaches for modeling user's raw GPS data. In their work, four different inference models including decision tree, Bayesian network, support vector machine (SVM) and conditional random field (CRF) are studied for modeling user's transportation mode. Supervised learning approach provides more flexibility than the manual approach for mobile context modeling because it depends on less domain knowledge and can learn from the raw context data automatically. However, it still needs to manually predefine the contexts. Moreover, it needs a number of labeled training data for model training. By contrast, the unsupervised learning approach for mobile context modeling is very flexible because it can learn a context from an individual user's neither raw context data without predefined contexts nor labeled training data. Thus, it can greatly improve the user experience due to less dependency on the user. Manually, the recognizing user's contexts in their daily life.

The third category of related work focuses on user modeling through unsupervised approaches. In a latest literature, proposed to use the eigenvector of user behavior for modeling individual users and infer community affiliations within the subjects' social network. Though they also used an unsupervised approach to discover the user context and behavior pattern from the user history data, the objective of their research is intrinsically different from their work. The goal is to discover the personalized mobile contexts which can be applied to context-aware services. In addition, they proposed exploits topic models, which are widely used generative probability models in document modeling. Typical topic models include the Mixture Unigram (MU) model, the probabilistic latent semantic analysis (pLSA) model, and the latent Dirichlet allocation (LDA) model. Most of other topic models are extended from them and applied to specific applications. In extended MU to MUC and extend LDA to LDAC for satisfying the constraint of context data. MUC is in a too complex form and it may not be feasible to calculate the parameters of the model directly.

3.4 A HYBRID RECOMMEND SYSTEM FOR RECOMMENDATIONS

In [4], W. Woerndl et al., proposed the context-aware recommendations of mobile applications. The context in recommender systems in the domain of mobile applications. The approach recommends mobile applications to users based on what other users have installed in a similar context. The idea is to apply a hybrid recommender system to deal with the added complexity of context. Users can select among several content-based or collaborative filtering components, including a rule based module using information on point of interests in the vicinity of the user, and a component for the integration of traditional collaborative filtering.

A hybrid recommender system combines different recommender systems to improve information retrieval. The combination can be done using several alternatives, for example: weighted, switching, mixed, feature combination or augmentation or cascading. This idea is to combine collaborative filtering with other recommenders to deal with the added complexity of context. From a conceptual point of view, this means reducing the complexity of the item-user context matrix by applying a cascading hybrid

recommender. That means, first only two dimensions of the matrix are analyzed, and in a second step the third dimension is considered in addition. In more detail:

1. Use content- or knowledge based filtering to find relevant items based on context, for Example taking the current end user device and location into account.

2. Apply collaborative filtering to rank and additionally filter the result set from step 1.

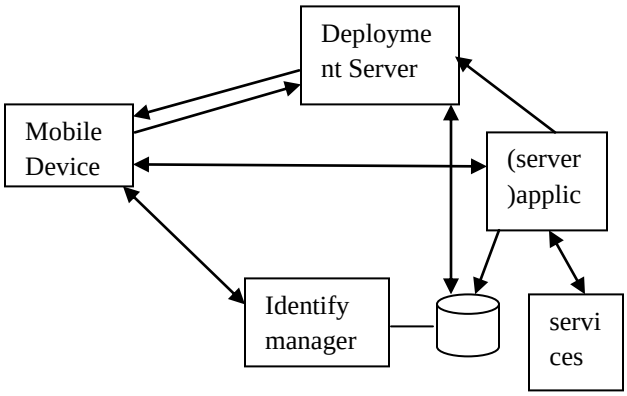

Fig 3.4.1 Hybrid Recommend system

Some important recommender components:

CFAppRecommender: Apply collaborative filtering to generate results.
LocationAppRecommender: This is a "context-item" recommender, applications that installed in a similar location are recommended.
PoiAppRecommender: Knowledge-based approach based on "point-of-interests" in the current location of the user.
RandomAppRecommender: Show a random list of applications.

3.5 BP GROWTH SEARCHING STRATEGIES FOR BEHAVIOR PATTERN MINING

In [5], X. Li et al., identify a problem on user understanding, which is critical for improving a wide range of personalized intelligence services. To mine the user behavior patterns which characterize the habits of mobile users and account for the associations between user interactions and context captured by mobile devices. They defined the behavior pattern kept two important notions which are confidence and support. Behavioral design patterns are design patterns that identify common communication patterns between objects and realize these patterns. These patterns increase flexibility in carrying out this communication.

The optimizing strategies for association rule mining to behavior pattern mining and proposed a novel and efficient algorithm named BP (**B**ehavior **P**attern

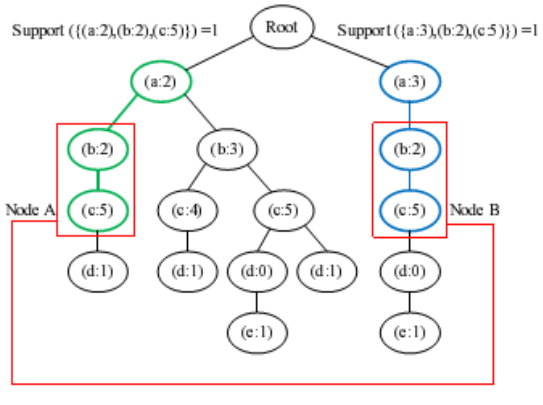

Fig. 1. The FP-Tree for the context log in Table II.

Growth) Growth for mining behavior patterns. Importantly, they used two algorithms redundant context records denote the context records which have duplicated contexts with adjacent context records and only contain empty interaction records. The pseudo code of Search BP () where for each promising 1 context C1p, an array is constructed for recording its support with respect to each interaction record.

FP-Tree is also widely used for reducing the size of the original transaction

database in association rule mining. FPTree is a prefix tree of the lists of frequent items of the original transaction database. A FP-Tree is usually much smaller than the original transaction database because transactions usually share the same prefix. This example of FP-Tree for the context log in Table II by taking contextual feature-value pairs as items.

Node A contains a support vector for recording the supports of {(a: 3), (b: 2), (c: 5)} with respect to I1, I2, and N0. In contrast, in association rule mining, each node of FP-Tree only records the frequency that the item set derived from the path to the root node occurs as prefixes of transactions.

3.6 AN EFFECTIVE APPROACH FOR MINING MOBILE USER HABITS

In [6], Q. Yang and E. Chen present the design of a mining for mobile user habits. The user interaction with the mobile device plays an important role in user habit understanding. They proposed to mine the associations between user interactions and contexts captured by mobile devices, or behavior patterns for short, from context logs to characterize the habits of mobile users. The rich user interaction information captured by the mobile device can be used to understand user habits, which can bring a great business value, such as targeted advertising and personalized recommendation

2.6.1 Context

A contextual feature set $F= \{f1, f2... fK\}$, a contextCi is a group of contextual feature value pairs, i.e., $Ci= \{(x1:v1), (x2:v2)..(xl:vl)\}$, where$xn \in F$and $vn$is the value for $x(1 \leq n \leq l)$. A context with $l$contextual feature-value pairs is called al-context.

2.6.2 Sub-context, super-context

Two contexts $Ci$and Cj, if $\forall pi \in Cjpi \in Ci$, where pi denotes a contextual feature-value pair, Cj is called a sub-context of Ci and Ci is called a super-context of Cj.A contextual feature denotes a type of context data, such as day period, location, audio level, etc. For simplicity of operating contexts, such as context comparison, which require that contextual feature-value pairs be sorted in a predefined order of contextual features.

2.6.3 Interaction record

An interaction record is an item in the interaction set Γ= $\{I1, I2...IQ\}$, where $(1 \leq n \leq Q)$ denotes a kind of user interaction. Interaction records capture the occurrences of user interactions with mobile devices, such as listening to music, message session or Web browsing.

2.6.4 Context record, context log

A context record $r=<Tid, Ci, I>$is a triple of a timestampTid, a context Ci, and a user interaction record I. A context log $R= r1r2...rNR$is a group of context records ordered by timestamps. A context record captures the most detailed available context and the occurrence of a user interaction during a time interval. Which mention "available" because a context record may miss the values of some contextual features though the set of context-features whose values should be collected is predefined. Moreover, interaction records can be empty (denoted as "Null") if no interaction happen during the time interval.

The problem has been addressed by the joining stage of GCPM. A CH-Tree has a tree-like representation of the nodes as follows.

Root Node (RN): Each CH-Tree has one root node. This node contains Kpointers that point to intermediate nodes or null, where $K$is the total number of contextual features in a given contextual feature set.

Intermediate Node (IN): Each intermediate node contains Hpointers that point to leaf nodes or null, where $H$is the output range of a predefined hash function.

Leaf Node (LN): Each leaf node contains a group of contexts. Each context Ci is associated with array of support counters $Ci.$, a Boolean tag $Ci.$To indicate whether the nearest context range of Ci contains at least one non-empty interaction record and a tag $Ci.$ To indicate whether Ci is matched by the current context record r.

4. CONCLUSION

The problem of automatic App classification along with the contextual information in App names is insufficient and sparse for achieving a good classification performance. An approach for classifying mobile Apps by leveraging both web knowledge and relevant real world context. And also to leverage real world context logs which record the usage of corresponding contexts to extract relevant contextual features. Finally to combine our classification approach with other context-aware services, such as context-aware App recommender system, to enhance user experiences.

REFERENCES

[1] H. Cao et al., "Context-aware query classification," in Proc. SIGIR, Boston, MA, USA, 2009, pp. 3–10.

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

[3] T. Bao, H. Cao, E. Chen, J. Tian, and H. Xiong, "An unsupervised approach to modeling personalized contexts of mobile users," in Proc. ICDM, Sydney, NSW, Australia, 2010, pp. 38–47.

[4] W. Woerndl, C. Schueller, and R. Wojtech, "A hybrid recommender system for context-aware recommendations of mobile applications," in Proc. ICDE, Istanbul, Turkey, 2007, pp. 871–878.

[5] X. Li, H. Cao, H. Xiong, E. Chen, and J. Tian, "BP-growth: Searching strategies for efficient behavior pattern mining," in Proc. MDM, Bengaluru, India, 2012, pp. 238–247.

[6] H. Cao, T. Bao, Q. Yang, E. Chen, and J. Tian, "An effective approach for mining mobile user habits," in Proc. CIKM, Toronto, ON, Canada, 2010, pp. 1677–1680.

[7] 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.

[8] R. Malouf, "A comparison of algorithms for maximum entropy parameter estimation," in Proc. COLING, Stroudsburg, PA, USA, 2002, pp. 1–7.

[9] 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.

[10] R. Malouf, "A comparison of algorithms for maximum entropyparameter estimation," in Proc. COLING, Stroudsburg, PA, USA, 2002, pp. 1–7.

[11] W.-T. Yih and C. Meek, "Improving similarity measures for short segments of text," in Proc. 22nd Nat. Conf. Artif. Intell., vol. 2. 2007, pp. 1489–1494.

[12] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian, "Towards personalized context-aware recommendation by mining context logs through topic models," in Proc. PAKDD, Kuala Lumpur, Malaysia, 2012, pp. 431–443.