Automatically Triggering Activity and Product Predictions in Mobile Phone Based on Individual's Activity

Kalpana Algotar
Retail Solution Division
Intel Corporation
Phoenix, USA
e-mail: Kalpana.a.algotar@intel.com

Sanjay Addicam
Retail Solution Division
Intel Corporation
Phoenix, USA
e-mail: Sanjay.v.addicam@intel.com

***Abstract*—The technological advances in mobile phone and their widespread use has resulted in the big volume and varied types of mobile data we have today. Researchers have begun to mine mobile data in order to predict a variety of social, economic, personal, location and health related events. Mobile data directly reflects individual's life without disclosing personal information, and therefore it is an important source to analyze and understand the underlying dynamics of human behaviors or activities. In this paper, we describe an innovative and challenging process to predict user's activity using mobile based data. We propose a graph-based framework that uses the user's activities, social network, and product-keywords in order to provide recommendations which are also delivered through mobile phones. This paper summarizes the different types of prediction logic algorithms by constructing graphs from different data sources. Our graph-based approach is highly scalable and can be used to predict individual's next activity, as well as prediction towards products purchase. The mobile recommendation engine incorporates three types of data to generate the graph and to predict activity and product. First, we collect product-keywords using text-rank algorithm. Second, we collect individual mobile's past data, such as accelerometer, call log, battery status, app usage, browsing history, Facebook data, and Twitter data. Third, we collect user's mobile phone activity 8 times during the day. By using multimap, we get fast prediction in real-time mobile.**

Keywords-Text-Rank Algorithm; Multimap; Adjacency List; Internal Prediction; External Prediction.

I. INTRODUCTION

The knowledge of user activities and habits is a crucial factor for the development of highly personalized applications that can be beneficial in many areas of daily life [3]. The mobile users' behaviors (e.g., SMS, call history, location, app usage, battery status, accelerometer Facebook, Twitter, etc.) are all related to real-world behaviors. This provides an unprecedented opportunity for us to understand the underlying dynamics of users' behaviors in the mobile data [2]. In this work, we aim to answer an interesting question, i.e., whether we can predict a user's next activities based on his/her historic behavior log/activities, such as call log, browser history, app usage and mobile social network information.

In this paper, we explore and develop novel methods for recognition of user's next activities. Mobile devices present an ideal platform for this task; they usually possess considerable computational resources, a rich set of wireless communication and multimedia features [3]. Nowadays, people tend to always carry their phones along. We employ a mobile phone as the main sensory and processing unit for learning and predicting user's behavior [3].

Recently, considerable related works have been conducted, e.g., activity recognition [4]-[9], dynamic emotion analysis [10]-[14], dynamic social network analysis [15]-[19], and social influence analysis [20]-[24]. Emotion analysis is to study how an individual's emotional state (e.g., happiness and loneliness) propagates through social relationships [10]-[12]. Dynamic social network analysis is to model how friendships drift over time using a dynamic model [18] or to investigate how different pre-processing decisions and different network forces such as selection and influence affect the modeling of dynamic networks [19].

This paper proposes a Mobile Recommendation Engine/Framework (MRF), which builds graph using different data sources, and applies different types of prediction logic using multimap structure which enables to get user's next activity e.g., walking, calling, eating etc., as well as product prediction that user is interested to see or purchase. Multimap is a generalization of a map or associative array abstract data type [25]. The reason behind using multimap is that it allows storing multiple values for every key, as well as, it allows storing duplicate keys. It has useful methods, i.e., invertFrom, which copies each key-value mapping in source into destination, with its key and value reversed without using loop. For backtracking logic, we traverse the path in the reverse direction until we hit the root node that is not have any parent node. In this case, consider every node in the reverse path as a value and find the attached keys for each corresponding values using invertForm method of multimap. Here, we present a different method using user's context (or activity) in mobile to predict user's next action.

The other sections in this paper are organized as follows. Section 2 illustrates the source of data. Section 3 describes our approach. Section 4 presents the process for deploying the experiment and the results of the process. Section 5 concludes the paper.

 ISBN: 978-1-61208-415-2

II. SOURCE OF DATA

In our experiment, we used the following different data sources.

A. Historical Data

We have (i) historical data, such as call log, battery status, app usage, browsing history, accelerometer data, and (ii) social network data, such as Twitter, Facebook data based on individual's past activities on mobile, etc. We have used this information without revealing individual's personal information.

B. Product Data

We crawled the Amazon site and collected around 750 products reviews using an HTML-embedded scripting language (PHP) script. We generated top keywords from those reviews using Text-Rank algorithm [27]. This algorithm provides unique keywords along with weight for each product. The Text-Rank algorithm helps to distinguish each product based on unique keywords.

C. User's Behaviour Data

Every day, we collect user's activities on mobile, such as call log, app usage, browser history, accelerometer data, social network data, such as Facebook, and Twitter at three hours intervals and build subgraph. After experimenting on various hours interval, we found that three hours interval is sufficient to collect the required amount of activities to build subgraphs.

III. OUR APPROACH

To build a mobile framework, we collect different types of data, such as historical data, product-keyword data, individual's activity on mobile, and build 200,000 nodes on a graph. We stored this graph in cloud because the size of the graph is big and one can send graph to individual's mobile phone based on his/her matching profile. We used graph cut algorithm on main graph that generates many subgraphs. Then, we used Approximate Subgraph Matching (ASM) [28] algorithm to pick the subgraph that is more relevant to individual's activity. This relevant subgraph is loaded on individuals' mobiles every 24 hours. We build backtrack and product prediction logic on subgraph that is loaded in mobile and based on last 3 hours user activities used to give future prediction.

IV. EXPERIMENTS

Graph technology is the process of analyzing large volume of data from different perspectives and summarizing it into useful information – information that can be used for prediction. Here, we developed three types of prediction logic to get three types of predictions. We implemented different graph algorithms, to build, merge, cut, and compare the graph, and to get prediction for individual's activity and product preference. First, we collect the data from the different sources. Second, we defined relationships between different data sources and put those data as nodes and edges on graph, as presented in Figure 1. For example, user node is connected with different activity nodes, such as call log, app usage, accelerometer, and browsing history. We extracted keywords from browsing history and that node is connected with extracted keywords and keyword nodes are connected with product nodes. We placed this big constructed graph on Content Management Server (CMS). Third, we collect user's activity every 3 hours and represent those activities as a subgraph. At the end of the day, we have 8 subgraphs. Fourth, using graph merge algorithm, we merge 8 subgraphs into one subgraph. Fifth, we compare this merged graph with main graph and put nodes on main graph which are not exist in main graph. In this stage, we update the main graph on CMS using merge graph. Sixth, using graph-cut algorithm, we cut the main graph into no. of subgraph. Lastly, using ASM algorithm, we compare the merge graph from step-4 with subgraph from step-6 based on node, edge weight, edge, direction and most likely matched subgraph send to that user's mobile from CMS every 24 hours. We are calculating edge weight in two ways. One is occurrence and the other is time difference. Occurrence is calculated based on repetition. For example, after outgoing_3, walking is done. This action is repeated 6 times, then the occurrence edge weight is 6. If the time difference between the call and walk is 45 minutes, then the time difference edge weight is 45 minutes. Next time, if the time difference is 30 minutes, the edge weight is updated as the average of 30 and 45 minutes. The 3rd time, if the time difference is 20 minutes then the edge weight is calculated as 20+30+45/3. For predicting user activity and product preference, we utilize user activities for the last 3 hours as an input.

A. Internal Prediction

We collect the last 3 hours of the user's activities from individual's mobile and use them as an input on subgraph, as shown in Figure 1. We search each activity as value on subgraph. If value exists on subgraph, then, we find the key for value recursively until we hit the orphan node that is not having any parent node. For example, we have two user actions 5 and 6 in the last 3 hours. First, take user action 5 as an input value and find the attached keys as 4 on subgraph and store key 4 as a key node into collection hashmap-A. Hashmap is a data structure used to store object and retrieve it in constant time O(1). It works on hashing principle in Java. Now, take one-by-one node as a value from hashmap-A and find the attached key as 3 for value 4 and append the key node into hashmap-A and we will have keys {4, 3} and marked node 4 as processed. We repeat the same process for unprocessed nodes of hashmap-A and append the result to hashmap-A. At the end, the result for user activity 5 will be {4, 3, 2, 1}. In this case we stop at node 1, because no edge is coming into it. Likewise, take the second user activity, say 6, and follow the same process as mentioned for node 5 and store the key of each recursive operation into new collection hashmap-B. The result for user activity 6 will be {2, 1}. At the end, we intersect two collections hashmap-A {4, 3, 2, 1} and hashmap-B {2, 1} and store the intersection result {2, 1} into the collection hashmap-C.

If more than two user actions exist, then, we follow intersection between first and second, second and third, third and fourth, and so on, and append each intersection result into

 ISBN: 978-1-61208-415-2

collection hashmap-C. Then, we apply Depth First Search algorithm [29] to find the path between source and destination. We take the first node, say 2, as a source from hashmap-C, and the first user activity from last 3 hours, say 5, as a destination and pass these data/nodes as parameters to Depth First Search algorithm. If a path exists between these two nodes, then, we give source node 2 as an internal prediction. If a path does not exist between source and destination, then, we choose second node 1 as source from hashmap-C and user activity 5 as destination and pass these values to depth-first search algorithm. We follow the above process recursively, until we find a path.

B. *External Prediction*

In this stage, we take one by one user's activities from last 3 hours storage and find on subgraph that is loaded in mobile every 24 hours. Here, we consider user action 5 and 7 as keys and find the corresponding attached values, say {8, 9, 10} and {11, 12}. Then after, we move in the forward direction with one degree depth. If a user activity exists in the subgraph Figure 1, and has more than one outgoing edge, then, choose the highest edge weight node that is connected with the user activity. We follow the same process for each user activity node, and, at the end, we select the outgoing edge with maximum weight as external prediction among all user activity's highest edge weight node. In this case, we select the node with highest edge weight as an external prediction out of other nodes {8, 9, 10, 11, 12}.

C. *Product Prediction*

In this stage, we created separate adjacency list for product-keywords in which a keyword acts as a key and products act as a value. We check if the type of user activity is keyword, then, we follow the product prediction logic using product-keywords adjacency list. We can consider that user activities 2 and 4 are keywords and find those nodes as keys on subgraph which is defined in Figure 1, and count how many edges come out from each keyword (user activity node). From Figure 1, we see that two edges are coming out from user activity 2 and three edges are coming out from user activity 4. We follow the same process for all user activity nodes whose type is keyword. Suppose that more than one user activity has type keyword; then, we choose the product as prediction based on the maximum number of keywords (user activity) connected with the same product. In our case, user activities 2 and 4 are both connected with product node 6. It means product node 6 has count 2, while others product for example 5, 7, and 3 has count 1. In this case, we give node 6 as a product prediction, because of a bigger number of user activities with type keyword is connected with product type node 6.

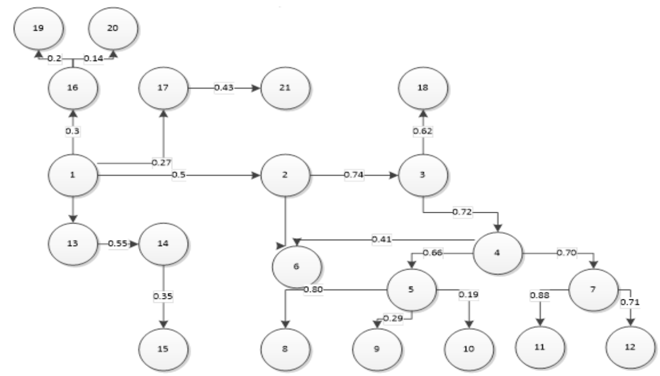

Figure 1. Subgraph in Mobile.

V. CONCLUSION

This paper proposed a novel backtracking approach of recognizing next individual's activities using individual's personal data across a plurality of users' data. This approach is better because multimap has useful utilities like invertForm, which helps to give accurate prediction in a real-time without using a supervised classification algorithm. Multimap, allows multiple values for every key, such as keyword power as a key, can map with multiple products as values, such as mobile, TV, dishwasher, etc. We can expand multimap dynamically as needed. In addition, multimap is lightweight component as it uses less memory.

Briefly described, the individual's interest disclosure pertains to personal data mining. More specifically, data mining technologies can be applied to personal user data provided by users themselves, gathered by others on their behalf and/or generated and maintained by third parties for their benefit or as required [26]. Mining of such data can enable identification of opportunities and/or provisioning of recommendations to increase user productivity and/or improve quality of life [26]. Further yet, such data can be afforded to businesses involved in market analysis, or the like, in a manner that balances privacy issues of users with demand for high quality information from businesses [26].

In accordance with an aspect of this disclosure, personal user data can be received or otherwise acquired from individual's mobile [26]. Graph techniques can be applied to the personal data across a plurality of user, for example, to identify patterns, relations and/or correlations amongst the data. Subsequently or concurrently, mining results and/or useful information based thereon can be provided to a user. We tested the recommendation engine with 25 users for a period of 6 months. We provided 25 users with Motorola G phones with voice plans and data plans and asked them to use these phones as their primary phone. We provide them recommendation every 3 hours and asked them feedback every week on the relevance of the feedback and overall experience. We provided them feedback in terms of Products, Actions and apps. There are no results yet since we are analyzing all the data and feedback collected in the last 6 months and will be producing that as another paper shortly.

REFERENCES

[1] Gomes, B., Phua, C. C. and Krishnaswamy, S. (2013), “Where Will You Go? Mobile Data Mining for Next Place Prediction,” International Conference on Data Warehousing and Knowledge Discovery - DaWaK 2013, Prague, Czech Republic.

[2] J. Gong, J. Tang “ACTPred: Activity Prediction in Mobile Social Networks,” tsinghua science and technology, issn 1007-0214 05/11 pp. 265-274, vol. 19, no. 3, June 2014.

[3] A. Papliatseyeu and O. Mayora, “Mobile Habits: Inferring and Predicting User Activities with a Location-Aware Smartphone,” 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, pp. 343-352.

[4] S. Abdullah, N. D. Lane, and T. Choudhury, “Towards population scale activity recognition: A framework for handling data diversity,” in Proc. AAAI’12, Toronto, Canada, 2012, pp. 851-857.

[5] J. Yin, Q. Yang, D. Shen, and Z. N. Li, Activity recognition via user-trace segmentation, ACM Transactions on Sensor Networks, vol. 4, no. 4, pp. 19-34, 2008.

[6] D. H. Hu and Q. Yang, Cigar: Concurrent and interleaving goal and activity recognition, in Proc. AAAI’08, Chicago, USA, 2008, pp. 1363-1368.

[7] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, Activity recognition using cell phone accelerometers, SIGKDD Explor. Newsl., vol. 12, no. 2, pp. 74-82, Mar. 2011.

[8] E. Kim, S. Helal, and D. J. Cook, Human activity recognition and pattern discovery, IEEE Pervasive Computing, vol. 9, no. 1, pp. 48 53, Jan.- Mar. 2010.

[9] Y. M. Zhang, Y. F. Zhang, E. Swears, N. Larios, Z. H. Wang, and Q. Ji, Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition, IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 10, pp. 2468-2483, Oct2013.

[10] J. H. Fowler and N. A. Christakis, Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study, British Medical Journal, vol. 337, no. a2338, pp. 1-9, Dec. 2008.

[11] J. T. Cacioppo, J. H. Fowler, and N. A. Christakis, Alone in the crowd: The structure and spread of loneliness in a large social network, Journal of Personality and Social Psychology, vol. 97, no. 6, pp. 977-991, Dec. 2009.

[12] J. Tang, et al., Quantitative study of individual emotional states in social networks, IEEE Trans. Affe. Comp., vol. 3, no. 2, pp. 132-144, April-June 2012.

[13] J. Jia, S. Wu, X. H. Wang, P. Y. Hu, L. H. Cai, and J. Tang, Can we understand van Gogh’s mood? Learning to infer affects from images in social networks, in Proc. ACM MM’13, Barcelona, Spain, 2013, pp. 857-860.

[14] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, Information diffusion through blogspace, in Proc. WWW’04, NewYork, USA, 2004, pp. 491-501.

[15] J. Kleinberg, Temporal dynamics of on-line information streams, in Data Stream Managemnt: Processing HighSpeed Data. New York, USA: Springer, 2005.

[16] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, Microscopic evolution of social networks, in Proc. KDD’08, Las Vegas, USA, 2008, pp. 462-470.

[17] P. Sarkar and A. W. Moore, Dynamic social network analysis using latent space models, ACM SIGKDD Explorations Newsletter, vol. 7, no. 2, pp. 31-40, Dec. 2005.

[18] J. Scripps, P. N. Tan, and A. H. Esfahanian, Measuring the effects of preprocessing decisions and network forces in dynamic network analysis, in Proc. KDD’09, Paris, France, 2009, pp. 747-756.

[19] A. Anagnostopoulos, R. Kumar, and M. Mahdian, Influence and correlation in social networks, in Proc. KDD’08, Las Vegas, USA, 2008, pp. 7-15.

[20] D. Kempe, J. Kleinberg, and E. Tardos, Maximizing the spread of influence through a social network, in Proc. KDD’03, Washington, DC, USA, 2003, pp. 137-146.

[21] J. Tang, J. M. Sun, C. Wang, and Z. Yang, Social influence analysis in large-scale networks, in Proc. KDD’09, Paris, France, 2009, pp. 807-816.

[22] J. Tang, S. Wu, and J. M. Sun, Confluence: Conformity influence in large social networks, in Proc. KDD’13, Chicago, IL, USA, 2013, pp. 347-355.

[23] J. Zhang, J. Tang, H. L. Zhuang, C. W. K. Leung, and J. Z. Li, Role-aware conformity influence modeling and analysis in social networks, in Proc. AAAI’14, Quebec, Canada, 2014, pp. 1-7.

[24] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, Feedback effects between similarity and social influence in online communities, in Proc. KDD’08, Las Vegas, USA, 2008, pp. 160-168.

[25] http://en.wikipedia.org/wiki/Multimap

[26] H. Johannes, “Personal Data Mining,” USPTO - US7657493, Apr 2008.

[27] https://github.com/samxhuan/textrank/tree/master/src/com/sharethis/textrank (Last access on 06/09/2015)

[28] http://sourceforge.net/projects/asmalgorithm/files/asm/ (Last accessed on 06/09/2015)

[29] http://en.wikipedia.org/wiki/Depth-first_search (Last accessed on 06/09/2015)