ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Context Aware User Interface Recommendation System

Natarajasivan [1], Govindarajan [2]

Assistant Professor, Department of Computer Science and Engineering, Annamalai University, Annamalai Nagar, Tamil Nadu, India[1]

Assistant Professor, Department of Computer Science and Engineering, Annamalai University, Annamalai Nagar, Tamil Nadu, India [2]

ABSTRACT: Mobile phones are nowadays becoming more of a computing device than a communication device. In addition to proving communication functions like making calls and sending messages, mobile phones allow a user to install an application of their choice such as social apps, gaming apps and multimedia apps. With more number of applications being installed and used the user has to place the application link on the home screen. When the application usage preference of a user changes, the home screen of the user will be changed. In this paper we propose a novel context-aware interface recommendation system that adapts to the user preference. The recommendation system uses Apriori and FP-Growth Association Rule mining techniques to identify the preference. The context-aware recommendations have been analyzed based on the user preference and its effectiveness is evaluated.

KEYWORDS: Context-aware Mobile Interface, Launcher, Association Rule Mining, Apriori, FP-Growth.

I. INTRODUCTION

The users nowadays are mobile dependent. As mobiles are becoming more powerful in terms of processing power, storage and battery life they are highly used by everyone. Their usage is more towards applications rather than communication. The basic user interface of a mobile phone consists of a home screen where the user places the frequently used applications link and application screen where all the applications installed on the mobile phone are listed. The main drawback of this approach is that the user has to manually place the application link on the home screen and when not used the link has to be manually removed by the user. This approach of home screen maintenance needs the user's time.

In recent times a special kind of application called the Launcher application are available for the android operating system. The Launcher application automatically places the most used application on the user's home screen. The Launcher application just considers the usage time of an application and whichever application that has the higher usage time is placed on the screen. This approach does not consider any contextual information regarding the applications usage such as, applications that are used at a specific time of the day or on a specific day.

The context-aware system can identify the user preference based on their usage and recommend the needed change on the home screen. In this paper we propose a recommendation system based on the usage patterns of the user. The major recommendations that are to be provided to the user are frequently used application on the home screen.

For this mobile phone usage data is collected using the Funf open sensing framework. It provides options to select what data are to be collected from the mobile phone. The main motivation behind this paper is to reduce the user's cognitive load and to increase productive usage of the mobile phone.

The remainder of the paper is structured as follows. Section II describes related work. Section III presents the Outline of the work. Section IV discussed the techniques used for rule generation. Section V elaborates the system implementation. Section VI concludes the paper.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

II. RELATED WORK

In [1] the authors have proposed a context aware mobile recommender system that learns from the user preference from the past. The recommender system is designed to recommend places to the users based on the user's mood, current weather and the time of the day. The system suggests places to the user based on the places that were visited by other users in similar contextual conditions. In this paper the authors have used Genetic Algorithm for implementing the system and used Mean Absolute Error for evaluating the results.
In [2] the authors have presented a collaborative recommender system that recommends university elective courses to students based on the courses taken by similar students. The proposed system uses association rules mining algorithm for identifying patterns between courses. K-fold cross verification with K=5 is used for implementation of the system. The performance is measured using the precision and recall.
An Agent framework for tourism recommender system is proposed in [3]. The authors have developed the framework by using the classical recommender systems techniques such as collaborative filtering and content-based filtering as agents. These agents are used to generate the recommendations. Linear combination method of data fusion is adapted by the authors to improve the performance.
In [4] a recommender system is presented by the authors that identify relatedness of concepts form digital publishing resources. The authors have adopted a two stage process. In the first stage concepts are extracted from the encyclopedias. In the second stage concept vectors are generated by skip-gram model. From the concept vectors semantic relatedness between the concepts are measured. Using the semantic relatedness information the recommender system recommends the users learning or reading.
In [5] the authors have reviewed various recommender system algorithms that are utilized in social networks in e-Learning systems. The recommender system helps students to choose from different learning objects to study. The authors have reviewed algorithms such s C4.5, K-Means, Support Vector Machine and Apriori.
A hybrid recommender system based on user-recommender interactions is proposed in [6]. The authors have proposed a three stage process. In the first stage, the user-recommender interaction is defined. The recommender system gives its recommendation to the user upon request. The user will continue using the recommender system if it provides recommendation that the user has interest. This process will continue until the recommender system provides recommendation that does not favors the user. In the second stage, the authors propose a hybrid recommender system combining random and k-nearest neighbor algorithms. In the third stage, the recommender system is evaluated using recall and diversity metrics. The results show that the hybrid algorithm outperforms non hybrid algorithm.
In [7] the authors have proposed an intelligent service recommendation model. The service adaptation process in formulated using artificial intelligence techniques such as Bayesian Network, fuzzy logic and rule based reasoning. The incoming call to the users mobile is classified as high priority call, low priority call and unknown call using Bayesian Network. Context situations are defined using the fuzzy linguistic and membership degrees. Rules are used for implementing a service. The context aware mobile is tested for library and class room scenario. The system service and recommendation precision of 79% is achieved.
SocialFusion is presented in [8], a context-aware system for combining mobile, social and sensor data to effectively implement a context-aware recommendation both for individual and for groups of people. The authors have outlined a multistage architecture for achieving fusion and described some of the major challenges encountered.
In [18] the authors have surveyed the recent developments in context-aware mobile recommendations. The authors have discussed about mobile context and how contextual information is collected. The existing work on context-aware modeling of mobile recommendation and mobile recommender application has been studied. Recommender applications in social networking, tourism and urban computing have been studied. The study also focuses on critical challenges in mobile recommendation system such as privacy problem, the energy-efficiency concern and the design of the user interface.
A summarized and comparative study of the available Association Rule Mining algorithms is done in [15]. Algorithms such as Weighted Apriori, Revised FP-Growth Frequent Pattern Growth and fuzzy Apriori are studied. The experiment results show that the weighted Apriori yields better results when compared to other algorithms.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

III. OUTLINE OF THE WORK

The outline of the work is depicted in Fig. 1. The works starts with collection of data from the mobile phone. The data is collected with the use of Funf Journal mobile application. Collected data needed to be tailored for the recommender system. In the next step the data is processed so that it can be used for generating rules. Data processing such as de-duplication, transformation and replacing missing values are performed. The processed data is carried forward to the next step. Using Apriori and FP-Growth algorithms rules are generated. In the final step the generated rules are given as recommendations to the user.

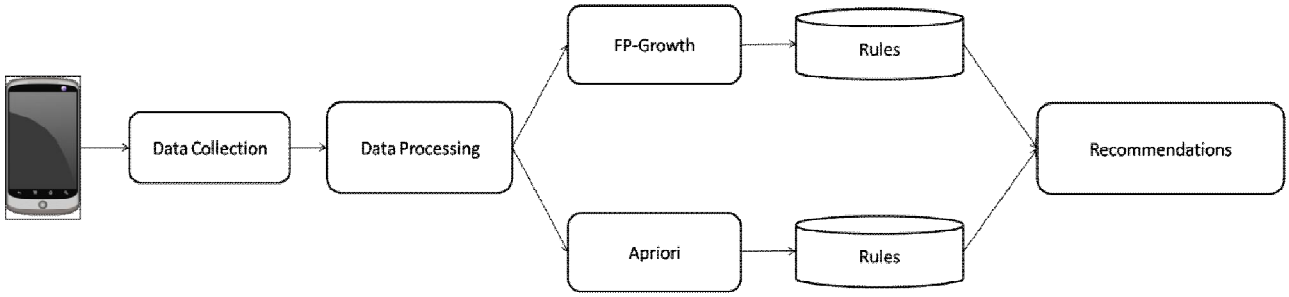

Fig. 1 Outline of the Work

IV. TECHNIQUES

In this work Apriori and FP-Growth Association Rule Mining algorithms are used to generate rules for the recommender system. The processed data is supplied to the algorithms that generate the rules. Association Rule [19] is the process of finding hidden pattern from large datasets. The aim of Association Rule Mining is to extract frequent patterns, interesting correlations or association among transaction or other repositories. Let I= $I_1, I_2, \ldots, I_n$ be a set of n distinct attributes, T be transaction that contains a set of items such that T $\subseteq$ I, D be a database with different transaction records T_s. An association rule is an implication in the form of X⇒Y, where X, Y⊂ I are sets of items called itemsets, and X ∩ Y = ∅. X is called antecedent while Y is called consequent, the rule means X implies Y.

The association rules are generated based on the measures *support(s)* and *confidence(c)*. The association rules generated for a large database is large to focus on only the frequent items that are more important the support and confidence values are provided by the user to remove the rules that are of less importance. The threshold values are called as minimal support and minimal confidence.

In association rule mining an association rules ***Support(s)*** is defined as the percentage/fraction of records that contain X ∪ Y to the total number of records in the database. The count for each item is increased by one every time the item is encountered in different transaction T in database D during the scanning process. Support(s) is calculated by the following equation

$$Support(XY) = \frac{Support\ count\ of\ XY}{Total\ number\ of\ trasaction\ in\ D}$$

The support of an item is a statistical significance of an association rule. If the support of an item is 0.1% this means that only 0.1 percent of the transaction contains that particular item.

In association rule mining an association rules ***Confidence*** is defined as the percentage/fraction of the number of transactions that contain X ∪ Y to the total number of records that contain X, where if the percentage exceeds the threshold of confidence an interesting association rule X⇒Y can be generated.

$$Confidence(X|Y) = \frac{Support(XY)}{Support(X)}$$

Confidence is a measure of strength of the association rules, if the confidence of the association rule X⇒Y is 85%, then 85% of the transactions that contain X also contain Y .

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Apriori:

Apriori algorithm is an improved algorithm over the AIS algorithm that makes many passes over the database for candidate itemsets generation. The principle that is adopted in Apriori algorithm is *"If an itemset is frequent, then all of its subsets must also be frequent"*. This idea is illustrated using the itemset lattice shown in Fig. 2.

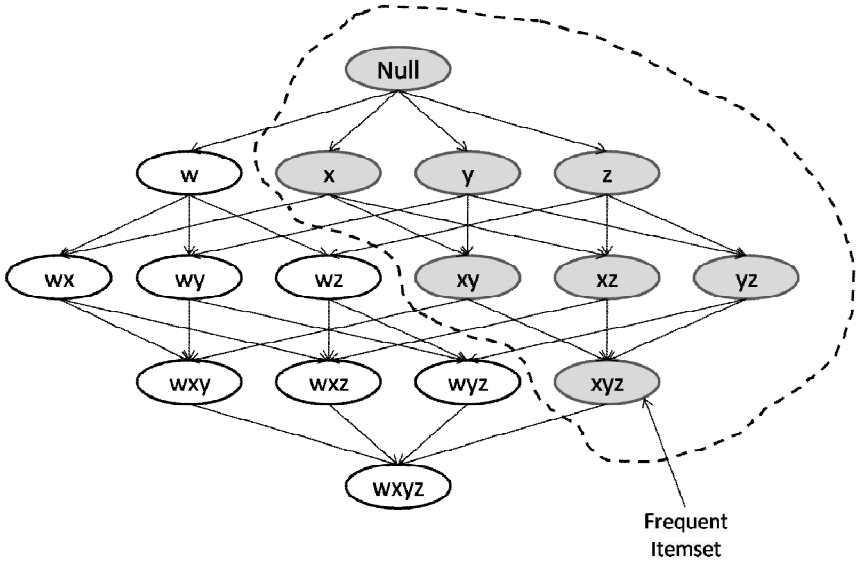

Fig. 2 An Illustration of Apriori Principle. *If {x,y,z}* is frequent, then all subsets of this items are frequent.

Suppose {*x*, *y*, *z*} is a frequent itemset. Then any transaction that contains {*x*, *y*, *z*} must also contain its subsets, {*x*, *y*}, {*x*, *z*}, {*y*, *z*}, {*x*}, {*y*}, and {*z*}. This rules states that if {*x*, *y*, *z*} is frequent, then all subsets of {*x*, *y*, *z*} (i.e., the itemsets that are shaded in Fig. 2) must also be frequent.

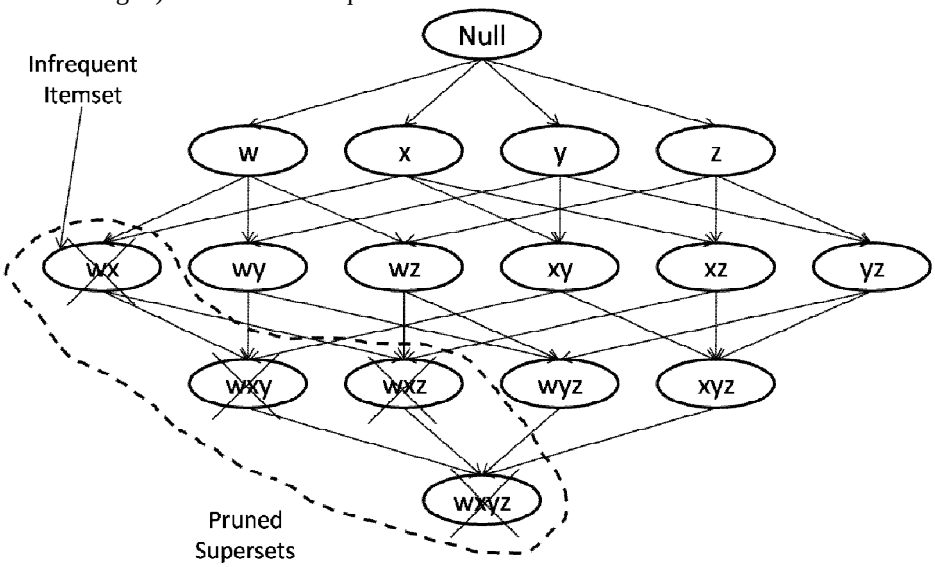

Fig. 3 An illustration of support-based pruning. If {*w*, *x*} is infrequent, then all supersets of {*w*, *x*} are infrequent.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Conversely, the algorithm suggests that if an itemset such as {*w*, *x*} is infrequent, then all of its supersets must be infrequent too. This concept is illustrated in Fig. 3 in which the superset of {*w*, *x*} is pruned immediately once it is found to be infrequent. This strategy of pruning the search space based on the support measure is known as support-based pruning.

FP-Growth:

The Apriori algorithm has two major factors that affect the performance of the algorithm. One is the complex candidate generation process that uses most of the time, space and memory. Another one is the multiple scan of the database. These factors of the Apriori algorithm were avoided in the FP-Growth algorithm. In FP-Growth [21] algorithm the frequent itemsets are generated with only two passes over the database and without any candidate generation process. The FP-Growth algorithm is faster that Apriori algorithm. In this algorithm the data set is encoded into a compact data structure called an FP-tree and the frequent itemsets are directly extracted from this structure.

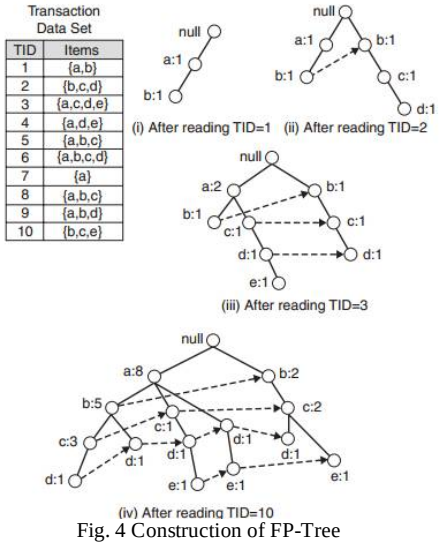

Fig. 4 Construction of FP-Tree

The process of constructing FP-Tree is given below.

1. The dataset is scanned once to determine the support count of each item. Infrequent items are discarded, while the frequent items are sorted in decreasing support count. For the data set shown in Fig. 4, a is the most frequent item, followed by b, c, d, and e.
2. FP-Tree is constructed by making another pass over the data. The nodes labeled as a and b are generated after reading the first transaction {a, b}. Frequency count is set as 1 for every node along the tree path.
3. A new set of nodes are created for the items b, c, and d after reading the second transaction {b, c, d}. Frequency count is set as 1 for every node along the path. Both the transactions are disjoint as they don't share a common prefix.
4. A common prefix is shared with the first transaction and the third transaction. As a result, the path of the third transaction overlaps with the path for the first transaction. This overlapping increases the frequency count of the node a by one resulting the frequency count of node *a* as two.
5. The above said process is continued until every transaction in the data has been mapped to one of the paths given in the FP-Tree. The resulting FP-Tree after reading all the transactions is shown at the bottom of Fig. 4.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Once the FP-Tree is generated the FP-Growth algorithm generates frequent itemsets by exploring the tree in a bottom-up fashion. For the FP-Tree shown in Fig. 4 the algorithm looks for frequent itemsets ending in e first, followed by d, c, b, and finally a. The tree paths ending with the above said items are shown in Fig. 5.

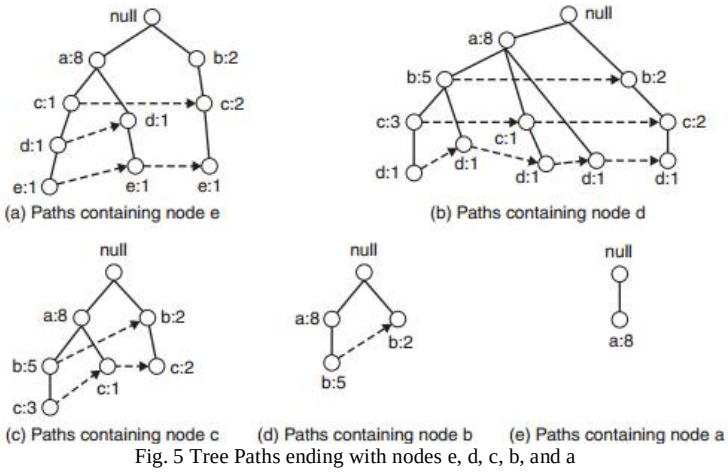

Fig. 5 Tree Paths ending with nodes e, d, c, b, and a

FP-Growth algorithm further decomposes each of these into smaller subproblems. In the subproblem the algorithm finds prefix paths and generates a conditional FP-Tree from the prefix paths. This process of finding prefix path and generating conditional FP-Tree is repeated until there is minimal support for the prefix path.

V. SYSTEM IMPLEMENTATION

The recommender system is implemented by collecting data from the mobile phone. The collected data are preprocessed making it usable by the mining algorithms. Rule generation was done using Apriori and FP-Growth algorithms. The generated rules are recommended to the user.

Dataset:

The dataset for the recommendation system is collected using Funf Journal Application (Android). Usage data of 5 participants is collected using the mobile application. Funf Journal is an Android application built using the Funf framework. The application allows the user to configure data collection parameters from over 30 different built-in data probes. The probes consist of all phone sensors, as well as additional data types.

Table. 1 Important Probes

Probe Name	Description
WifiProbe	Takes a snapshot of the Wifi hotspots available
LocationProbe	Uses cell and Wifi networks alogn with GPS to compute current location
ActivityProbe	Uses the accelerometer to estimate how active the user is
BluetoothProbe	Takes a snapshot of nearby Bluetooth devices
ScreenProbe	Records when the screen is turned on and off
BatteryProbe	Records the current state of the battery
ApplicationsProbe	Takes a snapshot of the apps that are installed
RunningApplicationsProbe	Takes a snapshot of the apps that are currently running
ContactProbe	Takes a snapshot of all the available contacts in the mobile

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

Table. 1 gives the list of some of the important probes that are available in the Funf Journal Mobile Application. The data collected from the selected probes can be stored in any of the following ways-by manually exporting the data via email or any other Android service that support file transfer, by manually copying it out of the devices memory card.

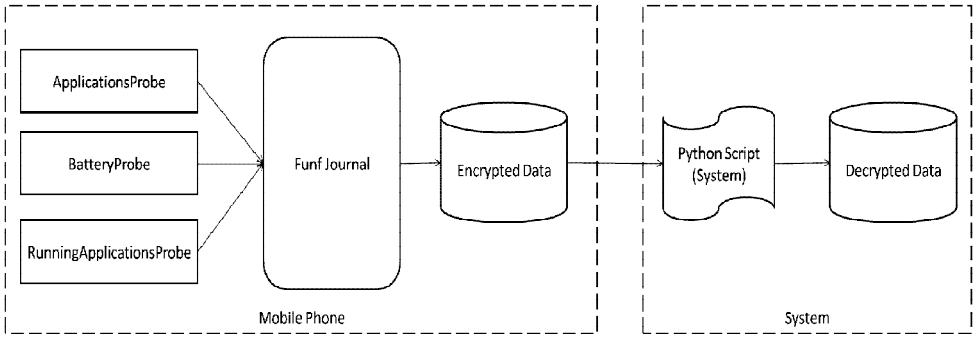

Fig. 6 Data Collection Process

The data collection process for this work is depicted in Fig. 6. The dataset used in this work is collected from the ApplicationsProbe, BatteryProbe and RunningApplicationsProbe which is encrypted and stored on the mobile phone. The data is manually exported out of the devices memory card. With the use of python script the encrypted data is decrypted on the system.

Data Preprocessing:

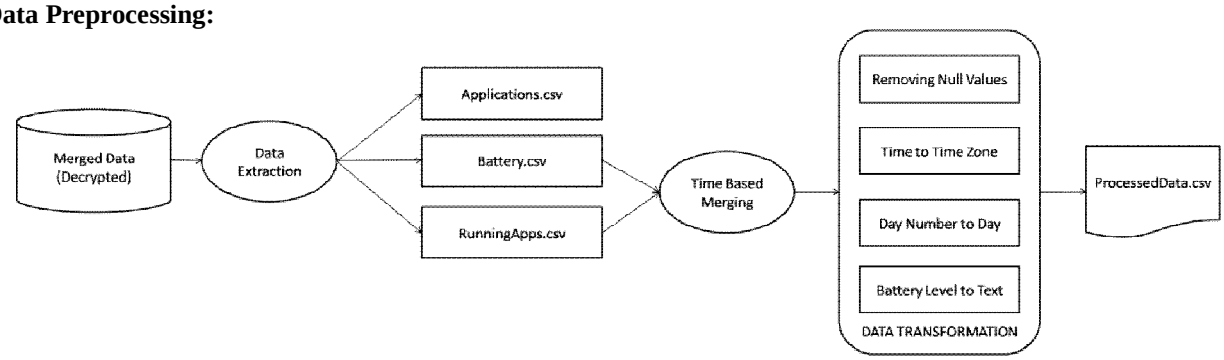

Fig. 7 Data Preprocessing

The data preprocessing done in this work is shown in Fig. 7. The decrypted data is stored in SQLite database. The data from the database is extracted using SQL query for each day and probe. As a result of this extraction process the data from database is stored as separate CSV file for each probe grouped by date. Each probe file will consist of the time the data is collected and the value for that corresponding probe. Data collected from the BatteryProbe is shown in Table. 2.

Table. 2 Data form BatteryProbe

Date	Time	Day	Charge_Rate	Health	Level	Plugged	Status	Temperature	Voltage
3/25/2016	15:20	5	0	2	98	0	3	346	4140
3/25/2016	15:21	5	0	2	98	0	3	346	4140
3/25/2016	15:22	5	0	2	98	0	3	348	4154
3/25/2016	15:23	5	0	2	98	0	3	339	4214

The BatteryProbe and RunningApplicationsProbe data are combined to generate the data for recommender system. The data are merged based on the date and time values. The date and time value of each probe is matched and the data are

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

merged. Once both the probe values are merged, the missing value in the final data is handled using the Last Observation Carried Forward (LOCF) technique. LOCF method is selected for this work as replacing missing values with average or random value is not suitable for this problem domain. After replacing the missing values in the data, the data is transformed so that Association Rule Mining algorithms can be implemented.

Time	TimeZone
06:00 AM to 11:59 AM	Morning
12:00 PM to 16:59 PM	Afternoon
17:00 PM to 20:59 PM	Evening
21:00 PM to 05:59 AM	Night

(a)

Day Number	Day
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

(b)

Battery Level Value	Text
0 to 25	Low
26 to 75	Average
76 to 100	Full

(c)

Fig. 8 Data Transformation (a) Time to TimeZone, (b) Day Number to Day and (c) Battery Level Value to Text

The major transformations that are performed are converting the actual time into time zone, converting day number to day and battery level value to text as shown in Fig. 8. The time to time zone conversion is done by converting the time 06.00 am to 11.59 am as morning, 12.00 pm to 16.59 pm as afternoon, 17.00 pm to 20.59 pm as evening and 21.00 pm to 05.59 am as night. The days numbered from 0 to 6 are converted to Sunday to Saturday respectively. The battery level values are transformed as 0 to 25% as Low, 26 to 75% as Average and 76 to 100% as Full.

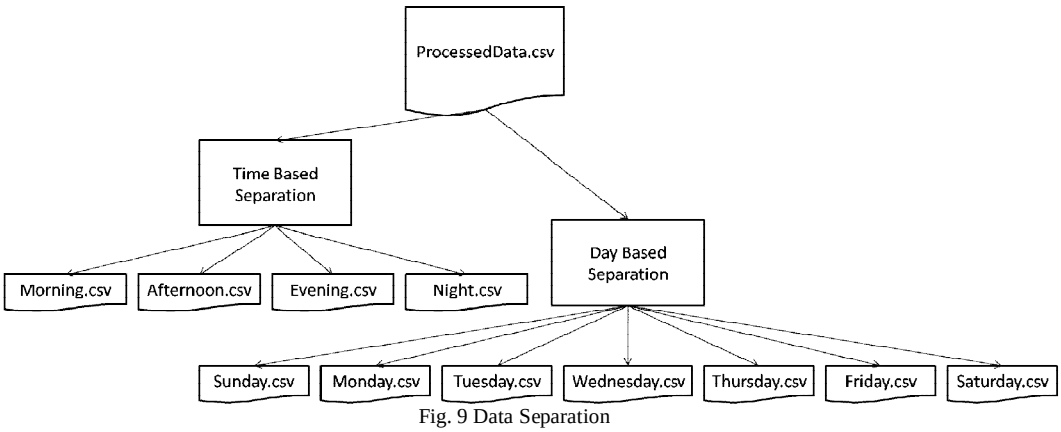

Fig. 9 Data Separation

The dataset is separated into sub-datasets for the recommender system based on time and day. In time based separation the data is separated into Morning, Afternoon, Evening and Night data. In day based separation the data is separated into Sunday, Monday, Tuesday, Wednesday, Thursday, Friday and Saturday data as shown in Fig. 9.

Evaluation using Apriori Algorithm:

The Apriori algorithm is implemented on the separated dataset. The minimum support and minimum confidence values of 0.001 and 0.2 are used respectively. Data collected from each participant is separated based on day and time. The separated data is implemented in the algorithm for rule generation.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

S.No	Rules	Support	Confidence	Lift
1	{Time=Morning,Battery=Full} => {App=MX Player}	0.10	0.475	2.27
2	{Time=Morning,Battery=Average} => {App=WhatsApp}	0.03	0.272	1.47
3	{Time=Night,Battery=Average} => {App=Opera Mini beta}	0.03	0.327	1.85
4	{Time=Evening,Battery=Average} => {App=WhatsApp}	0.03	0.246	1.33
5	{Time=Evening,Battery=Average} => {App=Opera Mini beta}	0.03	0.227	1.29
6	{Time=Afternoon,Battery=Low} => {App=Phone}	0.03	0.279	2.34
7	{Time=Night,Battery=Average} => {App=WhatsApp}	0.03	0.275	1.49
8	{Time=Night,Battery=Average} => {App=MX Player}	0.02	0.241	1.15
9	{Time=Evening,Battery=Low} => {App=MX Player}	0.02	0.434	2.07
10	{Time=Afternoon,Battery=Full} => {App=Chrome}	0.01	0.348	2.76

(a)

S.No.	Rules	Support	Confidence	Lift
1	{Time=Morning,Battery=Average} => {App=Chrome}	0.07	0.28	1.16
2	{Time=Morning,Battery=Average} => {App=Opera Mini beta}	0.06	0.24	1.54
3	{Time=Morning,Battery=Full} => {App=Chrome}	0.05	0.27	1.12
4	{Time=Night,Battery=Average} => {App=MX Player}	0.05	0.51	3.00
5	{Time=Afternoon,Battery=Average} => {App=Chrome}	0.04	0.27	1.10
6	{Time=Evening,Battery=Average} => {App=MX Player}	0.04	0.42	2.50
7	{Time=Afternoon,Battery=Average} => {App=YouTube}	0.03	0.21	4.86
8	{Time=Evening,Battery=Low} => {App=Chrome}	0.03	0.23	0.95
9	{Time=Evening,Battery=Low} => {App=MX Player}	0.02	0.22	1.32
10	{Time=Night,Battery=Low} => {App=Chrome}	0.02	0.44	1.83

(b)

S.No	Rules	Support	Confidence	Lift
1	{Time=Morning,Battery=Full} => {App=Opera Mini beta}	0.09	0.21	1.28
2	{Time=Morning,Battery=Low} => {App=Opera Mini beta}	0.01	0.36	2.26

(c)

S.No	Rules	Support	Confidence	Lift
1	{Time=Afternoon,Battery=Average} => {App=Phone}	0.15	0.21	1.03
2	{Time=Afternoon,Battery=Average} => {App=Chrome}	0.15	0.20	1.04
3	{Time=Afternoon,Battery=Low} => {App=Opera Mini beta}	0.05	0.25	1.76
4	{Time=Afternoon,Battery=Low} => {App=Phone}	0.04	0.21	1.01
5	{Time=Afternoon,Battery=Full} => {App=Chrome}	0.02	0.39	2.02

(d)

Fig. 10 Rules Generated for a single Participant (a) Sunday Data (b) Monday Data (c) Morning Data (d) Afternoon Data

The rules generated for a single participant data by day based, i.e. Sunday and Monday, and time based, i.e. Morning and Afternoon is shown in Fig 10. It is observed that the application usage preference differs based on day, time and battery level. In Fig. 10(a) the rule number 1 implies that on Sunday morning with full battery level MX Player application is used, in Fig. 10(b) the rule number 3 states that on Monday morning with full battery level Chrome application is used.

Evaluation using FP-Growth Algorithm:

The FP-Growth algorithm with minimum support value of 0.2 is implemented. The extracted and processed data of each participant is given as input to the algorithm. Similar to that of the Apriori rule generation the data is separated based on day and time in FP-Growth rule generation also.

S.No	Rules	Support
1	{Average, Afternoon} => {Phone}	0.09
2	{Average, Morning} => {Chrome}	0.07
3	{Average, Morning} => {MX Player}	0.05

(a)

S.No	Rules	Support
1	{Average, Morning} => {Chrome}	0.07
2	{Average, Morning} => {Opera Mini beta}	0.06
3	{Morning, Full} => {Chrome}	0.05
4	{Average, Night} => {MX Player}	0.05

(b)

S.No	Rules	Support
1	{Morning, Low} => {Chrome}	0.07
2	{Morning, Low} => {Opera Mini beta}	0.06

(c)

S.No	Rules	Support
1	{Afternoon, Average} => {Phone}	0.16
2	{Afternoon, Average} => {Opera Mini beta}	0.15
3	{Afternoon, Average} => {Chrome}	0.15
4	{Afternoon, Average} => {WhatsApp}	0.10
5	{Afternoon, Low} => {Opera Mini beta}	0.08

(d)

Fig. 11 Rules Generated for a single participant (a) Sunday Data (b) Monday Data (c) Morning Data (d) Afternoon Data

The rules generated by the FP-Growth algorithm are shown in Fig. 11. It is observed that form Fig. 11 (c) and (d) the most used application during morning is Chrome and in the afternoon is Phone.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

VI. CONCLUSION

In this work, researchers have proposed a context-aware user interface recommender system for home screen applications link placement. The application usage data collected from the participants are separated based on day and time for rule generation. The rules are generated using Apriori and FP-Growth algorithms. The existing Launcher application considers only the application usage time for recommendation.

S.No.	Applications	Usage Time (in Seconds)
1	Chrome	120617
2	MX Player	92696
3	Opera Mini beta	83829
4	WhatsApp	81393
5	Phone	66591
6	Adobe Acrobat	21733
7	Flipkart	21216
8	YouTube	14041
9	Amazon Shopping	12996
10	File Manager	10364

Fig. 12 Application Preference based on usage time

The application preference based on the usage time is shown in Fig. 12. When this is compared with the rules generated by the recommender system as shown in Fig. 10 and Fig. 11, it is observed that application recommendation made by the proposed system is based on the addition contextual information. From Fig. 10 it is observed that on Sunday during Morning with Full battery the most used application is MX Player. Similarly in Morning with Full Battery Level Opera Mini Beta application is used on all days. This kind of recommendation is not made by the existing system.

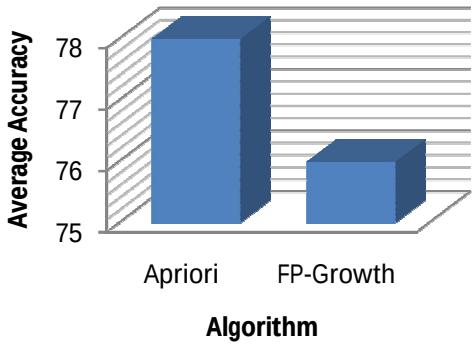

Fig. 13 Evaluation Results

The recommendations are evaluated by reviewing the participants about the recommendations made. The evaluation results are shown in Fig. 13. It is observed that the recommendations made by the Apriori algorithm are more efficient with 78% accuracy than the recommendations made by FP-Growth algorithm with 76% accuracy.

REFERENCES

[1] SohaA.El-Moemen Mohamed, Taysir Hassan A.Soliman, Adel A.Sewisy, "A Context-Aware Recommender System for Personalized Places in Mobile Applications", International Journal of Advanced Computer Science and Applications, Vol.7, Issue.3, pp.442-448, 2016.

[2] Amer Al-Badarenah, Jamal Alsakran, "An Automated Recommender System for Course Selection", International Journal of Advanced Computer Science and Applications, Vol.7, Issue.3, pp.1166-175, 2016.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 5, May 2016

[3] JiaZhi Yang, Gao Wei, Shi Yi Jin, "An Agent Framework of Tourism Recommender System", MATEC Web of Conferences 44, pp.01005.1–01005.5, 2016.
[4] Mao Ye, Zhi Tang, JianboXu, Lifeng Jin, "Recommender System for E-Learning Based on Semantic Relatedness of Concepts", Information, Vol.6, pp.443-453, 2015.
[5] Salama, A. A., Mohamed Eisa, ELhafeez, S. A., Lotfy, M. M., "Review of Recommender Systems Algorithms Utilized in Social Networks based e-Learning Systems & Neutrosophic System", Neutrosophic Sets and Systems, Vol.8, pp.32-41, 2015.
[6] Heng-Ru Zhang, Fan Min, Xu He, Yuan-Yuan Xu, "A Hybrid Recommender System Based on User-Recommender Interaction", Mathematical Problems in Engineering, Vol.2015, pp 145636.1-145636.11, 2015.
[7] Thyagaraju, G. S., Umakant P Kulkarni, "Design and Implementation of User Context aware Recommendation Engine for Mobile using Bayesian Network, Fuzzy Logic and Rule Base", International Journal of Computer Applications, Volume 40– No.3, pp.47-63, February 2012.
[8] Aaron Beach, Mike Gartrell, Xinyu Xing, Richard Han, Qin Lv, Shivakant Mishra, Karim Seada, "Fusing Mobile, Sensor, and Social Data To Fully Enable Context-Aware Computing", HOTMOBILE 2010 Proceedings of the Eleventh Workshop on Mobile Computing Systems & Applications, pp.60-65, 2010.
[9] Feng Xia, Haifeng Liu, Ivan Lee, Longbing Cao, "Scientific Article Recommendation: Exploiting Common Author Relations and Historical Preferences", IEEE Transactions on Big Data, Vol.PP , 99, pp.1-1, 2016.
[10] Agus Sasmito Aribowo, Nur Heri Cahyana, "Feasibility study for banking loan using association rule mining classifier", International Journal of Advances in Intelligent Informatics, Vol.1, Issue.1, pp.41-47, 2015.
[11] Sathiyapriya, K., Sudha Sadasivam, G., "A Survey on Privacy Preserving Association Rule Mining", International Journal of Data Mining & Knowledge Management Process, Vol.3, Issue.2, pp.119-131, 2013.
[12] Vidisha H. Zodape, Leena H. Patil, "Literature Survey On Formation Of Association Rule Using Secure Mining", International Journal of Scientific & Technology Research, Vol.4, Issue,1, pp.222-224, 2015.
[13] Karthikeyan, T., Ravikumar, N., "A Survey on Association Rule Mining", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 3, Issue 1, pp.523-527, January 2014.
[14] Yi Zeng, Shiqun Yin, Jiangyue Liu, Miao Zhang, "Research of Improved FP-Growth Algorithm in Association Rules Mining", Scientific Programming, Vol.2015, pp.910281.1-910281.6, 2015.
[15] Arthi Priadharsni, A., Ramaraj, E., "Comparative Study of Revised FP – Growth, Weighted Apriori and Fuzzy Apriori Algorithm", International Journal of Engineering Trends and Technology, Vol.13, Issue.6, pp.261-265, 2014.
[16] Mansi Sood, Harmeet Kaur, "Preference Based Personalized News Recommender System", International Journal of Advanced Computer Research, Vol.4, Issue.15, pp.575-581, 2014
[17] Wenxing Hong, Siting Zheng, Huan Wang, Jianchao Shi, "A Job Recommender System Based on User Clustering", Journal of Computers, Vol.8, Issue.8, pp.1960-1967, 2013.
[18] Qi Liu, Haiping Ma, Enhong Chen, HuiXiong, "A Survey of Context-Aware Mobile Recommendations", International Journal of Information Technology & Decision Making, Vol. 12, Issue. 1, pp.139-172, 2013.
[19] Agrawal, R., Imielinski, T., and Swami, "Mining association rules between sets of items in large databases". In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp.207-216, 1993.
[20] Agrawal, R. and Srikant, R., "Fast algorithms for mining association rules". In: Proc. 20th Int. Conf. Very Large Data Bases, pp.487-499, 1994.
[21] Jiawei Han, Jian Pei, and Yiwen Yin, "Mining Frequent Patterns without Candidate Generation", In 2000 ACM SIGMOD Intl. Conference on Management of Data, Vol.29, Issue.2, pp.1-12, 2000.