Ubiquitous and Mobile Computing CS 528
MobileMiner: Mining Your Frequent Patterns on Your Phone

Xiaoyan Sun

Mei Yang

Computer Science Dept.

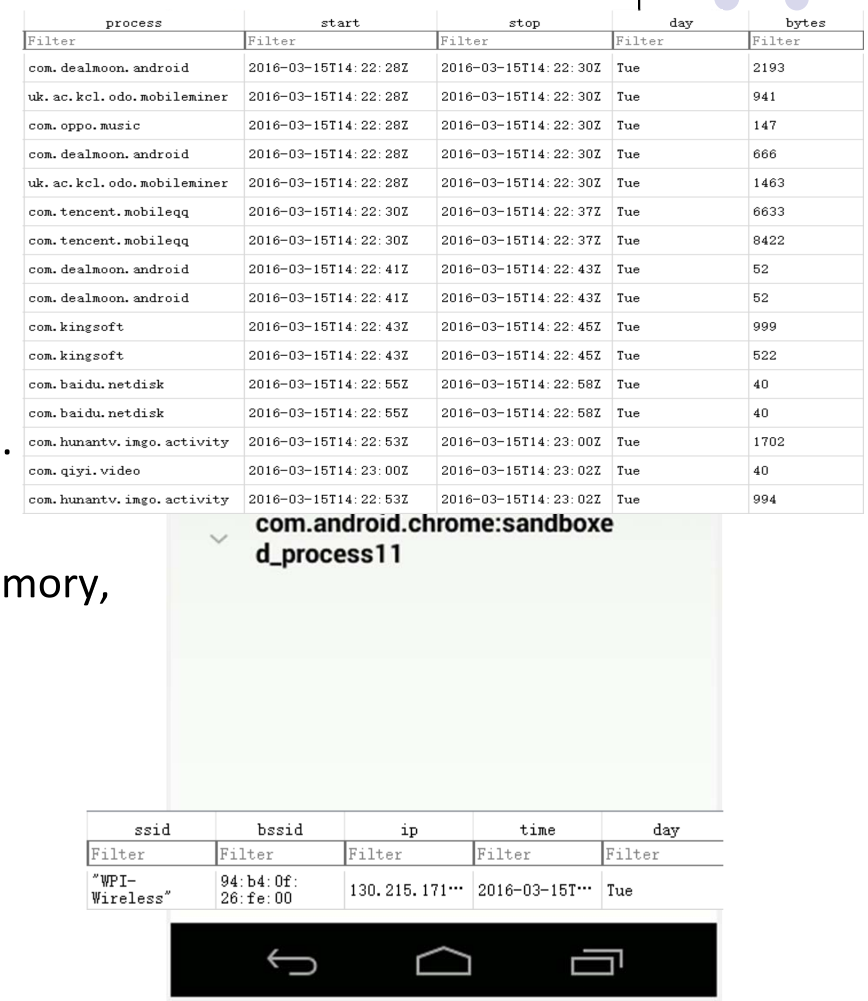
Worcester Polytechnic Institute (WPI)

Motivation:

- Long term Goal:
  - Develop novel middleware and algorithms
  - Mine user behavior patterns entirely on the phone
  - Utilize idle processor cycles
- Accomplished:
  - A novel general-purpose service: MobileMiner
    - http://kingsbsd.github.io/MobileMiner/
  - Discover frequent co-occurrence patterns on the phone
  - The patterns can be used by developers to improve UI

Overview of MobileMiner

- Currently it logs:
  - IDs of the GSM cells you visit.
  - Mobile networks that provide mobile data.
  - Names and BSSIDs of wifi hot-spots.
  - Processes that open network sockets.
  - Socket IP addresses and ports.
  - When 'net-enabled apps send notificationss.
- Export data:
  - Download directory of the device's flash memory, or SD card if it has one



process	start	stop	day	bytes
Filter	Filter	Filter	Filter	Filter
com.dealmoon.android	2016-03-15T14:22:28Z	2016-03-15T14:22:30Z	Tue	2193
uk.ac.kcl.odo.mobileminer	2016-03-15T14:22:28Z	2016-03-15T14:22:30Z	Tue	941
com.oppo.music	2016-03-15T14:22:28Z	2016-03-15T14:22:30Z	Tue	147
com.dealmoon.android	2016-03-15T14:22:28Z	2016-03-15T14:22:30Z	Tue	666
uk.ac.kcl.odo.mobileminer	2016-03-15T14:22:28Z	2016-03-15T14:22:30Z	Tue	1463
com.tencent.mobileqq	2016-03-15T14:22:30Z	2016-03-15T14:22:37Z	Tue	6633
com.tencent.mobileqq	2016-03-15T14:22:30Z	2016-03-15T14:22:37Z	Tue	8422
com.dealmoon.android	2016-03-15T14:22:41Z	2016-03-15T14:22:43Z	Tue	52
com.dealmoon.android	2016-03-15T14:22:41Z	2016-03-15T14:22:43Z	Tue	52
com.kingsoft	2016-03-15T14:22:43Z	2016-03-15T14:22:45Z	Tue	999
com.kingsoft	2016-03-15T14:22:43Z	2016-03-15T14:22:45Z	Tue	522
com.baidu.netdisk	2016-03-15T14:22:55Z	2016-03-15T14:22:58Z	Tue	40
com.baidu.netdisk	2016-03-15T14:22:55Z	2016-03-15T14:22:58Z	Tue	40
com.hunantv.imgo.activity	2016-03-15T14:22:53Z	2016-03-15T14:23:00Z	Tue	1702
com.qiyi.video	2016-03-15T14:23:00Z	2016-03-15T14:23:02Z	Tue	40
com.hunantv.imgo.activity	2016-03-15T14:22:53Z	2016-03-15T14:23:02Z	Tue	994

ssid	bssid	ip	time	day
Filter	Filter	Filter	Filter	Filter
"WPI-Wireless"	94:b4:0f:26:fe:00	130.215.171…	2016-03-15T…	Tue

Introduction:

- Why Mine Co-occurrence Patterns:
  $\{Morning, Breakfast, AtHome\} \rightarrow \{ReadNews\}$
  - Preload news ahead
  - Intuitive UI
  - Send a smart reminder to charge the phone

- Why Mine Patterns on the Mobile Device:
  - Mining algorithms can run during idle time(sleeping)
    - Charging
    - At least 80% battery
  - Better privacy guarantees
  - Benefits to users with lower-end phones

- Main Idea:
  - Log raw contextual data
    - Orthogonal and longitudinal
  - Higher level user context to behavior patterns
  - Pattern mining entirely on device

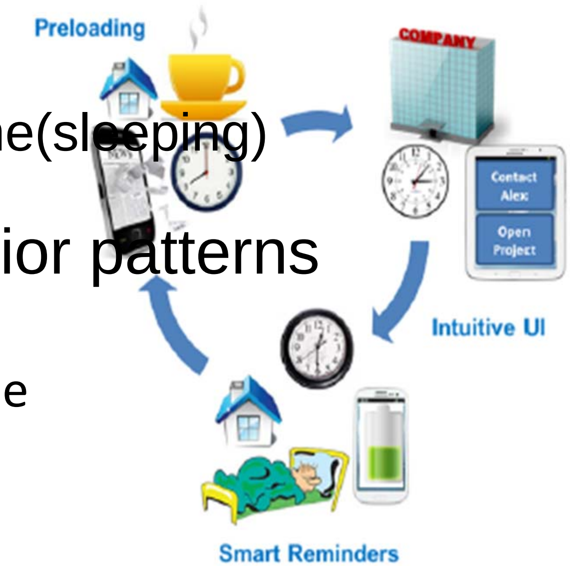


Figure 1. Example co-occurrence patterns and their uses.

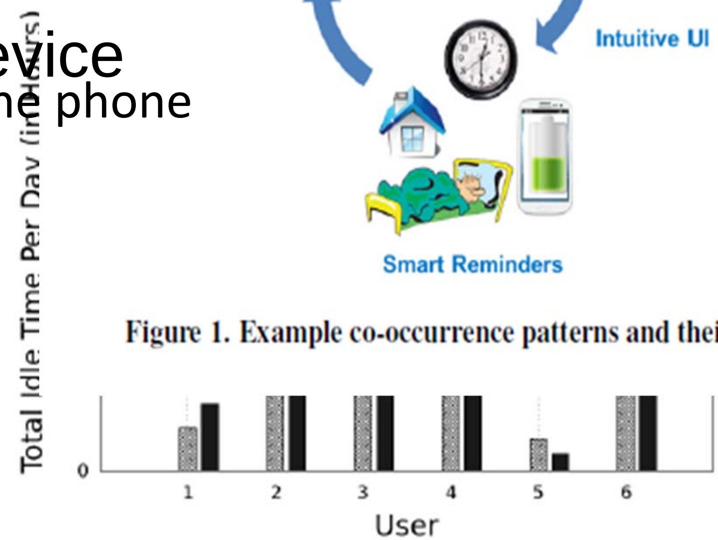


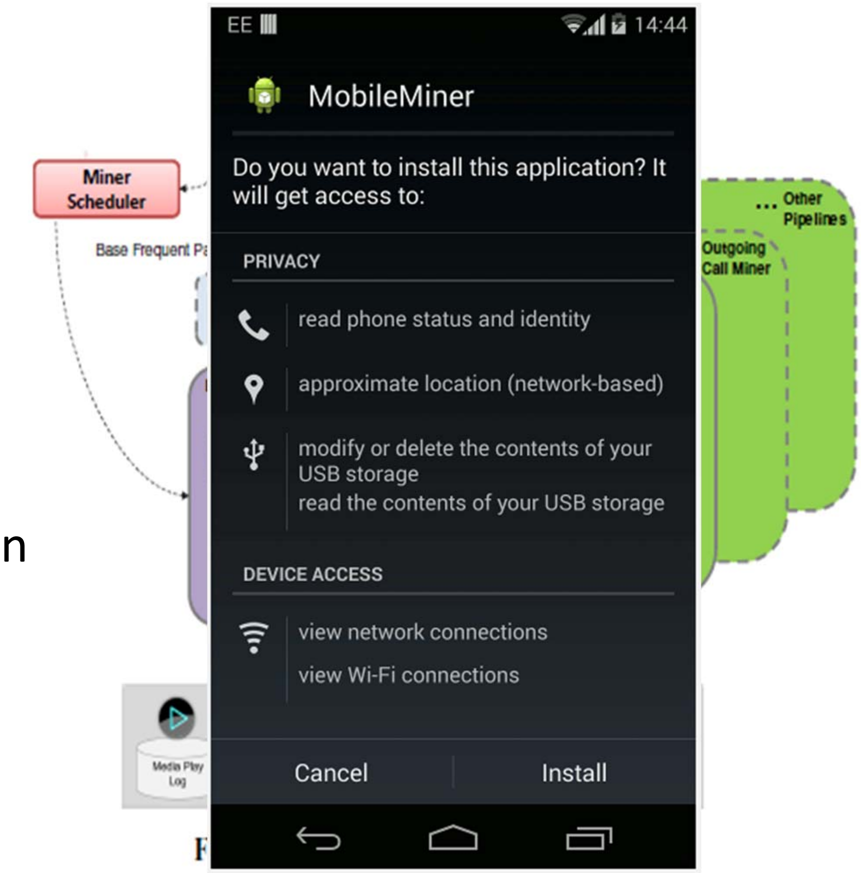
Figure 2. Average smartphone idle time per day for six sample users.

Contributions:

- System Design:
  - Use limited phone resources
  - Provide a prediction engine
- Performance:
  - Feasibility of experimental running by using context logs
    - 106 users over 1-3 months
    - Faster than widely used Apriori algrithm
    - Low consumes (0.01-3% battery)
- Pattern mining:
  - Analyze patterns individually as well as groups
- UI improvement:
  - Predict next outgoing call or app, and provide shortcut icons for them

System Design:

- Data collection:
  - User activity logs (few permissions)
- Base pattern miner:
  - Base Basket extractor
  - Base rule miner
- App usage miner:
  - Filter use threshold
  - App rule miner
  - App pattern retriever: prediction
- Minner scheduler



Basket Extraction and Filtering

- Basket Extraction
  - Continuous context to a set of small discrete values.
    - Eg: locations, battery
  - Timestamped context baskets based on temporal overlap
  - Compress duplicate baskets

- Basket Filtering
  - Boolean expression
  - Utility functions
  - Benefits:
    - More accurate
    - Faster
    - Free of noise

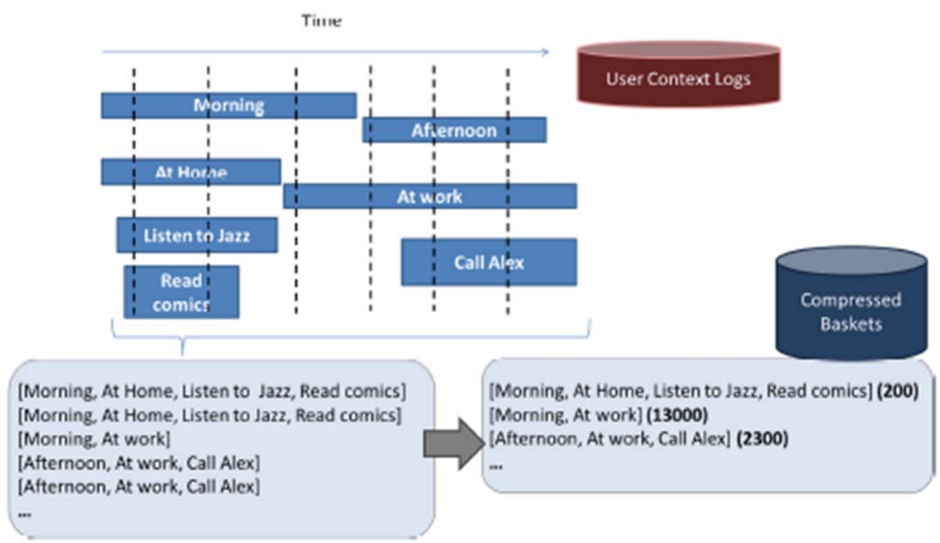


Figure 4. Compressed Basket Extraction.

Weighted Rule Mining

- Input : weighted context baskets
- Output:
  - Association rules: $\underset{\text{antecedent}}{A} \rightarrow \underset{\text{consequent}}{B}$ $\{AtHome, UsingWiFi, 10-11pm\} \rightarrow \{ChargingPhone\}$
  - Threshold:
    - Support P(A,B)
      - Long duration activities: high
      - Short duration activities: low
    - Confidence P(B|A)
- Generate frequent itemsets:
  - Occur many times as support threshold
  - Confidence exceeds confidence threshold
  - Help predict

Challenge:

Lower support value

↓

Increasing running time

Apriori VS WeMiT

- *Optimized Apriori*
  - *Bottom up*
    - *If F_n is frequent, generate F_{n+1}*
    - *Downward closer of support*
    - *Onepass through*
  - *Optimized*
    - *If subset of F_{n+1} of size n is not frequent, pruned F_{n+1}*
    - *Avoid single pass through*

http://magpiehall.com/apriori-algorithm/

- *WeMiT*

(Weighted Mining of Temporal Patterns)

  - *Compressed weighted baskets*
    - *92.5% decrease compare to uncompressed*
    - *Modified definition of support*

$$B = \{b_1^{w_1}, b_2^{w_2}, , b_n^{w_n}\} \quad \sum_{i=1}^{n} contain(b_i^{w_i}, X).w_i$$

    - *Running time reduced 15 times on average allow several passed through*

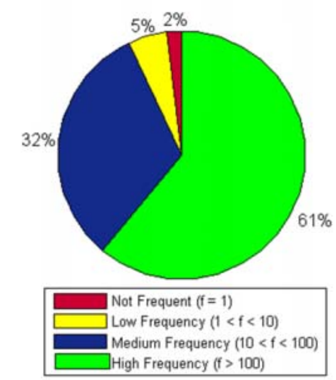


Figure 5. Distribution of basket frequencies for a sample user.

Context Prediction

- Search association rules generated by MobileMiner
  - Input: co-occurrence patterns
    - Current context: {Morning, Atwork}
    - Targe context: {UseGmail} , { UseOutlook}
  - Prediction based on confidence with decrease order
    - Return max
- Example:
  - $\{Morning\} \rightarrow \{UseGmail\}$ with confidence 0.9 but {UseOutlook} with 0.8
    - Return Gmail
  - $\{AtWork, Morning\} \rightarrow \{UseOutlook\}$ with confidence 0.9
    - Return both Gmail and Outlook

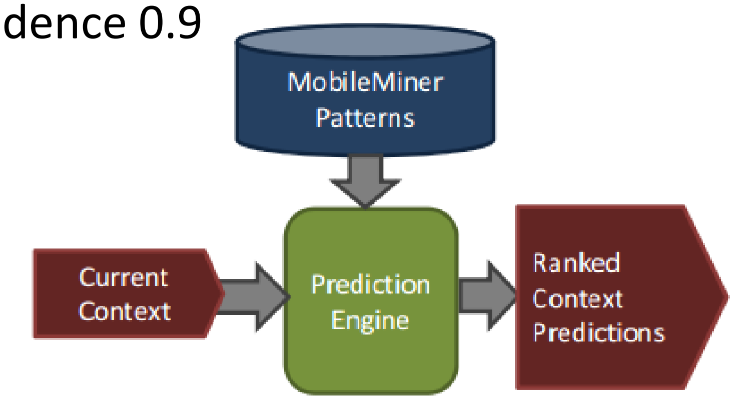
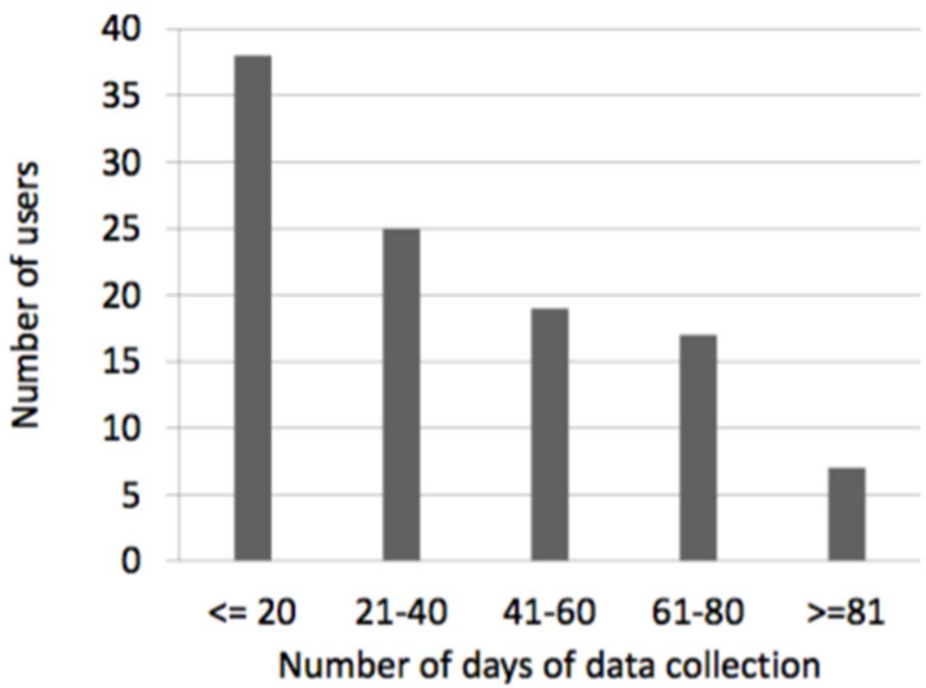


Figure 6. Context prediction using co-occurrence patterns.

Evaluation—Context Data Collection

- 106 participants
  - At least 40 Users collected more than 40 days
  - Around 25 Users collected 21-40 days



Evaluation—Context Data Collection

- 440 unique context events
  - Call events
    - Type
    - Duration
    - Number
  - SMS events
    - Type
    - Number
  - Inferred place identifiers
    - Home
    - Work
    - Outside
  - Location cluster label
  - Phone charging status
  - Battery levels
  - Foreground app usage events
  - Wi-Fi or cell connective
  - Cell id of current location
  - Binary movement status

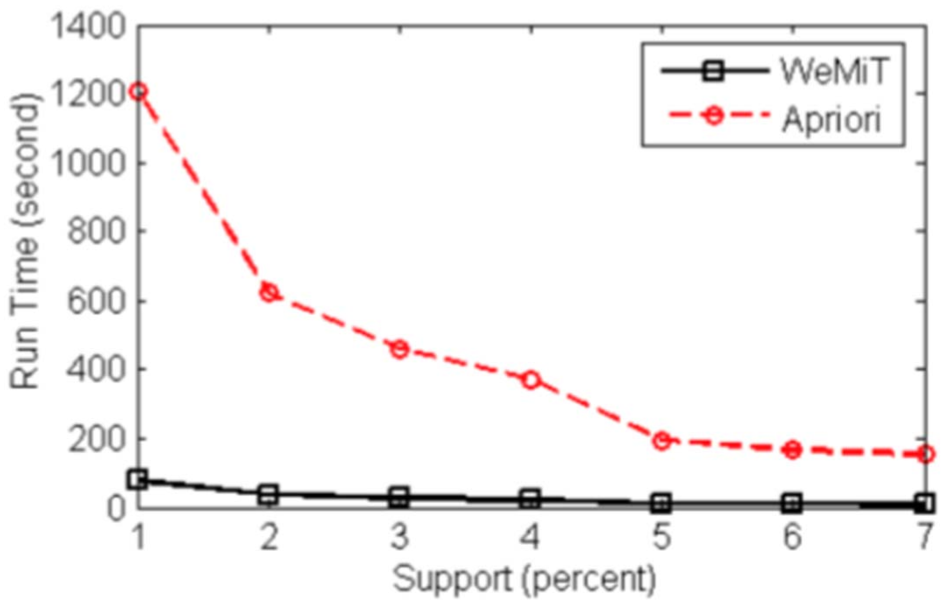
Evaluation—System Performance

- Is it feasible to run MobileMiner components on the phone?

Performance Metric	Base Basket Extraction	Base Rule Mining	App Usage Filtering	App Usage Rule Mining
Execution time	1.7 seconds	16.5 minutes	1.4 seconds	21.2 seconds
Memory	9.9 MB	44.2 MB	11.6 MB	1.0 MB
CPU Utilization	22.9 %	24.3 %	20.8 %	21.9 %
Number of baskets or rules	114275 baskets 8559 compressed	46675 rules	752 baskets 327 compressed	1062 rules
Energy per day as % of full battery	<0.01 %	0.45 %	<0.01 %	0.01 %

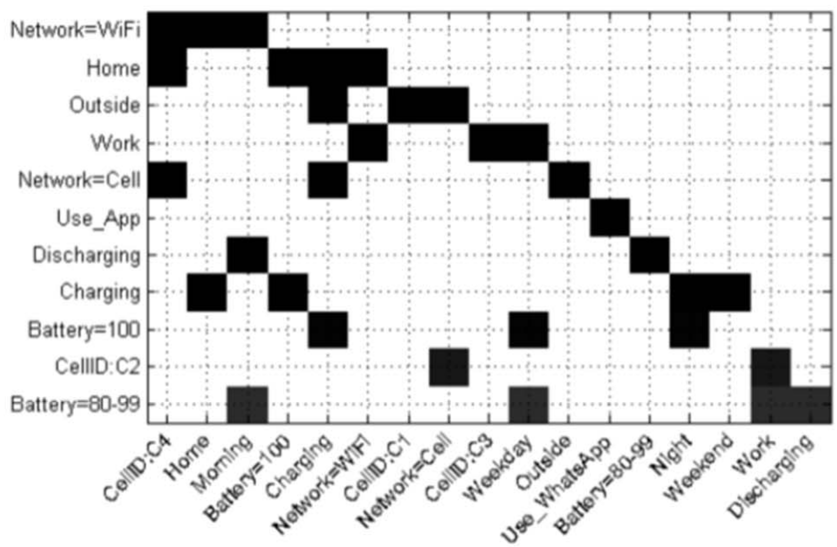
Evaluation—System Performance

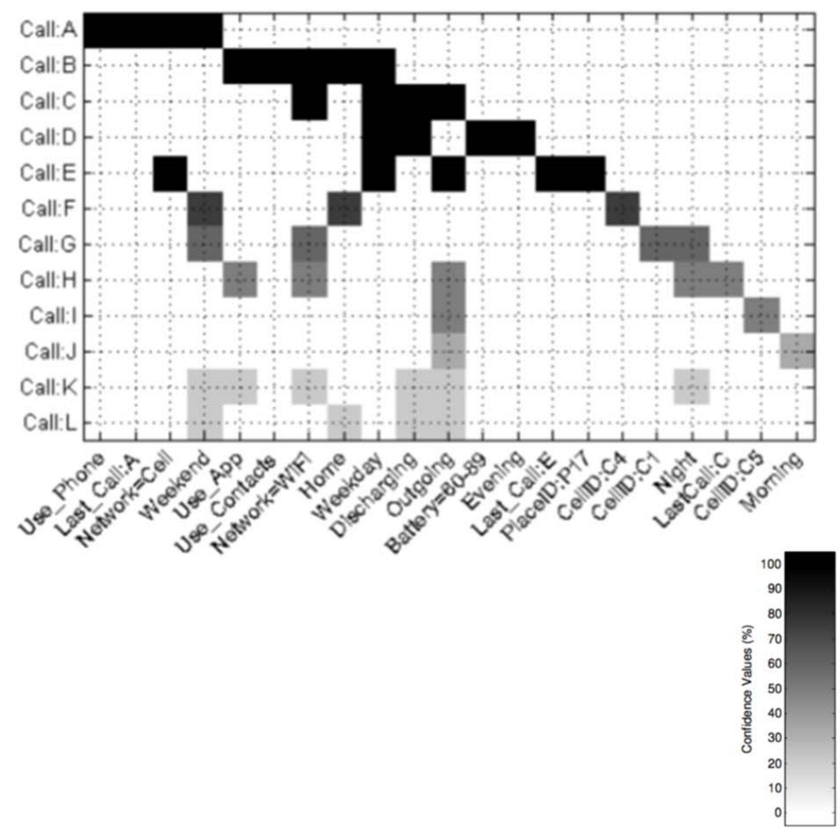
- How does WeMiT compare to Apriori



Evaluation——Patterns Generated

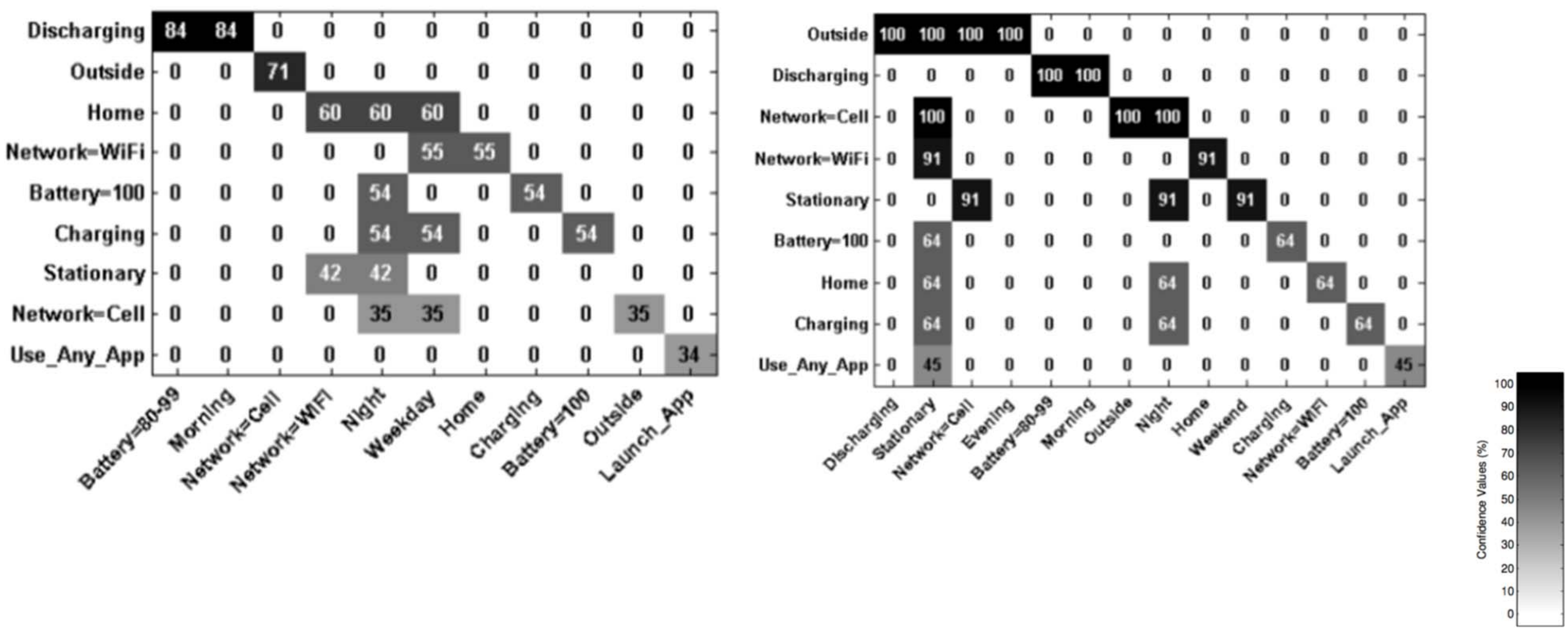
- What are some sample patterns and how do they use them?
  - Analyze the patterns generated by MobileMiner.
  - Get the confidence of each rule in the matrix visualization.





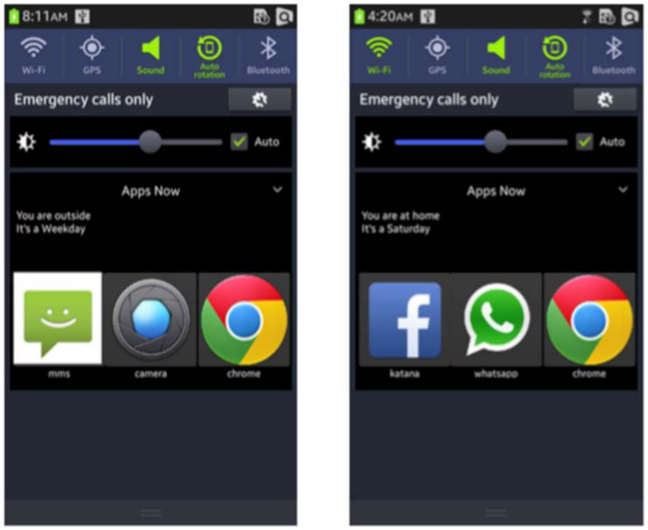
Evaluation—Patterns Generated

- What are some common patterns across multiple users?
  - Rather than the confidence, they show the percentage of users the pattern occurs in, either among all users (left) or among smaller group with very similar co-occurrence patterns (right)



Example Use Cases—App and Call Prediction

- App recommendation service with short cut icons
  - For each app, they show the reason why is was displayed based on the matching co-occurrence pattern

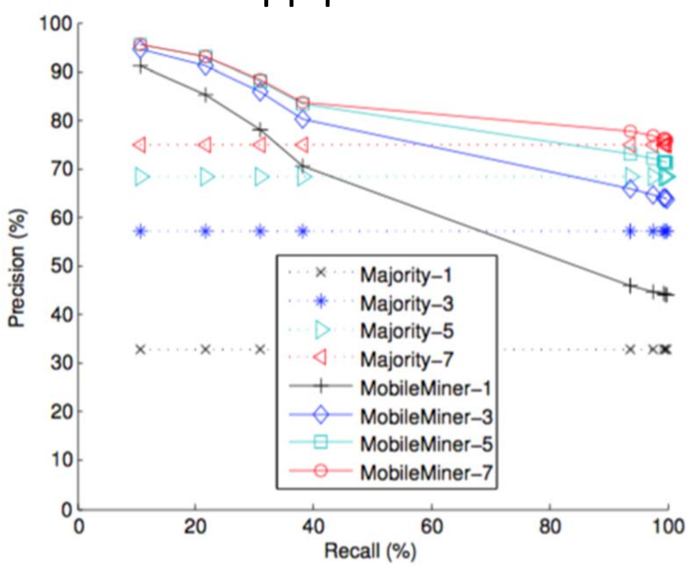


- Two evaluation metrics:
  - **Recall** is the proportion of app launches or outgoing calls for which **they show recommendations to the users**.
  - **Precision** is the proportion of **times the user uses one of shortcut icons** to complete his task.

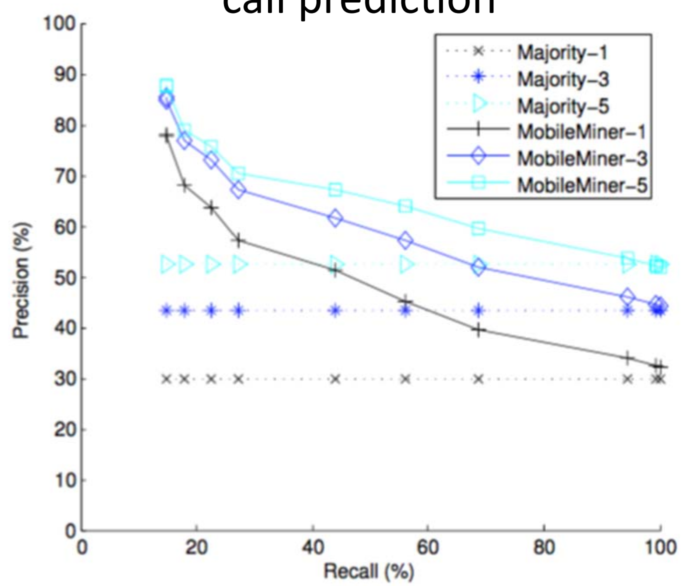
Example Use Cases—App and Call Prediction

- What is the Recall-precision tradeoff of predictions?

App prediction



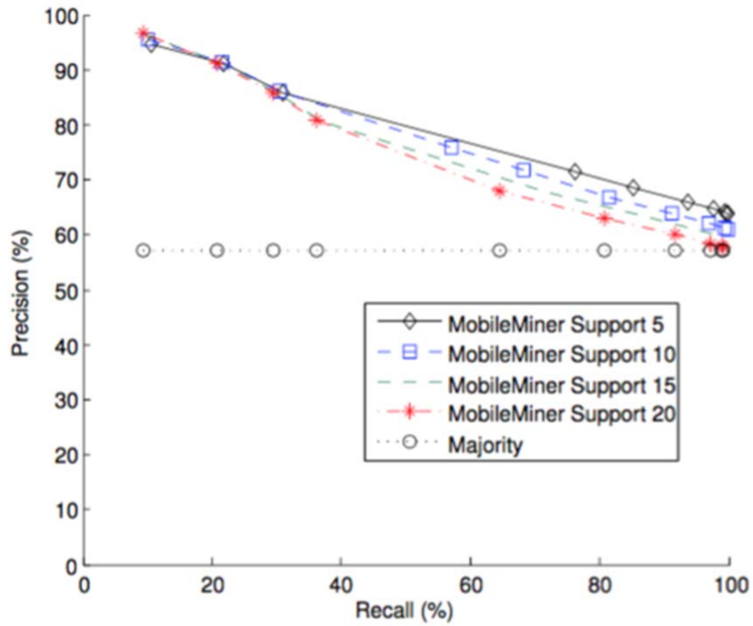
call prediction



- Recall-precision tradeoff for 1,3,5 and 7 shortcut shown.
- Typically, higher recall results in a lower precision, vice versa.

Example Use Cases—App and Call Prediction

- How do they choose the support value for mining patterns?



  - 4-5% improvement in precision as the support values decrease from 20 to 5.
  - Developer should choose a appropriate support threshold to achieve reasonable prediction accuracy without incurring too much phone resources.

User survey

- How often will users use our app recommendation service?
  - Use the service regularly: 57%
  - Use sometimes: 42%
- Where should the shortcut icons be placed?
  - Phone’s lock screen: 40%
  - Phone’s quick panel: 26%
  - Main tool bar: 33%
- How many shortcut icons should be displayed?
  - More than 6 icons: Very few
  - 4-6 icons: 71%
  - 1-3 icons: 26%

User survey

- Will users prefer a recall less than 100% for improved precision?
  - Higher precision: 54%
  - Always receive recommendations: 9%
  - Either case: 35%
- What is recall-precision tradeoff preferred by users?

Precision	No. of recommendations	Recall	Responses
90%	3	35%	30.95%
80%	3	51%	16.67%
80%	5	68%	23.81%
80%	7	80%	11.90%
75%	3	66%	4.76%
75%	5	87%	11.9%
75%	7	100%	19.05%

Related Work

- Focus on on-device mining of co-occurrence patterns over users mobile context data
  - Compare with other context-aware computation on mobile devices using longitude context data
    - Deal with privacy, data cost, and latency concerns
  - Compare with approaches use specialized predictive classifiers
    - More generalizable
    - Provides more configurability
    - Make predictions with lower accuracy even with missing context events
    - Co-occurrence patterns are more readable and directly usable by end users
  - A preliminary version of the work has been presented

Conclusions

- The novel MobileMiner system efficiently generates patterns using limited phone resources
  - 15 times performance improvement over Apriori
  - Generate overall frequent patterns in 16 minutes
  - Detail app usage pattern in 21 seconds
- Found interesting behavior patterns
- Improve the phone UI for launching app or calling contacts
- Future work
  - Explore co-occurrence patterns of events over long time durations
  - Systematically determine the correct frequency of running the mining algorithm
  - Perform a comparison of the context prediction approach
  - Extend to other types of patterns

References

1. Aggarwal, C. C., and Yu, P. S. A new approach to online generation of association rules. *IEEE Transactions on Knowledge and Data Engineering 13*, 4 (2001), 527–540.
2. Agrawal, R., and Srikant, R. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*, Morgan Kaufmann (1994).
3. Aharony, N., Pan, W., Ip, C., Khayal, I., and Pentland, A. Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing 7*, 6 (2011).
4. Allen, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM 26*, 11 (1983), 832–843.
5. Android operating system. http://www.android.com/.
6. Azizyan, M., Constandache, I., and Roy Choudhury, R. Surroundsense: Mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom '09)* (2009).

References

7. Banerjee, N., Agarwal, S., Bahl, P., Chandra, R., Wolman, A., and Corner, M. Virtual compass: Relative positioning to sense mobile social interactions. In *Proceedings of the 8th International Conference on Pervasive Computing (Pervasive '10)*, Springer-Verlag (2010).
8. Borgelt, C. Efficient implementations of apriori, eclat and fp-growth. http://www.borgelt.net, August 2013.
9. Cheung, D. W., Han, J., Ng, V. T., and Wong, C. Maintenance of discovered association rules in large databases: An incremental updating technique. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*, IEEE (1996), 106–114.
10. Samsung galaxy s4. http://www.samsung.com/latin_en/consumer/mobile-phones/mobile-phones/smartphone/GT-I9500ZKLTPA-spec.
11. Samsung gear. http://www.samsung.com/us/mobile/wearable-tech.
12. Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 2011.

References

13. Hao, T., Xing, G., and Zhou, G. isleep: Unobtrusive sleep quality monitoring using smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)*, ACM (2013).
14. Ifttt mobile recipes. https://ifttt.com/recipes.
15. ios 7. https://www.apple.com/ios/what-is/.
16. Kwapisz, J. R., Weiss, G. M., and Moore, S. A. Activity recognition using cell phone accelerometers. *SIGKDD Explorations Newsletter 12*, 2 (2011), 74–82.
17. Li, W., Han, J., and Pei, J. Cmar: accurate and efficient classification based on multiple class-association rules. In *Proceedings of IEEE International Conference on Data Mining (ICDM '01)*, IEEE (2001).
18. Lin, K., Kansal, A., Lymberopoulos, D., and Zhao, F. Energy-accuracy trade-off for continuous mobile device location. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys '10)*, ACM (2010).