When Did Your Smartphone Bother You Last?

Jeremiah Smith
Department of Computing
Imperial College London
180 Queen's Gate
London, SW7 2AZ, UK
jeremiah@imperial.ac.uk

Anna Lavygina
Department of Computing
Imperial College London
180 Queen's Gate
London, SW7 2AZ, UK
a.lavygina@imperial.ac.uk

Alessandra Russo
Department of Computing
Imperial College London
180 Queen's Gate
London, SW7 2AZ, UK
a.russo@imperial.ac.uk

Naranker Dulay
Department of Computing
Imperial College London
180 Queen's Gate
London, SW7 2AZ, UK
n.dulay@imperial.ac.uk

UbiComp '14, September 13 – 17 2014, Seattle, WA, USA

http://dx.doi.org/10.1145/2638728.2641701

Abstract
This paper is prompted by the overall question 'what is the most effective way to recognise disruptive smartphone interruptions?'. We design our experiments to answer 3 questions: 'Do users revise what they perceive as disruptive incoming calls as time goes by?', 'How do different types of machine-learners (lazy, eager, evolutionary, ensemble) perform on this task?' and 'Can we restrict the initial amount of data and/or the number of features we need to make predictions without degrading performance?'. We consider these questions using Cambridge University's Device Analyzer dataset.

Author Keywords
Smartphones, Notifications, Interruptions, Learning

ACM Classification Keywords
H.5.m [Information interfaces and presentation]:

Introduction and Background
With the rapid adoption of mobile devices, notifications from mobile devices increasingly compete for the user's attention. While some notifications can be beneficial, such as an alarm to remind the user about a meeting, some others can be disruptive, such as a notification about an unrelated event while they are in a meeting. We consider a notification disruptive to the user if the

disturbance of receiving the notification outweighs the perceived benefit [5]. In this paper we restrict ourselves to disruptive incoming phone calls.

Disruptive incoming calls have been studied in [1][4][3] and [5]. The datasets used in these studies range from 2 to 20 users, span 1 to 16 weeks and were collected by either querying the user about his interruptibility at specific times or reading the smartphone's state in the background as in the Device Analyzer dataset [6]. Previous approaches used machine-learning (k-nn, logistic regression, genetic programming, support vector machines and logic programming) to predict whether incoming calls should be silenced or allowed to ring and recognised the need to account for user preferences and change in user behaviour.

Because some users might prefer predictions that are biased towards recognising disruptive calls at the cost of some false positives, we use a weighted accuracy measure as our performance metric which takes into account the weighting of the two classes, disruptive or not disruptive.

Key findings

To answer our 3 questions, we extracted 10 datasets from the Device Analyzer dataset spanning from 137 to 839 days (mean 515) with sizes ranging from 604 to 3723 (mean 1487). Based on features used in other studies, we collected the following 11 features: *1) month of call*, *2) day of week*, *3) time of call*, *4) incoming number*, *5) cell tower id*, *6) location area code*, *7) SSID*, *8) screen on/off*, *9) time since screen was on last (sec)*, *10) time since screen was off the last (sec)*, *11) time since last call (sec)*. Exploratory data analysis revealed that incoming call patterns were heterogeneous across users, leading to large standard deviations in the number and types of calls received each day. For instance, on average, users answered 2.38 calls per day with a standard deviation of 1.26, such calls were labelled as non-disruptive, whereas calls that were declined, on average 0.93 per day with a standard deviation of 0.65, where labelled as disruptive (missed calls were discarded from datasets as no label could be inferred for them).

In order to find which supervised learning setting, offline, online or windowed, is appropriate for potential changes in user behaviour over time, we compared the accuracy of a random forest ensemble on the second versus the last third of the data while training on the first third. On average, the accuracy was 5% higher on the second third. The large standard deviation indicating that some users did not change their behaviour while others did. Interestingly we did not see any correlation between the number of days the data was gathered across and the presence of the change, again underlining the heterogeneity of the user population.

The evidence for changes in user behaviour suggested that online learning would outperform offline learning. We compared 4 types of learners: 3-NN, logistic regression, random forest and genetic programming in the three supervised learning settings mentioned above. While all learners, except 3-NN, managed to outperform the naive solutions, we found that the best performance was attained when no data was forgotten but that similar performance could be achieved by keeping only the last 1000 points in the training window. In all cases, the random forest ensemble learner performed best, followed closely by genetic programming, which, depending on the application area, can be more user-friendly as it can output human readable rules.

Lastly, we used recursive feature elimination on the 11

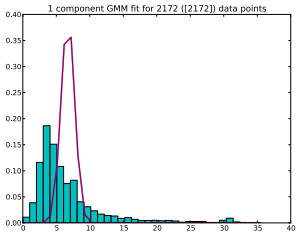

Figure 1: Modelling answered incoming calls times in seconds (x axis) using a 1 component Gaussian distribution

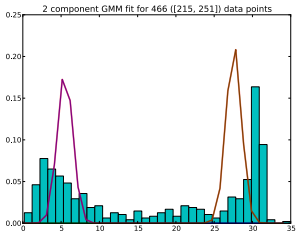

Figure 2: modelling unanswered (declined and missed) call time in seconds (x axis) using a 2 component Gaussian distribution

initial features and found that the top 5 (ordered by discriminative power) could be used without any significant loss in performance.

Collected data

The Device Analyzer dataset groups anonymised smartphone usage traces from more than 17000 users. The data is collected through a downloadable Android app that logs a wide range of Android system events (see http://deviceanalyzer.cl.cam.ac.uk/keyValuePairs.htm) for each user. The datasets we used, summarised in Table 1, were obtained by going through the event logs from top to bottom, each line corresponding to a single system event, rebuilding the phone state at each point in time. The events we collected were from the *phone, screen, conn* and *wifi* keys yielding 11 features for each incoming phone call with an extra feature used to distinguish unanswered incoming calls (declined versus missed).

Set	Days	Answered(avg/day)	Declined(avg/day)	Missed(avg/day)	Size
784c	137	605 (4.42)	266 (1.94)	340 (2.48)	871
7ab7	444	908 (2.05)	113 (0.25)	333 (0.75)	1021
1db8	152	350 (2.30)	285 (1.88)	585 (3.85)	365
d0a9	839	3070 (3.66)	653 (0.78)	1687 (2.01)	3723
c4e5	737	1399 (1.90)	497 (0.67)	934 (1.27)	1896
eb40	585	470 (0.80)	427 (0.73)	175 (0.30)	897
b9ae	587	932 (1.59)	324 (0.55)	205 (0.35)	1256
c260	370	1429 (3.86)	618 (1.67)	441 (1.19)	2046
bd04	688	482 (0.70)	122 (0.18)	228 (0.33)	604
3c7e	611	1513 (2.48)	410 (0.67)	134 (0.22)	1923
Mean	515±237	1116±810 (2.38±1.26)	372±186 (0.93±0.65)	506±479 (1.27±1.19)	1487±953

Table 1: Summary of the 10 datasets used in our experiments

Data labelling

We labelled datapoints as follows: answered calls were labelled as *non-disruptive* while declined calls where the user consciously chose not to answer the call by pressing the decline button, were labelled as *disruptive*. Answered calls could be recognised due to a *phone|OFFHOOK* event taking place in between a *phone|RINGING* and *phone|IDLE* event. On the other hand, missed calls, when the user either ignored or was unaware of the call happening, were initially indistinguishable from declined calls as they were both characterised by a *phone|RINGING* followed by a *phone|IDLE* event. To separate the two cases, we modelled the distribution of ringing times for both answered and unanswered calls using a mixture of respectively a one and two Gaussian component distribution, an example is shown in Figures 1 and 2.

We labelled unanswered calls as disruptive if they fell within 2 standard deviations of the answered call ringing time distribution mean, making the assumption that the user would take the same amount of time to press the answer button as the decline button and ignoring caller abandons, where the caller drops a call before either reaching the user or their voicemail.

The resulting datasets were unbalanced with an average of 372±186 disruptive (declined) calls, for an average dataset length of 1116±810.

Learning

Evaluating machine-learners

For some users, correctly predicting disruptive calls might be more important than correctly predicting non-disruptive calls or vice-versa. This can make the traditionally reported accuracy misleading in terms of a user's experienced performance. In an unbalanced dataset, accuracy can be high if there is a majority of non-disruptive calls and the learner should be biased toward recognising those calls. Together with the class imbalance problem comes a class weighting problem. Ideally, a learner is able to adapt to a user's preference in terms of which class of calls they judge more important to predict correctly. We thus report performance in terms of weighted accuracy [5] computed according to Equation .

Equation 1

$$wA = \frac{w * r * |c^0| + s * |c^1|}{w * |c^0| + |c^1|}$$

Where r is the recall (sensitivity) of a classifier, s is the specificity, c^0 and c^1 are the set of positive (disruptive) and negative (non-disruptive) calls, $w = \frac{truepositiveweight}{truenegativeweight} \in [0, \infty)$ is the user preference ratio of true positive (correctly predicted disruptive calls) importance, to true negative (correctly predicted non-disruptive calls) importance, $truepositiveweight \in [0, \infty)$ and $truenegativeweight \in (0, \infty)$.

Set name	Acc. set 1	Acc set 2	delta
784c	70.6	65.9	4.7
7ab7	89.4	88.2	1.2
1db8	57.5	53.7	3.8
d0a9	85.7	83.5	2.2
c4e5	72.3	71.5	0.8
eb40	56.8	52.8	4.0
b9ae	84.9	81.1	3.8
c260	84.1	59.7	24.4
bd04	78.1	76.2	1.9
3c7e	80.6	76.9	2.7
Mean	76.0	71.0	5.0

Table 2: The accuracy of an offline random forest ensemble when trained on the first third of the datasets and tested separately on the second (set 1) and third (set 2) thirds of them

Choice of learners

We benchmarked the following 4 learners: a 3-NN classifier to represent lazy learning, also used in [1][5]; a logistic regression learner, an efficient eager learner also used in [4]; a random forest learner, a popular ensemble learning technique; and genetic programming, an evolutionary technique that learns trees that can be read as rules, also used in [5]. Our choice of metric required learners to be cost-sensitive, which was achieved (with the exception of 3-NN and genetic programming) by under sampling the lesser weighted class in proportion to the weight of the minority class. For instance, $w = 2$ would undersample c^1 by 50%. In genetic programming weighted accuracy was used as the fitness function.

Learning paradigms

Learners were evaluated in 3 supervised learning settings with an initial 30% of training data: offline, our performance baseline; online, setting in which the learner retrains after each prediction using the point's label; windowed, where learners are only given the most recent n training instances. The last approach can be thought of as online learning with *forgetting* to account for the potential changes in data distributions due to users changing what they consider to be a disruptive call as time goes by.

Change in user behaviour

In data obtained from naturally occurring processes, class distributions and/or boundaries are likely to change over time, a phenomenon known as concept drift. In the case of smartphone notifications, a person might change his schedule, which in turn might change the times at which he considers notifications to be disruptive.

To test for these changes, we divided each dataset into 3 equal parts and tested for a difference in classification accuracy (w = 1 in Equation 1) of the random forest ensemble, in the second and third segments while training on the first segment only. Since datapoints are taken in chronological order, a higher performance on the second segment indicates that testing points that are closer in time to training points are better predicted i.e. that a user's disruptive call patterns change as time goes by.

Table 2 shows that the performance on the second testing set is never greater than on the first, while in 5 out of 10 sets the performance between the two are within 3%. In the rest, the performance delta went up to 24.4% (set c260). Interestingly, the time span of datasets does not correlate with changes in disruptive call patterns. For instance, the long set c4e5 (737 days 1896 points) has very low performance delta between the two testing sets. This highlights again the user population's heterogeneity. Nevertheless, most users do change their incoming call preferences over time and this suggests that online and windowed learning will outperform offline learning.

Comparing machine-learners

Figures 3 and 4 show that random forests and genetic programming outperform naive solutions, which always predict the same class c^0 or c^1, in all settings and for all user preferences. 3-NN's consistently weaker performance shows that case based learning is not adapted to this application. As expected, online and large-training-window windowed learning outperforms offline learning as shown in Figure 5. Although users do change their behaviour, it seems that the benefit of forgetting all old data does not markedly outweigh the cost of losing certain points that remain relevant for predictions. Here perhaps a more sophisticated forgetting heuristic would yield better results.

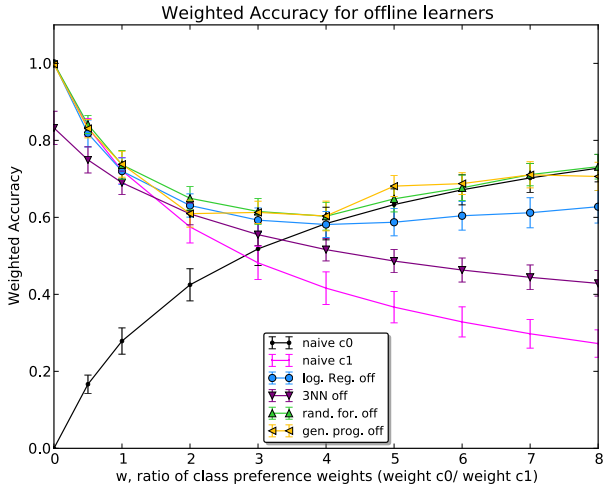

Figure 3

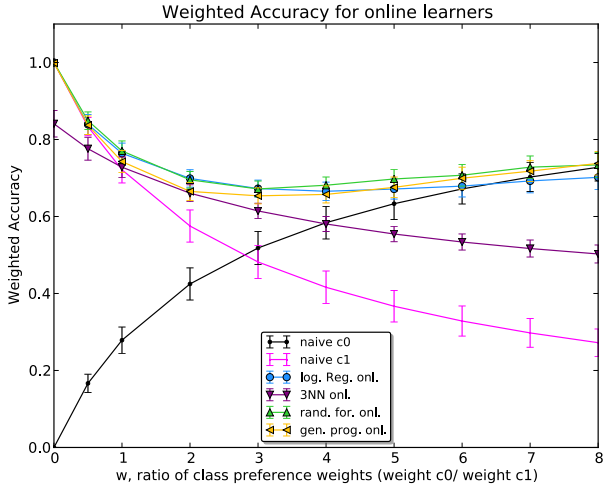

Figure 4

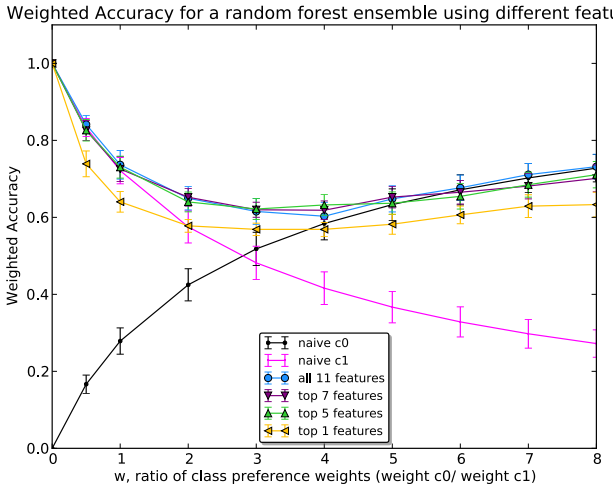

Figure 5

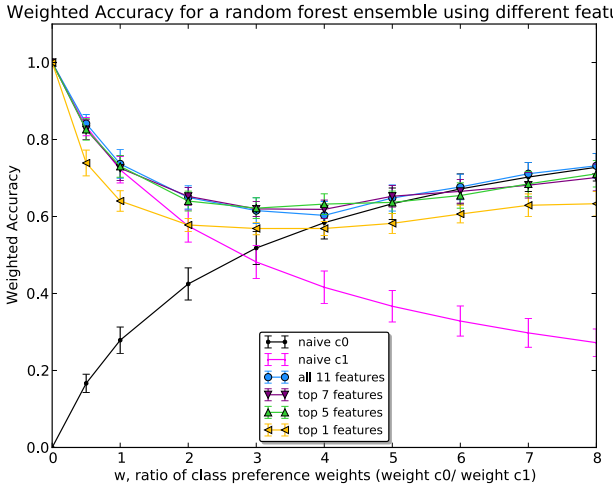

Figure 6

	Feature										
Set	1	2	3	4	5	6	7	8	9	10	11
784c	8	5	3	4	11	10	6	9	7	2	1
7ab7	6	8	2	7	3	10	9	11	5	4	1
1db8	8	6	4	2	5	7	10	11	3	9	1
d0a9	6	7	3	2	4	9	11	10	5	8	1
c4e5	7	5	3	6	4	8	10	1	2	9	1
eb40	7	6	3	5	4	8	9	11	2	10	1
b9ae	9	6	4	3	5	10	7	11	2	8	1
c260	8	6	3	2	5	7	11	10	4	9	1
bd04	6	7	5	2	3	10	8	11	4	9	1
3c7e	6	8	5	4	3	7	9	11	2	10	1
overall	7	6	2	4	5	9	10	11	3	8	1

Table 3: Feature ranking obtained for datasets using recursive feature elimination with random forests. The ranks presented are the average ranks for datsets over all values for w of in Equation 1. Smaller ranks correspond to the features with higher predictive power.

Feature selection

To examine which features are most discriminative we used recursive feature elimination (RFE) [2]. Using an estimator, in our case random forests, RFE recursively ranks features until a predefined number of features is found. In our experiments, the features with lowest ranks (least impact on weighted accuracy) were eliminated recursively until the most discriminant features remained. Table 3 presents the feature ranking for all datasets.

We evaluated the performance on subsets of the best ranked features. Figure 6 shows that the subset of five best ranked features: *11) time since last call*, *3) time of call*, *9) time since screen was on last*, *4) incoming number* and *5) cell tower id* is enough to obtain results similar to using the full set.

This confirms the intuition that the disruptiveness of a call depends on the current activity of the user (9 and 11), for instance, if the user just finished a call it might be a good time to receive another one; temporal factors (3), the user might dislike if his phone rings late at night; locational factors (5), a user might find all calls while he is at yoga class disruptive; and contextual factors (4) a call from your boss/partner during working hours might be beneficial/disruptive.

Discussion

Although users are found to be different in terms of call frequencies and amount of change in disruptive call patterns as time goes by, we focused on methods that worked well across all sets, since in practice it is difficult to know the type of user a dataset comes from. Whether the performance achieved during our experiments is high enough to be part of a practical/user-friendly application will depend on two factors: finding a way to compute a user's w parameter and whether the mistakes in predictions are acceptable for the user (for instance, a user may not be bothered by wrong predictions in exceptional circumstances e.g. an incoming phone call in the middle of the night). These two areas will be addressed in future work.

Acknowledgements

This work was supported by EU FP7 projects iCareNet (grant 264738) and Allow Ensembles (grant 600792).

References

[1] Fisher, R., and Simmons, R. Smartphone interruptibility using density-weighted uncertainty sampling with reinforcement learning. In *ICMLA '11* (2011).

[2] Granitto, P., Furlanello, C., and Gasperi, F. Recursive feature elimination with random forest for ptr-ms analysis of agroindustrial products. *Chemometrics and Intelligent Laboratory Systems* (2006).

[3] Markitanis, A., Corapi, D., Russo, A., and Lupu, E. C. Learning user behaviours in real mobile domains. In *Proc. of ILP'11* (2011).

[4] Rosenthal, S., Dey, A. K., and Veloso, M. Using decision-theoretic experience sampling to build personalized mobile phone interruption models. In *Proc. of Pervasive'11*, Springer-Verlag (2011).

[5] Smith, J., and Dulay, N. Ringlearn: Long-term mitigation of disruptive smartphone interruptions. In *Symposium on Activity and Context Modeling and Reasoning (ACOMORE)*, PerCom '14 (2014).

[6] Wagner, D. T., Rice, A., and Beresford, A. R. Device analyzer: Understanding smartphone usage. In *Proc. of MobiQuitous '2013* (2013).