Understanding and Managing Notifications

Swadhin Pradhan\textsuperscript{1}, Lili Qiu\textsuperscript{1}, Abhinav Parate\textsuperscript{2}, and Kyu-Han Kim\textsuperscript{3}.

\textsuperscript{1}UT Austin \quad \textsuperscript{2}Lumme Inc. \quad \textsuperscript{3}Hewlett-Packard Laboratories
Problem with notifications
Problem with notifications
Problem with notifications
Problem with notifications
Problem with notifications
Problem with notifications

- Too many
  - Overwhelming
  - Information overload
Problem with notifications

- Too many
  - Overwhelming
  - Information overload
Problem with notifications

- Too many
  - Overwhelming
  - Information overload

- Interrupting at inopportune moments
  - Disrupting tasks
  - Interfering lifestyle

- Demanding high responsiveness
  - High attention demand .. (Stress)
How to schedule notifications?

- Scheduling *important notifications at opportune moments*
How to schedule notifications?

- Scheduling *important notifications* at *opportune moments*
How to schedule notifications?

- Scheduling *important notifications* at *opportune moments*

  Importance  Interruptibility
How to schedule notifications?

- Scheduling *important notifications* at *opportune moments*
How to schedule notifications?

- Scheduling *important notifications at opportune moments*
Remaining Outline

- Datasets
- Insights gained from the data
- Assessing notification importance
- Building a notification manager
- Conclusion
Evaluation datasets
Evaluation datasets

- **Data Set – I (Notifbase App)**
  - 40 users recruited (30 users data > 2 weeks)
  - App usage, Screen on/off, Wi-Fi status, ringer mode, sound level, notification properties, shade opening, notification action etc.
  - Android Accessibility service used

- **Data Set – II (Snotify App)**
  - 12 users recruited from the above set
  - **Explicit feedbacks** from users for perceived importance (Online survey of 402 users)
Notifications are disrupting
Notifications are disrupting
Notifications are disrupting

Receives $\sim 60$ notifications/day
Notifications are disrupting

Receives ~60 notifications/day
Notifications are disrupting

Receives ~60 notifications/day

Opens notification drawer ~15 times/day
Users ignore most notifications
Users ignore most notifications
Users ignore most notifications

- Users tend to ignore 20-50% of the generated notifications.
- Less than 20% of these are causing app launch events.
Users have limited attention span
Users have limited attention span
Users have limited attention span

Takes \( \sim 1-20 \) seconds within a notification drawer session
Users have limited attention span

Takes \(~1-20\) seconds within a notification drawer session
Users have limited attention span

Takes \(~1\text{-}20\) seconds within a notification drawer session

Attention span does not vary with number of notifications
Apps fail to evaluate ‘importance’
Apps fail to evaluate ‘importance’

Developers assign Default or High Priority to notifications
Apps fail to evaluate ‘importance’

Developers assign **Default** or **High Priority** to notifications

Users’ response time almost **constant**
Summary of Insights

- Users receive large number of notifications
- Users take action to prevent disruption
- Users ignore most notifications (20%-50%)
- Users have limited attention span (~10s)
- Apps tend to assign overly high priorities
Summary of Insights

- Users receive large number of notifications
- Users take action to prevent disruption
- Users ignore most notifications (20%-50%)
- Users have limited attention span (~10s)
- Apps tend to assign overly high priorities

Correct assessment of notification importance is critical towards removing unwanted notifications and utilizing users’ limited attention span.
Predicting notification importance

- *Engagement level* as an indicator.
Predicting notification importance

- **Engagement level** as an indicator.
- Users engage with notifications in several ways:
  - Ignore
  - Read (e.g. notification drawer open)
  - Read and dismiss
  - Take some action (e.g. “Archive” or “Delete” a mail)
  - Launch an app
Predicting notification importance

- Recent Phone activity
- Notification Info
- Engagement history
- Surrounding Noise
- Location

Notification Importance
Predicting notification importance

**Binary classification**: C4.5 Decision tree, Linear Regression, Random forest, SVM using **22 features**
Feature Ranking

- Application name of notification
- Temporal features related to interaction (e.g. notification post time, clear time etc.)
- Hour of the day
- Notification title
- Ringer mode of the phone
- Weekend status
- Location cluster etc.
Evaluating a personalized predictor

Results shown on 10-fold cross validation (ground-truth via explicit feedback) and can achieve ~87% accuracy
Generic v Personalized v Clustered

- A generic model trained on a subset users’ data and predict the rest.

- 10-fold cross validation on personal data of each user.

- Cluster users based on \#applications used, \#unique locations visited … Predict within cluster.
Generic v Personalized v Clustered

Random Forest

SVM

Decision Tree

Linear Regression

Generalized  Clustered  Personalized
Clustered model performs better than the personalized model
A smarter notification manager

SmartNotify
A smarter notification manager

SmartNotify

Model Trainer

Feature and Engagement Level Extractor

Context Monitor

Usage Monitor

Sensors
Time and Location
Audio
Phone Events
App Usage

Notification Events
Notification Usage
A smarter notification manager

Users

Applications

SmartNotify

Decision Engine

Model Trainer

Feature and Engagement Level Extractor

Context Monitor

Usage Monitor

Notifications

Phone Events

App Usage

Sensor Events

Notification Usage

Sensors

Time and Location

Audio

Notification Events

App Usage

Notification Usage

Users interacts with apps and gives feedback

Feature and Engagement Level Extractor

Users interacts with apps and gives feedback

Decision Engine

Model Trainer

Feature and Engagement Level Extractor

Context Monitor

Usage Monitor

Sensors

Time and Location

Audio

Phone Events

App Usage

Notification Events

Notification Usage
A smarter notification manager

Selected Notifications

Users

Applications

Interacts with apps and gives feedback

Suppressed/Delayed Notifications

Decision Engine

Feature and Engagement Level Extractor

Model Trainer

Context Monitor

Usage Monitor

Sensors
Time and Location
Audio
Phone Events
App Usage

Notification Events
Notification Usage

SmartNotify

Users interacts with apps and gives feedback.
Notification manager performance

- Used Weka based C4.5 decision tree for model training and *InterruptMe* library.
Notification manager performance

- Used Weka based C4.5 decision tree for model training and *InterruptMe* library.
Notification manager performance

- Used Weka based C4.5 decision tree for model training and *InterruptMe* library.

Feature extraction ~10s for and Tree building ~50s for 1000 instances
Conclusion

- Users are getting disrupted by notifications
- Can *suppress* unwanted notifications if we can predict user’s engagement level
- Implemented a *smarter* notification manager which can predict notification importance with ~87% accuracy
Future works

- A first step toward understanding *micro user interaction* with notification.

- Prediction model can be used to decide *display order* or *modality* of notifications across *multiple devices*
Thank you & any question?

Codes for data collector apps:
https://bitbucket.org/swadhinp/notifbase
https://bitbucket.org/swadhinp/snotify

Online learning based prediction

![Graph showing accuracy vs batch size for different loss functions and norms.](image)

- SGD with L2 norm
- SGD with L1 norm
- Hinge Loss function
- Squared Loss function

Accuracy vs Batch Size for different loss functions and norms.
Online learning based prediction

Stochastic Gradient Descent with L2 norm gives best performance and stabilizes with 500 batch size
How do users avoid disruptions?
How do users avoid disruptions?

Setting their devices to silent or vibrating mode.