Nurture: Notifying Users at the Right Time Using Reinforcement Learning

Bo-Jhang Ho  Mehmet Koseoglu  Bharathan Balaji  Mani Srivastava
The usefulness of mobile services diminishes if users do not want to interact with them.

**Intervention-based services**

- Headspace
- Nike
- Google Fit
- Health Kit
- Runkeeper
- My Rehab Recovery
The usefulness of mobile services diminishes if users do not want to interact with them.

**Intervention-based services**
- **HEADSPACE**: TREAT YOUR HEAD RIGHT
- **Nike**: Apple Watch
- **Google Fit**: Health kit
- **MY REHAB RECOVERY**:

**Context-based reminder**
- **Tasker**:
- **Medisafe**:
- **Google Keep**:
- **OK Google**: List, Calendar, Note, Sound, Sound, Graph, Motor, Motor, Motor, Motor, Motor, Motor.
The usefulness of mobile services diminishes if users do not want to interact with them.

**Intervention-based services**
- HEADSPACE
- Nike
- Google Fit
- Health kit
- My Rehab Recovery
- Quora
- Amazon Mechanical Turk
- Lyft
- Stack Overflow

**Context-based reminder**
- Tasker
- Medisafe
- Google Keep
- "OK Google"
- Waze
- Tripadvisor
- Yelp
- Yahoo Answers
- Facebook

**Crowdsourcing**
The usefulness of mobile services diminishes if users do not want to interact with them.
The consequences of inappropriate notifications

Low adherence

Poor response quality

Ignoring alerts

Causing distraction
Interruptibility
Interruptibility - Related Work

- Transitioning between activities
  - Oasis - TOCHI’10
  - Attelia - PerCom’17
  - Fischer et al. - MobileHCI’11
  - Slide to X - CHI’14
  - ProactiveTasks - MobileHCI’14

- Infer from user context
  - Sarker et al. - UbiComp'14
  - Thyme - ICMLA'17
  - Mehrotra et al. - UbiComp'15
  - PrefMiner - UbiComp'16
  - InterruptMe - UbiComp'14
  - Pielot et al. - UbiComp'15
Reinforcement learning setup

1. Get state
2. Perform action
3. Obtain reward
Reinforcement learning setup

1. Get state
2. Perform action
3. Obtain reward
Reinforcement learning setup

1. Get state
2. Perform action
3. Obtain reward

Environment

Agent

Nurture system
Reinforcement learning setup

Environment

1. Get state
   User context = \{time, location, motion activity, last notification\}

Agent

2. Perform action

3. Obtain reward

Nurture system
Reinforcement learning setup

Environment

1. Get state
   User context = {time, location, motion activity, last notification}

2. Perform action
   - To send a notification now
   - To keep silent

3. Obtain reward

Agent

Nurture system
Reinforcement learning setup

Environment

1. Get state
   User context = \{time, location, motion activity, last notification\}

2. Perform action
   - To send a notification now
   - To keep silent

3. Obtain reward
   User response: Accept (+1), ignore (0), dismiss (-5)

Agent

Nurture system
Advantages in reinforcement learning

1 Sequential decision process
Advantages in reinforcement learning

1. Sequential decision process
2. Exploration

I've seen this state before, let me try action 2 this time!
Advantages in reinforcement learning

1. Sequential decision process
2. Exploration
3. Online learning
Experiment 1: Synthetic simulation setup
Experiment 1: Synthetic simulation setup
Experiment 1: Synthetic simulation setup

- Describe user daily patterns (e.g., when to leave home)
- 4 different patterns
- Repeat after a week

- We estimate the response by *looking up* the MTurk survey results
- We collected 3,019 surveys in total

**Daily routing model**

*ExtraSensory dataset*

**Response model**

*Amazon Mechanical Turk*
#1 It is 10:30 AM on Saturday. You're sitting (or standing) at work. You responded to (or clicked on) a notification 1 hours 15 minutes ago. Now you receive a notification from our app. The notification says “Can you take 10 seconds complete this questionnaire?” What will you do?

What would be your action?

- Dismiss this notification
- Leave the notification and answer it later
- Take ten seconds to answer the questionnaire
Experiment 1: Synthetic simulation - Learning parameters

- Supervised learning algorithms
  - Support vector machine as baseline
  - 4 weeks for training
  - 15 notifications per day during training phase

- Reinforcement learning algorithms
  - Contextual bandit
  - Q-learning

Decision point
(get observation and make an action)
Experiment 1:
Synthetic simulation results - Response rate

Contextual bandit

Baseline (SVM): 0.77

Q-learning

Baseline (SVM): 0.77
Experiment 1:
Synthetic simulation results - Notification quantity

**Baseline (SVM): 17.8 / day**

Contextual bandit

Q-learning

Baseline (SVM): 17.8 / day
Experiment 1: Synthetic simulation results - Notification quantity

Converge after 3 weeks

Contextual bandit

Q-learning
Experiment 2: 
Online interactive simulation setup

1. Generate question and deploy

Nurture system

2. Obtain result and update

#1 It is 10:30 AM on Saturday. You’re sitting (or standing) at work. You responded to (or clicked on) a notification 1 hours 15 minutes ago. Now you receive a notification from our app. The notification says “Can you take 10 seconds complete this questionnaire?” What will you do?

What would be your action?

<table>
<thead>
<tr>
<th>Option</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dismiss this notification</td>
</tr>
<tr>
<td>Leave the notification and answer it later</td>
</tr>
<tr>
<td>Take ten seconds to answer the questionnaire</td>
</tr>
</tbody>
</table>
Experiment 2:
Online interactive simulation result (Q-learning)

Response rate

Notification quantity
Experiment 2:
Online interactive simulation result (Q-learning)

Response rate

Find the opportune moments

Notification quantity

Reduce number of notifications due to low response rate

Increase the confidence
Experiment 2:
Online interactive simulation result (Q-learning)

Response rate

Notification quantity

Keep number of delivered notifications with decent response rate

Drop the amount since user preference changed
Real deployment
Future work and conclusion

• Future work
  • Field study to collect notification interactions in the wild
  • Study the consequence of the reward function

• Conclusion
  • Reinforcement learning can increase response rate and number of responses compared against supervised learning
  • Reinforcement learning can adapt to user behavior changes
Nurture: Notifying Users at the Right Time Using Reinforcement Learning

Bo-Jhang Ho, Mehmet Koseoglu, Bharrathan Balaji, and Mani Srivastava

Thanks for listening

Our code is online:
https://github.com/nesl/Nurture-UbiTtention18