Hit-or-Wait: Coordinating Opportunistic Low-effort Contributions to Achieve Global Outcomes in On-the-go Crowdsourcing

Yongsung Kim
Darren Gergle
Haoqi Zhang
Imagine that you lost your wallet on your way to the venue.
How might we leverage CHI attendee’s existing routine and route to help you find your wallet?
Opportunistic/Pull-based
Directed/Push-based

Nope, already headed to the party
Our goal: leverage existing route to notify when it is convenient for helpers
Our goal: and achieve globally effective outcomes
Conceptual approach

- Achieve globally effective outcomes in physical crowdsourcing by indirectly coordinating opportunistic contributions with people on-the-go.
Our Approach

Sadilek et al. ICWSM 2013
Kim et al. HCOMP 2016
Doryab et al. Ubicomp 2018

Directed Approach

Teodoro et al. CHI 2014
Thebault-Spieker et al. CSCW 2015

Opportunistic Approach
When do we notify people of a task?
When do we notify people of a task?

- we do not want to notify all the time, but only notify at the “best moment”
When do we notify people of a task?

- we do not want to notify all the time, but only notify at the “best moment”

- we can optimize task assignment \cite{Kandappu2016}, but people may not accept the task and we also do not know where they may go next
When do we notify people of a task?

• we do not want to notify all the time, but only notify at the “best moment”

• we can optimize task assignment [Kandappu et al. CSCW 2016], but people may not accept the task and we also do not know where they may go next

• we may send low-valued tasks or miss opportunities to notify.
Opportunistic Hit-or-Wait

• Uses decision-theory to decide **on-the-fly** whether to notify a helper of a task right now or wait for better opportunities in the future, in ways that reason both about system needs across tasks and about a helper’s changing patterns of mobility.
Hit-or-Wait Example
Hit!
Region 1
Region 2
Region 3
Region 4
Region 5
Our approach: Modeling a sequence of Hit-or-Wait decisions with a Markov Decision Process (MDP)

• State $s$: task location

• Reward function $R(s, a)$: value of notifying a task at state $s$

• Transition function $P(s'|s)$: likelihood of reaching state $s'$ from state $s$.

• Action: $\{\text{hit, wait}\}$
  
  • if hit, it moves to terminal state
  
  • if wait, it moves to next state
Modeling a sequence of Hit-or-Wait decisions with a Markov Decision Process (MDP)

\[ V^t(s) = \max(R(s, \text{hit}), \sum_{s'} P(s' | s) V^{t-1}(s')) \]
1. Encode value of notifying a helper of a task

\[ V^t(s) = \max(R(s, \text{hit}), \sum_{s'} P(s' | s) V^{t-1}(s')) \]
2. Model likelihood of reaching next location from current location

\[ V^t(s) = \max(R(s, \text{hit}), \sum_{s'} P(s' | s) V^{t-1}(s')) \]
3. Compare the expected value of notifying now with the expected value from making a decision later if we wait.

\[ V^t(s) = \max(R(s, \text{hit}), \sum_{s'} P(s' | s) V^{t-1}(s')) \]
Studies

- A simulation study
- A field deployment
Simulation study

• compare hit-or-wait with other approaches

• understand how hit-or-wait mechanism works

• understand how model accuracy affects the performance
Simulation setup

- Dataset: 5,983 running routes from 2,419 users in RunKeeper

- Measure:
  - Search Quality: likelihood of finding an item given searches

- Conditions:
  - Hit-or-Wait
  - Node Counting
  - Optimal solution (full knowledge of routes)
Hit-or-Wait maximizes user contributions without explicit coordination
Node Counting + Threshold

Decision-Theoretic “Hit-or-Wait”

derived : over searched

Optimal under searched
A field deployment

- How does Hit-or-Wait work in practice?
- What are some failure cases in practice?
- What are users’ perception of their contributions?
Field deployment setup

- 10-day study with 25 users (13M, 12F) in lost and found Scenario
- Dataset: Pre-study (N=11) with location tracking
- Measures:
  - Search quality
  - Value of hit or wait decisions
- Conditions:
  - Hit-or-Wait
  - Optimal (full knowledge)
Results

• 248 notifications sent and 60 searches conducted along their routes (24.19% acceptance rate)

• Among the searches, 4 different participants found 4 items out of the 9 search requests
Hit-or-Wait reached 84% of optimal solution in practice
Hit-or-Wait made effective wait decisions
Failure cases in misprediction
Failure cases in misprediction
Follow-up interviews

- 7 participants who helped at least once
- Hit-or-Wait example visualizations
Users wanted to have better understanding in how they contributed to the global goal

- “Maybe also having information like if someone does find the item, then I would know I was being helpful...I was helping part of that even if I wasn’t the exact person to find it.” — P7
Applicability of Hit-or-Wait

- Volunteer-based peer-to-peer services where the system goal is to effectively provide help for each other.
- Low-effort sensing and community sensing where the goal is to ensure data coverage and details.
Decision theory as a way to support communities

Context-sensitive task notifications

Decision theory as a way to support communities

**Context-sensitive task notifications**


**This work: Individual-level Coordination**
Decision theory as a way to support communities

Context-sensitive task notifications

This work: Individual-level Coordination

Future work: Community-level Coordination
Takeaways

• Hit-or-Wait allows volunteers to go about their routine, but indirectly coordinates their contributions to achieve better system needs and helper convenience.

• Introduces ways to use decision-theoretic approach to not only optimize but to support convenient interactions.
Hit-or-Wait: Coordinating Opportunistic Low-effort Contributions to Achieve Global Outcomes in On-the-go Crowdsourcing

Hit or Wait and chat after the session

Yongsung Kim
Darren Gergle
Haoqi Zhang

@DeltaLabNU

@DeltaLabNU