Trichromatic Online Matching in Real-time Spatial Crowdsourcing

Tianshu Song 1, Yongxin Tong 1, Libin Wang 1, Jieying She 2, Bin Yao 3, Lei Chen 2, Ke Xu 1

1 SKLSDE Lab and IRI, Beihang University, China
2 The Hong Kong University of Science and Technology, Hong Kong, China
3 Shanghai Jiao Tong University, China

1{yxtong, songts, lbwang, kexu}@buaa.edu.cn, 2{she, leichen}@cse.ust.hk, 3yaobin@cs.sjtu.edu.cn

Introduction

• Spatial Crowdsourcing (a.k.a Mobile Crowdsourcing)
  • Online platforms that facilitate spatial tasks to be assigned and performed by crowd workers, e.g. O2O applications.

• Motivation
  • Most O2O platforms work on real-time scenarios.
  • Some emerging O2O applications need to assign three types of objects:
    • Sports trainers, sports facilities and users.
    • Hairstylists, salon and customers.

The GOMA Problem

• Given
  • A set of tasks requester \( T \): location \( t_r \), arriving time \( b_r \), leaving time \( e_r \) and range radius \( r_r \).
  • A set of crowd workers \( W \)
    • Each \( w \in W \): location \( t_w \), arriving time \( b_w \), leaving time \( e_w \), range radius \( r_w \).
  • A set of crowd workplaces \( P \)
    • Each \( p \in P \): location \( t_p \), arriving time \( b_p \), leaving time \( e_p \).
  • Utility Function: \( U(t, p, w) \).

• Find a matching \( M \) to maximize the total utility
  \[ \text{MaxSum}(M) = \sum_{t \in T, p \in P, w \in W} U(t, p, w) \] s.t.
  • Deadline Constraint.
  • Range Constraint.
  • Invariable Constraint: Once a task \( t \) is assigned to a worker \( w \), the allocation of \( t \) cannot be changed.

Online Algorithm Evaluation: Competitive Ratio (CR)

• Randomized Algorithm

\[ \text{CR} = \frac{\min_{\text{OPT}(T, W, P, U)} \text{MaxSum}(\text{OPT})}{\text{MaxSum}(M)} \]

Greedy Algorithm

• Match all triples when it is possible

Basic-Threshold Algorithm

• Steps
  1. Choose an integer \( k \) from 1 to \( \ln(U_{\text{max}} + 1) \)
  2. Filter the edges with weights greater than \( e^k \).
  3. Use a greedy strategy on the remaining edges.

Adaptive-Threshold Algorithm

• Adaptively adjust the probability distribution of choosing different thresholds.
• When an object appear, choose a new threshold according to the learned probability distribution

\[ \text{MaxSum} \geq (1 - \epsilon) \text{MaxSum(OPT Basic-Threshold)} - \frac{(1 - \epsilon) \epsilon}{2\epsilon} \sum_{t \in T} (p(t))^2 - \frac{\epsilon^2}{4} \ln(\epsilon) \]

Experimental Evaluation

Object | Location | Arrival Time | Leaving Time |
--- | --- | --- | --- |
\( t_1 \) | (4.50,6.00) | 8.00 | 8.10 |
\( t_2 \) | (4.50,4.75) | 8.02 | 8.12 |
\( t_3 \) | (5.50,5.00) | 8.05 | 8.15 |
\( t_4 \) | (3.00,4.50) | 8.08 | 8.18 |
\( t_5 \) | (2.50,3.00) | 8.10 | 8.20 |
\( t_6 \) | (4.00,3.25) | 8.11 | 8.21 |
\( t_7 \) | (3.25,2.00) | 8.13 | 8.23 |
\( t_8 \) | (5.00,3.50) | 8.15 | 8.25 |
\( t_9 \) | (5.50,2.00) | 8.17 | 8.27 |
\( t_{10} \) | (4.50,2.00) | 8.19 | 8.29 |

Utility Score | Match | Utility Score | Match |
--- | --- | --- | --- |
\( (t_1, p_1, w_1) \) | 18 | \( (t_1, p_1, w_1) \) | 20 |
\( (t_2, p_2, w_2) \) | 10 | \( (t_2, p_2, w_2) \) | 12 |
\( (t_3, p_3, w_3) \) | 90 | \( (t_3, p_3, w_3) \) | 48 |
\( (t_4, p_4, w_4) \) | 20 | \( (t_4, p_4, w_4) \) | 72 |
\( (t_5, p_5, w_5) \) | 20 | \( (t_5, p_5, w_5) \) | 12 |