上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

An Approach to Rapid Worker Discovery in Software Crowdsourcing

Song Feiya

Shanghai Jiao Tong University

Agenda

1. Introduction
2. Motivation
3. System Design
4. Key Algorithms
5. Experiments

1. Introduction

Crowdsourcing

--Outsource problems or tasks to an undefined network of people

--An online, distributed problem-solving and production model

INNOCENTIVE®

知乎
与世界分享你的知识、经验和见解


[topcoder]™

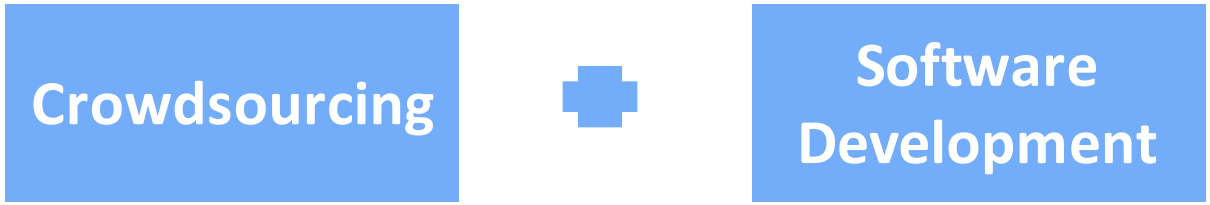
微差事

amazonmechanical turk
beta

1. Introduction

Challenge of Traditional Software Development :

- Talent — Hire professional developers
- Scale —Difficult to meet the large-scale task
- Closed — Internal development process

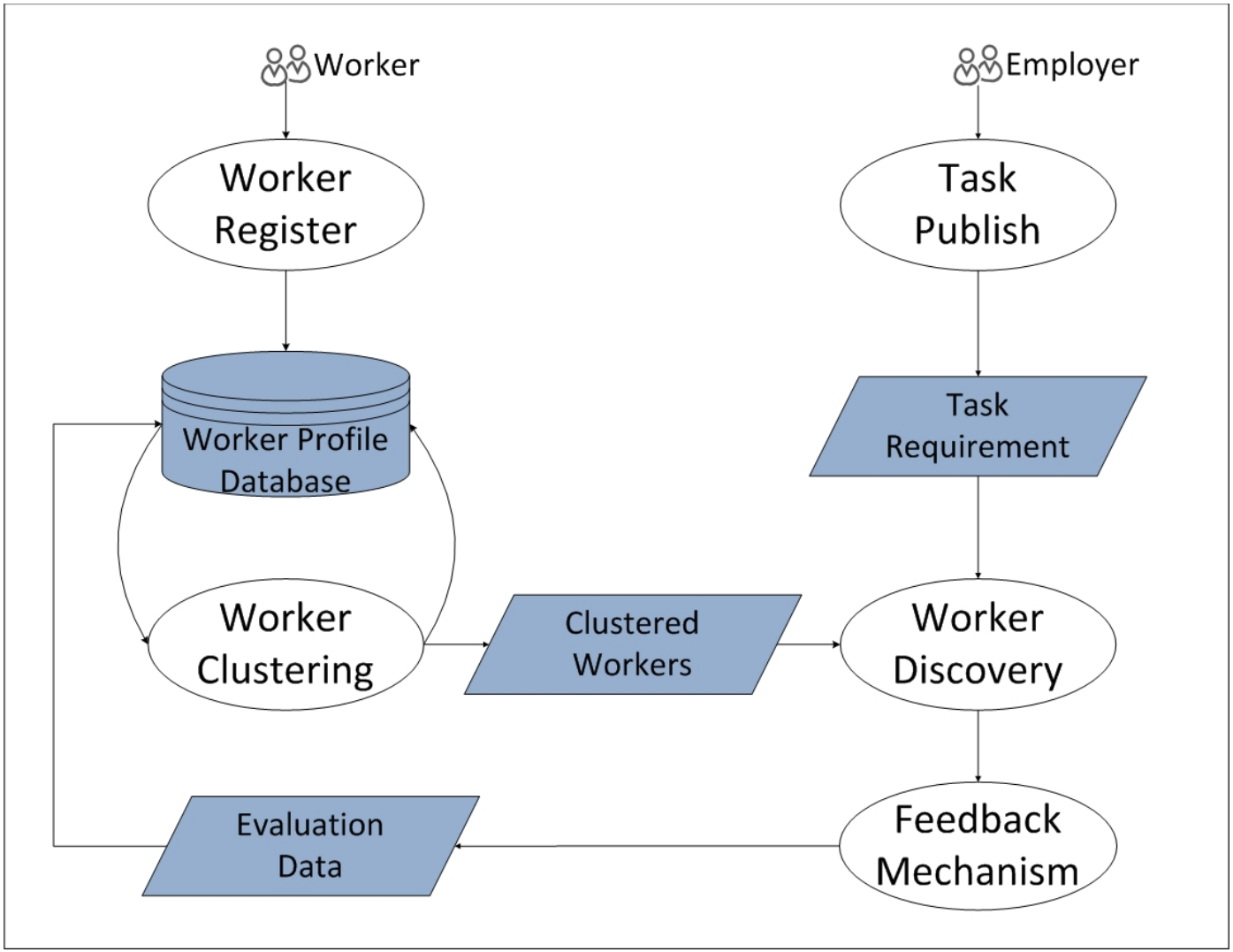


2. Motivation

- Existing Software Crowdsourcing Platform -- Upwork

Distributed Workforce	Worker Registration	Select Job
> 1 million 80,000 jobs	Skill set Self-evalution	Browse Task Specifications Application Interview

3. System Design



4. Key Algorithms

Clustering method

Algorithm1: clustering method

Input: worker set $W = W_1 \cup W_2 \cup \ldots \cup W_k$; center set $C = \{C_1, C_2, \ldots C_k\}$;

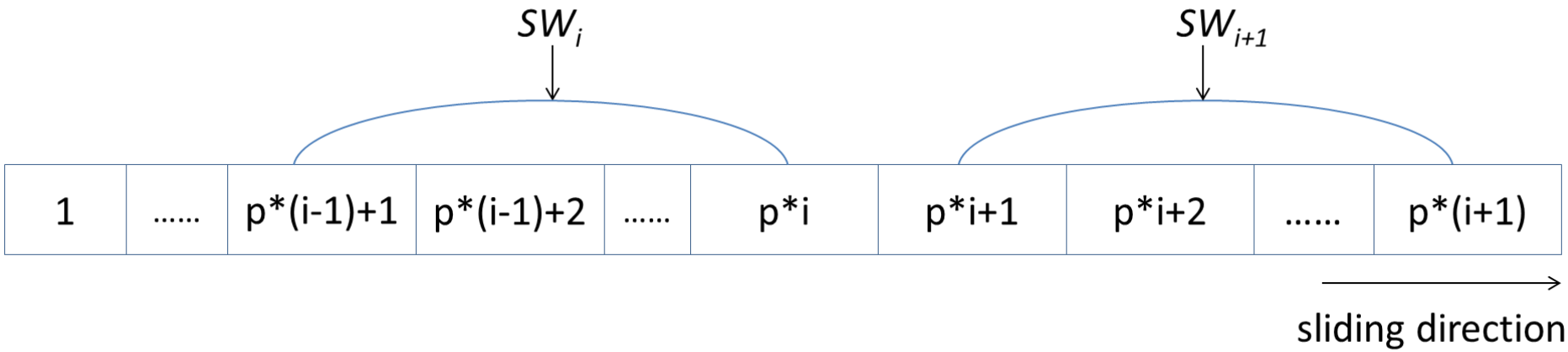
Output: updated worker set $W' = W_1' \cup W_2' \cup \ldots \cup W_k'$; updated center set $C' = \{C_1', C_2', \ldots C_k'\}$

Function:

```
begin
  while true
    W' <- W, C' <- C
    iter_count++  //record number of iterations
    for each worker w_i in W
      Distance_set_i <- {distance(w_i,C_1), …distance(w_i,C_k)}
      Nearest_center_i <- min(Distance_set_i)
      W_target <- w_i, Nearest_center_i ∈ W_target
      //add each worker to the nearest cluster
    end for
    C' <- Update_Centers(C') //recalculate each center
    if
        ∀w ∈ W, exists w' ∈ W' and w == w'
        ∀w' ∈ W', exists w ∈ W and w' == w
        break
    else if
        iter_count ⩾ MAXIMUM //reach maximum iteration
        break
  end while
  Output(W', C')
end
```

4. Key Algorithms

- Sliding Window Analysis



$$SD_{qi} = W * SD_{q(i-1)} + (1 - W) * \frac{\sum_{j=1}^{m} FD_{qj}}{m}$$

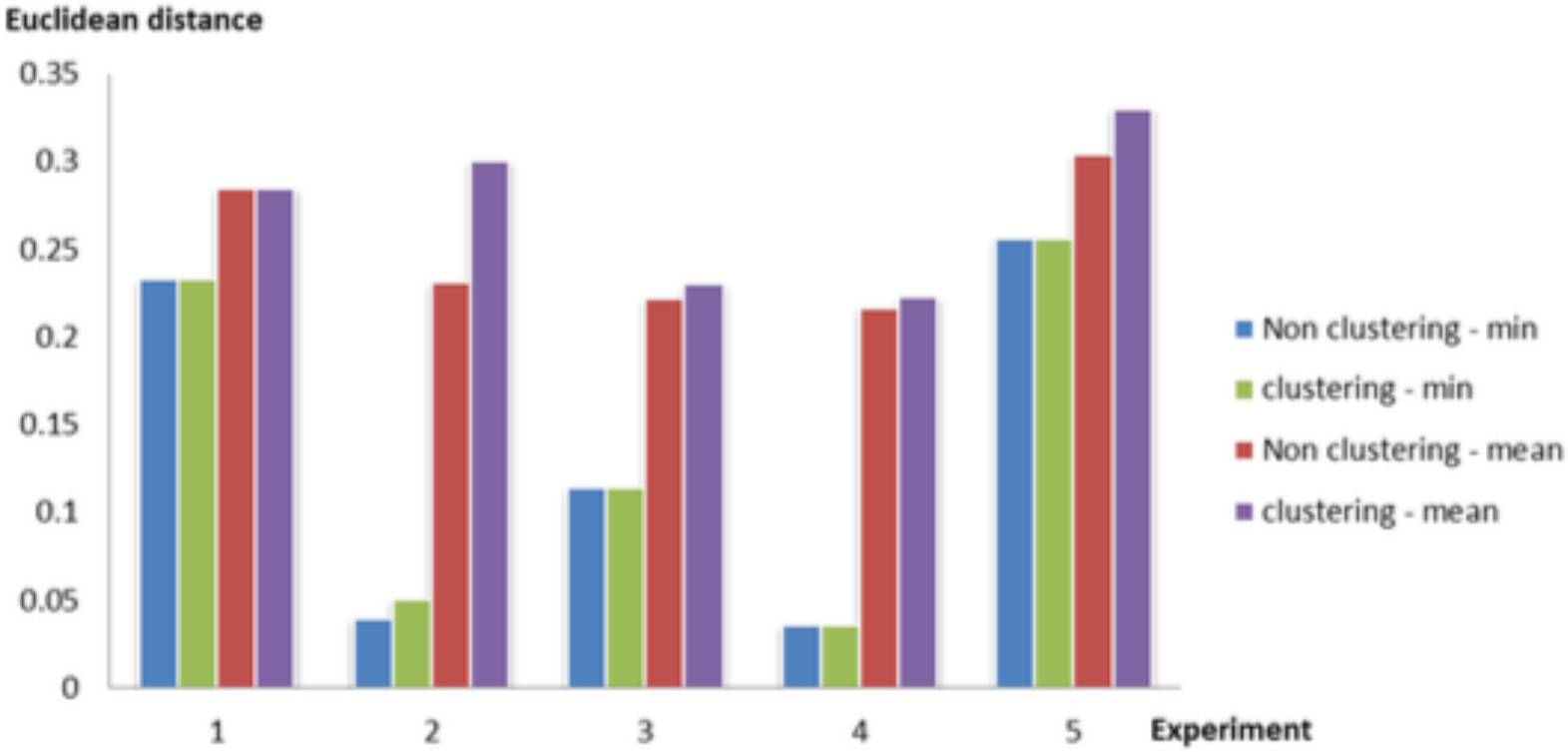
5. Experiments

- Time cost of clustering

	k	Num	N	N'_c	$Cost_{clustering}$	$Time_{clustering}$	$Time_{non\text{-}clustering}$
1	10	1000	10000	917	3.61	0.10	1
2	10	1000	10000	1060	1.13	0.12	1.12
3	10	1000	10000	979	1.35	0.10	1.03
4	10	1000	10000	1122	1.35	0.12	1.09
5	10	1000	10000	983	1.58	0.10	1.02

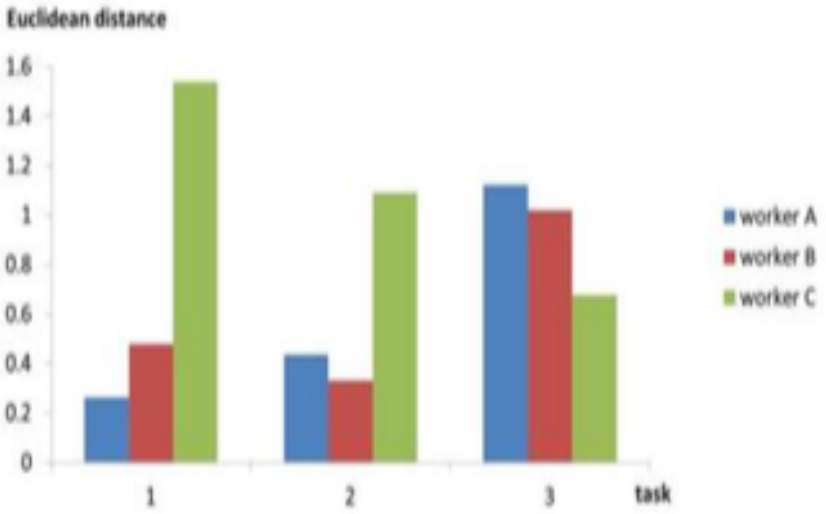
5. Experiments

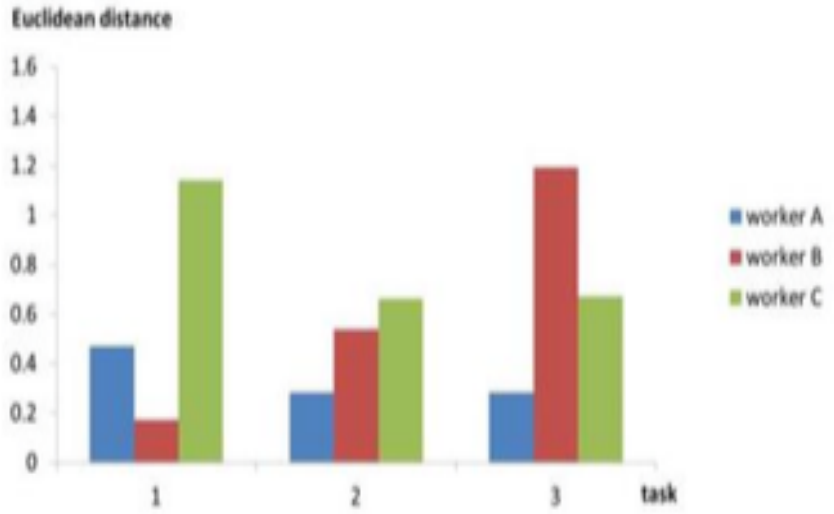
- Accuracy of clustering



5. Experiments

- Accuracy of Sliding Window





Q&A

Thank You!