Crowdsourcing in Information and Knowledge Management

Lei Chen, HKUST, China
Dongwon Lee, Penn State, USA
Meihui Zhang, SUTD, Singapore

Slide available @ **http://is.gd/7Vxetk**

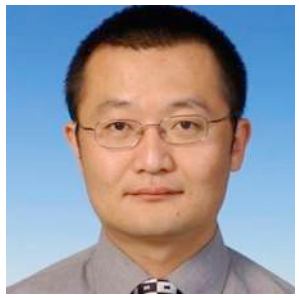
CIKM 2014 Tutorial

TOC

- Part 1
  - Crowdsourcing Database Systems

- Part 2
  - Human Powered Database Operations

- Part 3
  - Quality Control



Part 1

CROWDSOURCING DATABASE SYSTEMS

Crowdsourcing – What is it?

Crowdsourcing is a process that involves outsourcing tasks to a distributed group of people. The difference between crowdsourcing and ordinary outsourcing is that a task or problem is outsourced to an undefined public rather than a specific body, such as paid employees.

Crowdsourcing on Web 2.0

WIKIPEDIA
The Free Encyclopedia

YAHOO! ANSWERS

Shanghai

From Wikipedia, the free encyclopedia

Coordinates: 31°12'N 121°30'E

For other uses, see Shanghai (disambiguation).

Shanghai is the largest Chinese city by population[8][9] and the largest city proper by population in the world.[10] It is one of the four direct-controlled municipalities, with a population of more than 24 million as of 2013.[5] It is a global financial center,[11] and a transport hub with the world's busiest container port.[12] Located in the Yangtze River Delta in East China, Shanghai sits at the mouth of the Yangtze in the middle portion of the Chinese coast. The municipality borders the provinces of Jiangsu and Zhejiang to the north, south and west, and is bounded to the east by the East China Sea.[13]

For centuries a major administrative, shipping, and trading town, Shanghai grew in importance in the 19th century due to European recognition of its favorable port location and economic potential. The city was one of five opened to foreign trade following the British victory over China in the First Opium War while the

Shanghai
上海市

Municipality

Shanghai Municipality

Clockwise from top: A view of the Pudong skyline; Yu Garden, China Pavilion along with the Expo Axis, neon signs on Nanjing Road, and The Bund

Is there a difference between Hongqiao Airport and Pudong Airport in Shanghai?

I'm flying domestically in China (Guangzhou-Baiyun to Shanghai) and is there any difference between these 2 airports? They're both same price? Which airport is better?

Best Answer Asker's Choice

Marcela P answered 5 months ago

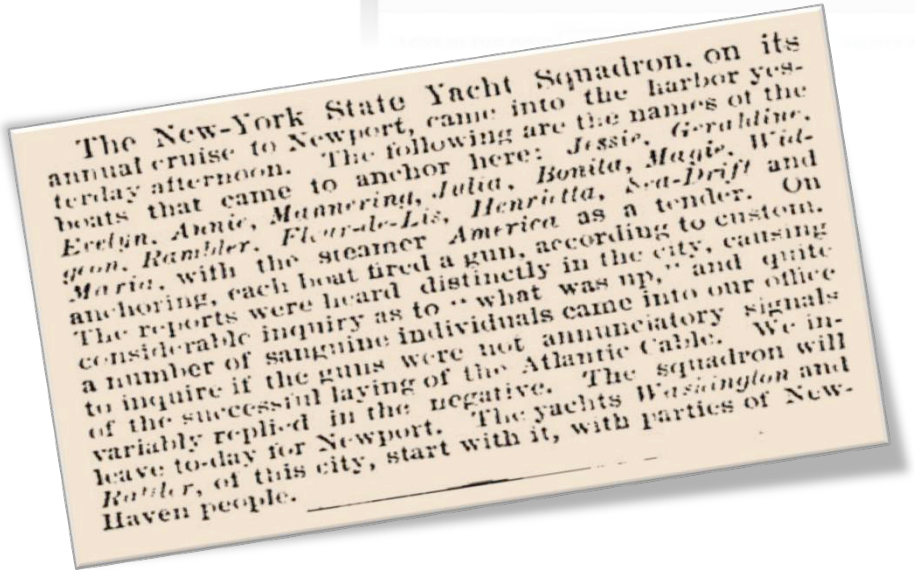
Pudong (PVG) is normally used by international traffic while Hongqiao (SHA) normally used by domestic flights. PVG is quite far from the city, while SHA is close to downtown.

Asker's rating & comment

*****

Crowdsourcing on Image Tagging

Crowdsourcing over the Internet



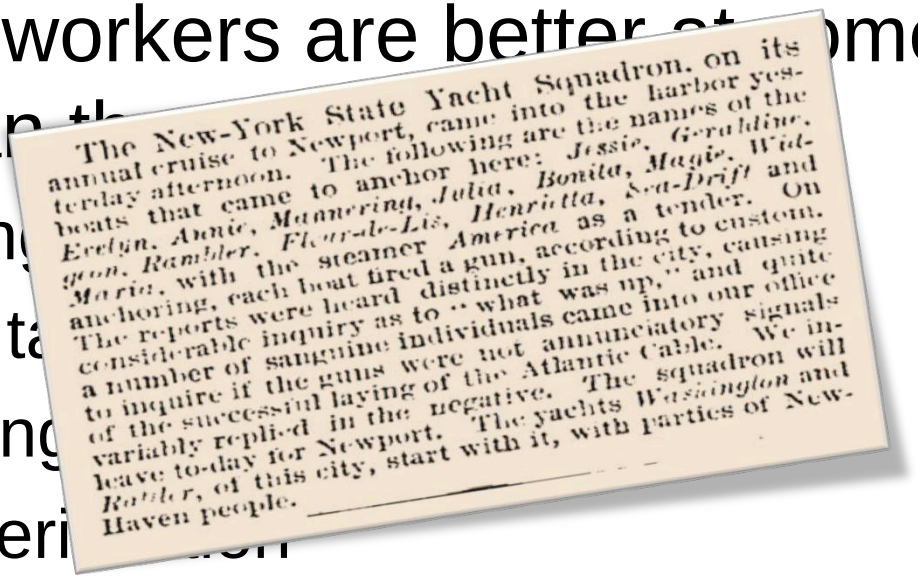
The New-York State Yacht Squadron, on its annual cruise to Newport, came into the harbor yesterday afternoon. The following are the names of the boats that came to anchor here: *Jessie*, *Geraldine*, *Evelyn*, *Annie*, *Mannering*, *Julia*, *Bonita*, *Magic*, *Widgeon*, *Rambler*, *Fleur-de-Lis*, *Henrietta*, *Sea-Drift* and *Maria*, with the steamer *America* as a tender. On anchoring, each boat fired a gun, according to custom. The reports were heard distinctly in the city, causing considerable inquiry as to "what was up," and quite a number of sanguine individuals came into our office to inquire if the guns were not announciatory signals of the successful laying of the Atlantic Cable. We invariably replied in the negative. The squadron will leave to-day for Newport. The yachts *Washington* and *Butler*, of this city, start with it, with parties of New-Haven people.

Harness Human Intelligence

- Human workers are better at some complex jobs than the machines
  - Creating content
  - Image tagging
  - Digitizing paper work
  - Data verification
  - Data cleansing
  - Classification
  - Rating
  - Sentiment analysis
  - More …

Harness Human Intelligence

- Human workers are better at some complex jobs than th
  - Creatin
  - Image ta
  - Digitizing
  - Data veri
  - Data cleansing
  - Class
  - Ratin
  - Sentir
  - More …



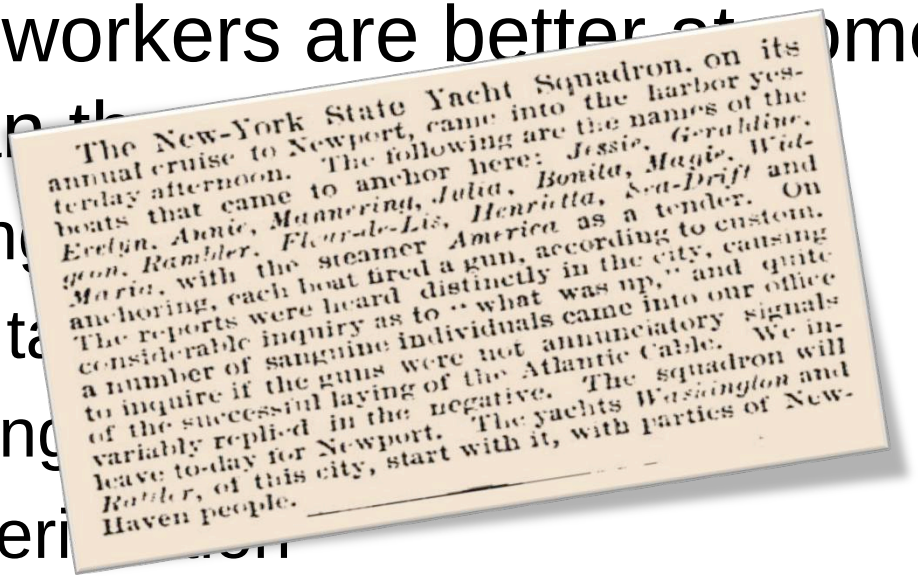
The New-York State Yacht Squadron, on its annual cruise to Newport, came into the harbor yesterday afternoon. The following are the names of the boats that came to anchor here: *Jessie*, *Geraldine*, *Evelyn*, *Annie*, *Mannering*, *Julia*, *Bonita*, *Magic*, *Wildgoon*, *Rambler*, *Fleur-de-Lis*, *Henrietta*, *Sea-Drift* and *Maria*, with the steamer *America* as a tender. On anchoring, each boat fired a gun, according to custom. The reports were heard distinctly in the city, causing considerable inquiry as to "what was up," and quite a number of sanguine individuals came into our office to inquire if the guns were not announciatory signals of the successful laying of the Atlantic Cable. We invariably replied in the negative. The squadron will leave to-day for Newport. The yachts *Washington* and *Rattler*, of this city, start with it, with parties of New-Haven people.

OCR Transcription (Errors Not Highlighted)

' letz-1- rrk fit: 1' . on its to Vc ,rt, cann into tlm yc H_ tcr,la, .n. 'l l; , arc ti:(h of thc 1",ats that to ltc rc: ,;. , l; ., l: rel!;n. tani., , ./olio, lJuteilu, . 1!'i./_ ;lr"n. liam! Jr.r. F'l,nr_.Z.._%i;;, ,, : rt-lrn: am/ tf.rri.:, t?m steamer as a tr nW r. Uu ,tin;t, c ac?1 1",at firm/ a t;nn, accor.liu; to .t rn. 'Cl.w r. wu ru lm:nui MistinW /y in u;th, -. ink ;:,k as to "what w ax 1111, :111(l vle:iR a of ;: (,am(into, mnr r-, tm if tlm wo r(uu.i n:' of t?u : la?:lv. \ 'c : ol in thc , ucr:atic , , Tlau :; will h:aw tu-li.r \. '1'lm yap?tts ll ,,n an,/ l, ,rr:l. r, (,t tf,is r:ity, start witli it, with lu:rtic: ol \ 1- e:l.k.

Harness Human Intelligence

- Human workers are better at some complex jobs than th
  - Creating
  - Image ta
  - Digitizing
  - Data veri
  - Data
  - Class
  - Rating
  - Sentir
  - More



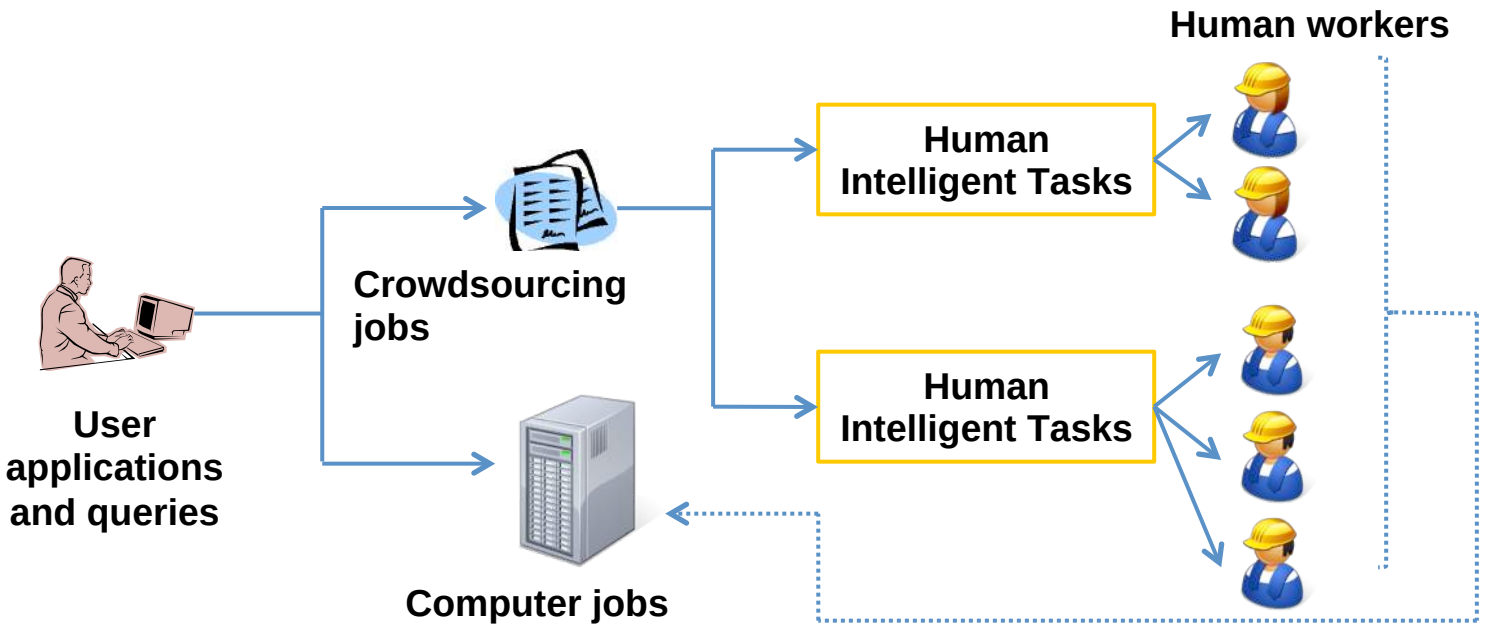
The New-York State Yacht Squadron, on its annual cruise to Newport, came into the harbor yesterday afternoon. The following are the names of the boats that came to anchor here: *Jessie*, *Geraldine*, *Evelyn*, *Annie*, *Mannering*, *Julia*, *Bonita*, *Magic*, *Wildgoon*, *Rambler*, *Fleur-de-Lis*, *Henrietta*, *Sea-Drift* and *Maria*, with the steamer *America* as a tender. On anchoring, each boat fired a gun, according to custom. The reports were heard distinctly in the city, causing considerable inquiry as to "what was up," and quite a number of sanguine individuals came into our office to inquire if the guns were not annunciatory signals of the successful laying of the Atlantic Cable. We invariably replied in the negative. The squadron will leave to-day for Newport. The yachts *Washington* and *Rabbler*, of this city, start with it, with parties of New-Haven people.

reCAPTCHA Transcription

The New-York State yacht Squadron, on its annual cruise to Newport came into the harbor yesterday afternoon. The following are the names of the boats that came to anchor here: Jessie, gera loliv erelun Annie, Mannering, Julia, Bonita, Magic wut, Rambler, floumblie, Henrietta, Sea-Drift and Maria, with the steamer America as a tender. On anchoring each boat fired a gun, according to custom. The reports were heard distinctly in the city, causing considerable inquiry as to "what was up," and quite a number of sanguine individuals came into our office to inquire if the guns were not annunciatory signals of the successful laying of the Atlantic Cable. We invariably replied in the negative. The squadron will leave to-day for Newport. The yachts Washington and buub r of this city, start with it, with parties of New Haven people.

Harness Human Intelligence

- Work flow of crowdsourcing applications



Crowdsourcing Platforms

amazon mechanical turk™
Artificial Artificial Intelligence

clickworker™

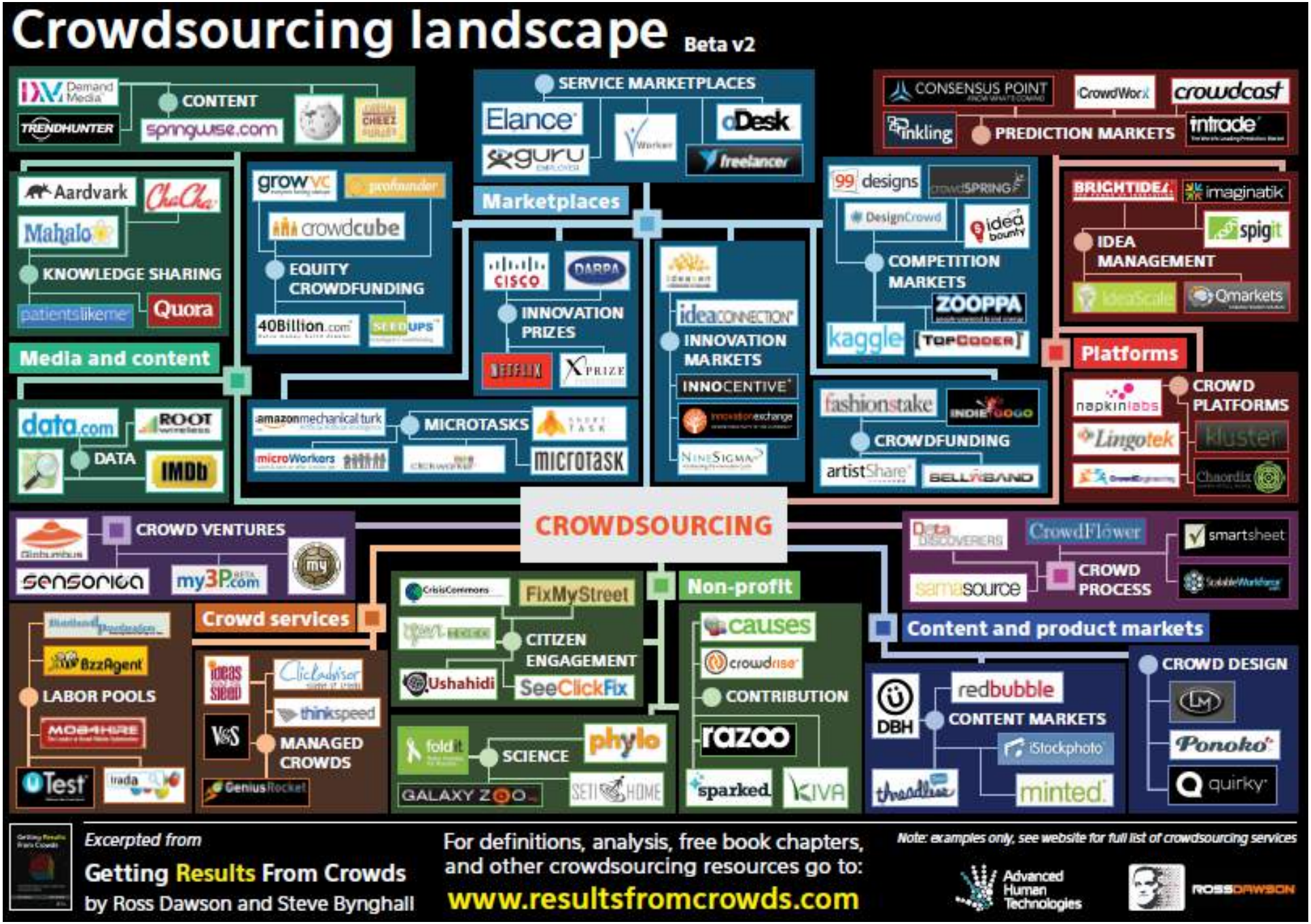
CloudCrowd
We're working on it. Lots of us.

samasource

microtask

CROWDGURU
Crowdsourcing for business.

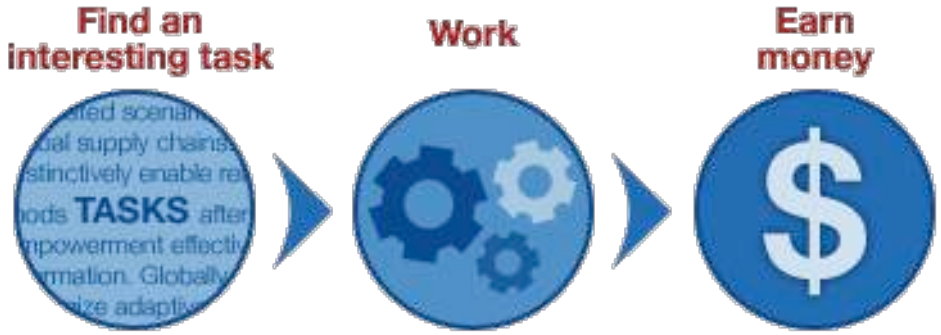
CrowdFlower



../features/crowdsourcing-landscape/ ../features/crowdsourcing-services/

Amazon Mechanical Turk (AMT)

worker



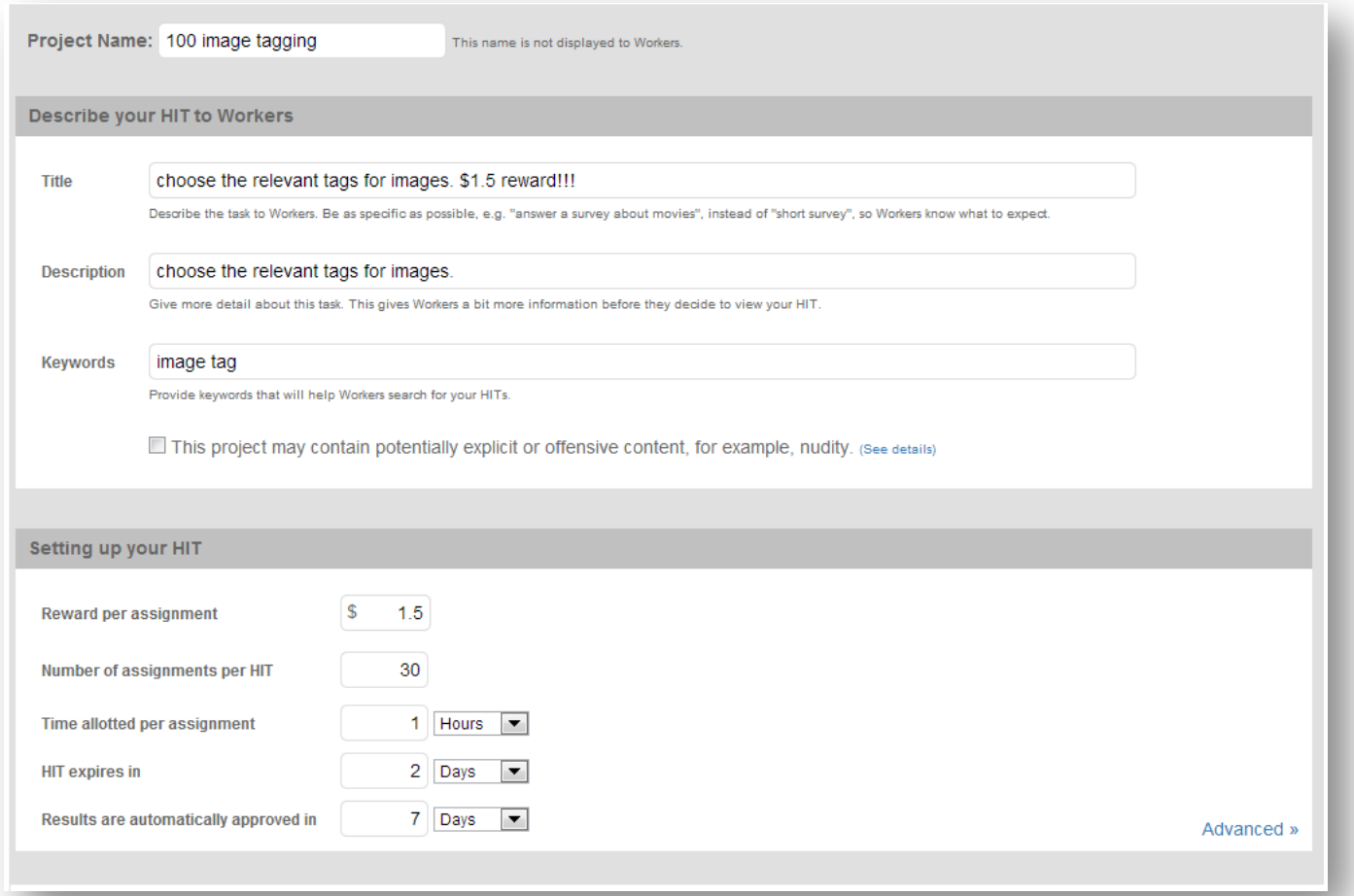
Make Money
by working on HITs

Get Results
from Mechanical Turk Workers

requester

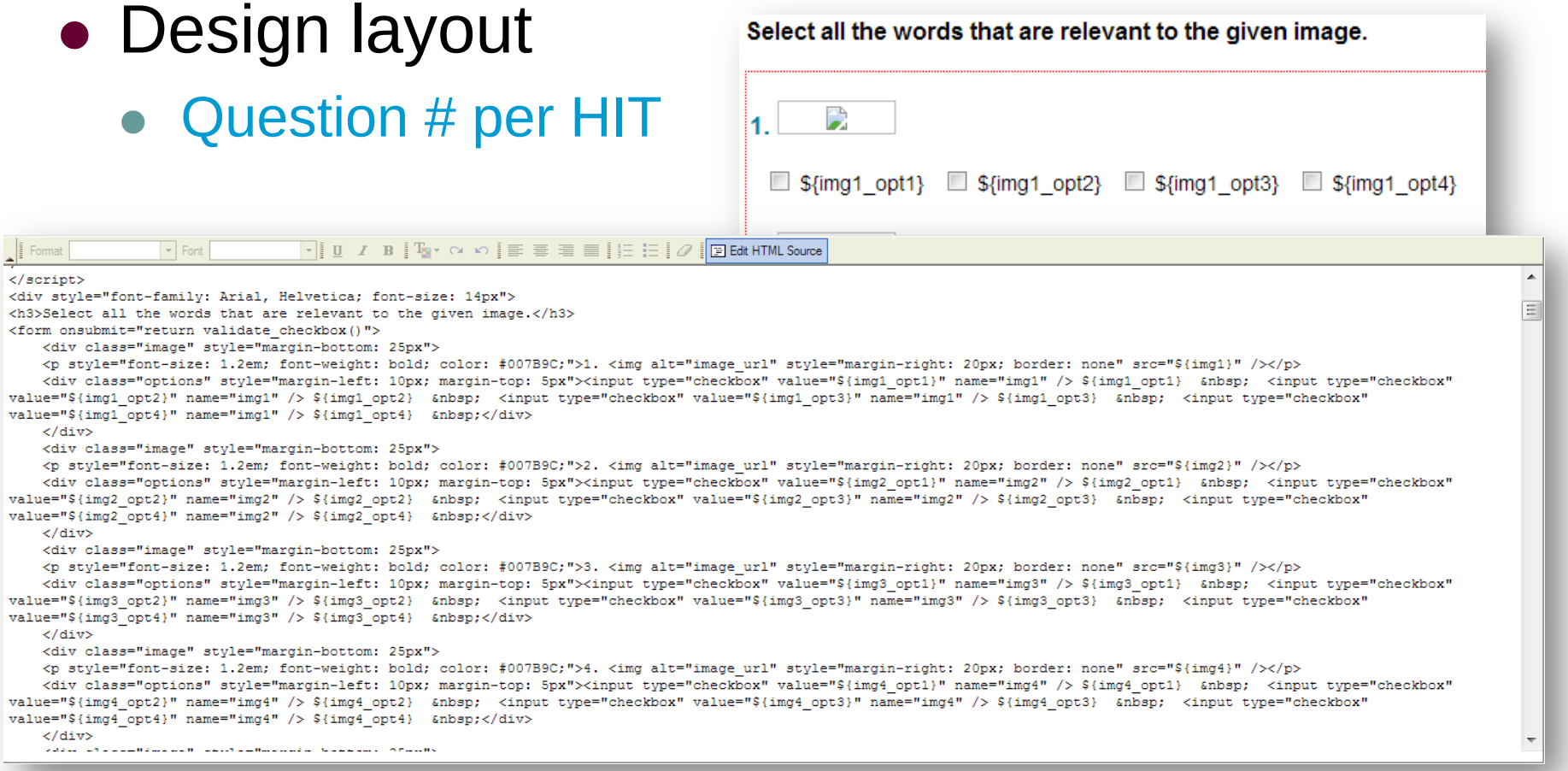
Publish HITs on AMT

- Set up your HIT
  - Assignment # per HIT



Publish HITs on AMT

- Set up your HIT
  - Assignment # per HIT
- Design layout
  - Question # per HIT




```
</script>
<div style="font-family: Arial, Helvetica; font-size: 14px">
<h3>Select all the words that are relevant to the given image.</h3>
<form onsubmit="return validate_checkbox()">
    <div class="image" style="margin-bottom: 25px">
    <p style="font-size: 1.2em; font-weight: bold; color: #007B9C;">1. <img alt="image_url" style="margin-right: 20px; border: none" src="${img1}" /></p>
    <div class="options" style="margin-left: 10px; margin-top: 5px"><input type="checkbox" value="${img1_opt1}" name="img1" /> ${img1_opt1}     <input type="checkbox"
value="${img1_opt2}" name="img1" /> ${img1_opt2}     <input type="checkbox" value="${img1_opt3}" name="img1" /> ${img1_opt3}     <input type="checkbox"
value="${img1_opt4}" name="img1" /> ${img1_opt4}   </div>
    </div>
    <div class="image" style="margin-bottom: 25px">
    <p style="font-size: 1.2em; font-weight: bold; color: #007B9C;">2. <img alt="image_url" style="margin-right: 20px; border: none" src="${img2}" /></p>
    <div class="options" style="margin-left: 10px; margin-top: 5px"><input type="checkbox" value="${img2_opt1}" name="img2" /> ${img2_opt1}     <input type="checkbox"
value="${img2_opt2}" name="img2" /> ${img2_opt2}     <input type="checkbox" value="${img2_opt3}" name="img2" /> ${img2_opt3}     <input type="checkbox"
value="${img2_opt4}" name="img2" /> ${img2_opt4}   </div>
    </div>
    <div class="image" style="margin-bottom: 25px">
    <p style="font-size: 1.2em; font-weight: bold; color: #007B9C;">3. <img alt="image_url" style="margin-right: 20px; border: none" src="${img3}" /></p>
    <div class="options" style="margin-left: 10px; margin-top: 5px"><input type="checkbox" value="${img3_opt1}" name="img3" /> ${img3_opt1}     <input type="checkbox"
value="${img3_opt2}" name="img3" /> ${img3_opt2}     <input type="checkbox" value="${img3_opt3}" name="img3" /> ${img3_opt3}     <input type="checkbox"
value="${img3_opt4}" name="img3" /> ${img3_opt4}   </div>
    </div>
    <div class="image" style="margin-bottom: 25px">
    <p style="font-size: 1.2em; font-weight: bold; color: #007B9C;">4. <img alt="image_url" style="margin-right: 20px; border: none" src="${img4}" /></p>
    <div class="options" style="margin-left: 10px; margin-top: 5px"><input type="checkbox" value="${img4_opt1}" name="img4" /> ${img4_opt1}     <input type="checkbox"
value="${img4_opt2}" name="img4" /> ${img4_opt2}     <input type="checkbox" value="${img4_opt3}" name="img4" /> ${img4_opt3}     <input type="checkbox"
value="${img4_opt4}" name="img4" /> ${img4_opt4}   </div>
    </div>
```

Publish HITs on AMT

- Set up your HIT
  - Assignment # per HIT
- Design layout
  - Question # per HIT
- Prepare and upload data file

Publish HITs on AMT

- Set up your HIT
  - Assignment # per HIT
- Design layout
  - Question # per HIT
- Prepare and upload data file
- Wait for the result

Publish HITs on AMT

- Set up your HIT
  - Assignment # per HIT
- D
  -
- P
- Wait for the result
- Download, parse and analyze the result
  - Conflicting answers

Answer.m1_r1	Answer.m1_r2	Answer.m1_r3	Answer.m1_r4	Answer.m1_r5
Negative	Negative	Positive	Positive	Positive
Negative	Positive	Positive	Positive	Positive
Positive	Negative	Positive	Negative	Positive
Positive	Positive	Positive	Positive	Positive
Negative	Negative	Positive	Positive	Positive

Need from Crowdsourcing

- A “smarter” system that takes care of
  - Layout design
  - HIT generation
  - Worker assignment
  - Result analysis
  - Declarative query
  - Optimization
  - More…

MultiChoice (*$image, $tags*)

Data

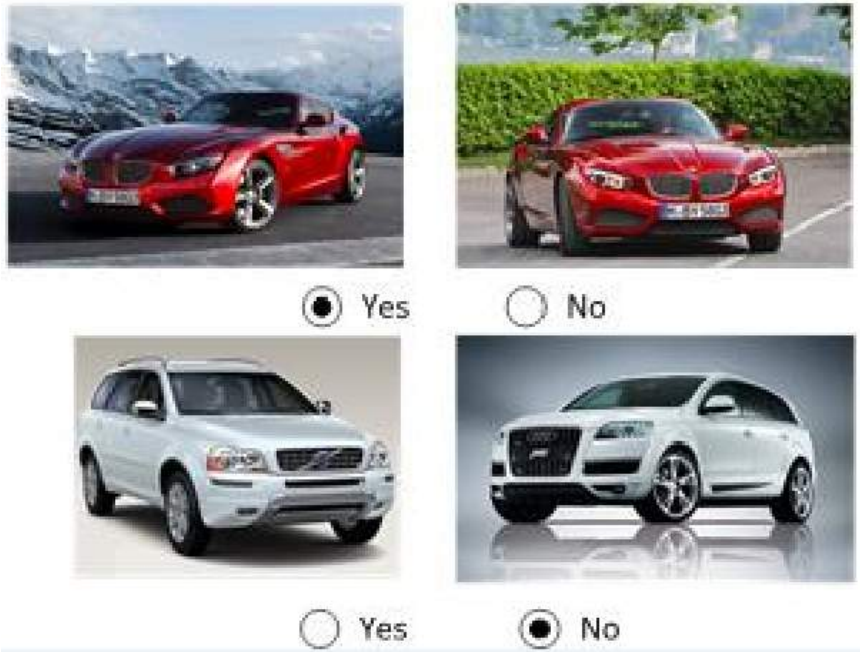
ID	Image	Tags
1	i1	t1
…	…	…

Need from DB Systems

Find CEO and HQ for a list of companies?

Company	CEO	Headquarter
IBM		
Google		
Microsoft		
Apple		
Oracle		

Are they the same car?



- Crowd helps on DB-hard queries
  - Seeking information
  - Adding intelligence

Crowdsourcing DB Systems

- Add crowd functionality into traditional databases for processing queries that cannot be easily answered by machines

- Automate certain tasks and provide higher-level services than existing platforms

Challenges of Crowdsourcing

- Quality
  - Human workers are prone to errors, and crowdsourcing results are inevitably noisy
    - Maliciousness, worker skills, task difficulty
- Cost
  - Very expensive to publish a huge amount of tasks
    - E.g., Finding matched pairs from 1000 records
    - Straightforward: 1000 * 1000 = 1M tasks
    - $ 0.01 per task * 1M tasks = $ 10000 !
- Latency
  - It depends on worker pool and job attractiveness
  - Dependency among tasks

New Challenges to DB Systems

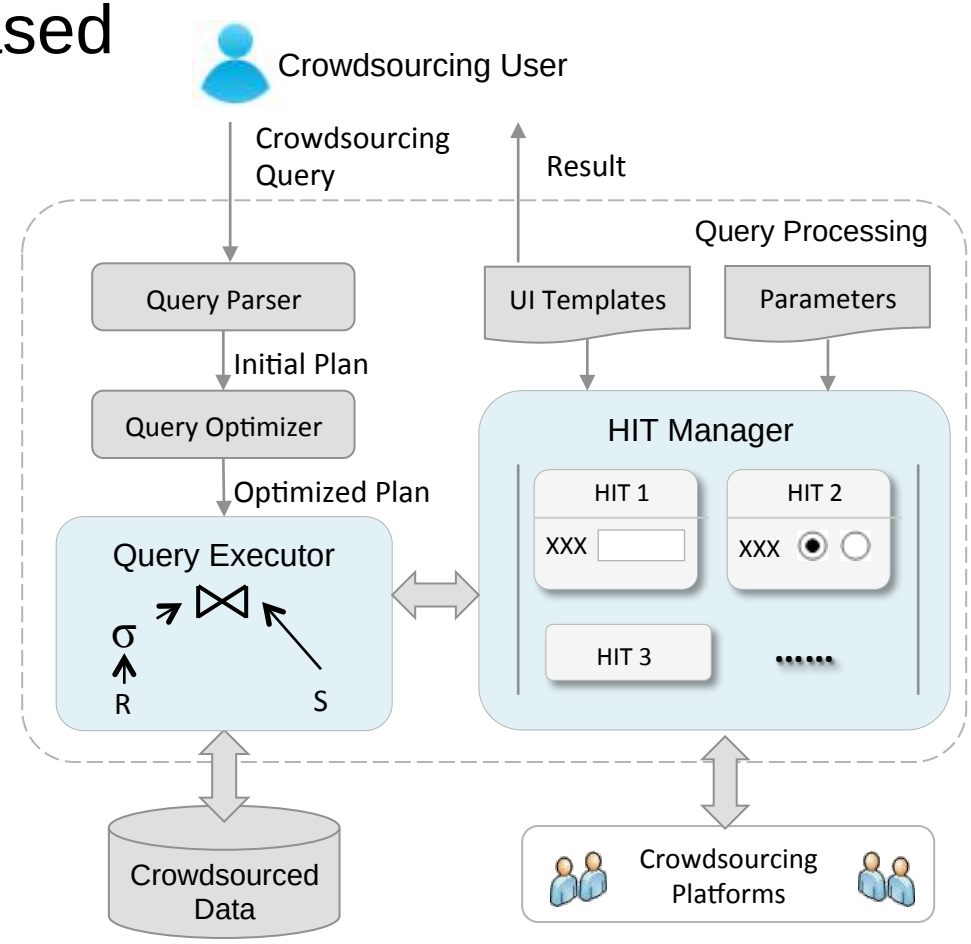
- Closed world → open world
  - Unbounded amount of data available
- Non-deterministic nature
  - Human factors → uncertainty
- Trade-off among cost, latency and quality
  - Optimizations

Crowdsourcing DB Systems

- **Qurk – MIT**
  - [Marcus-CIDR2011]
  - [Marcus-SIGMOD2011]
  - [Marcus-VLDB2012]
- **CrowdDB – Berkeley and ETH**
  - [Feng-VLDB2011]
  - [Franklin-SIGMOD2011]
- **Deco – Stanford and UCSC**
  - [Parameswaran-CIKM2012]
  - [Park-VLDB2013]
- **CDAS – NUS**
  - [Liu-VLDB2012]
  - [Gao-SIGMOD2013]
  - [Ooi-SIGKDD2014]

Crowdsourcing DB Systems

- Data model
  - Relational model based
- Query language
  - Declarative
  - SQL-like
- Query processing and optimization



System Architecture

Crowdsourcing DB Systems

- **Qurk – MIT**
  - [Marcus-CIDR2011]
  - [Marcus-SIGMOD2011]
  - [Marcus-VLDB2012]
- **CrowdDB – Berkeley and ETH**
  - [Feng-VLDB2011]
  - [Franklin-SIGMOD2011]
- **Deco – Stanford and UCSC**
  - [Parameswaran-CIKM2012]
  - [Park-VLDB2013]
- **CDAS – NUS**
  - [Liu-VLDB2012]
  - [Gao-SIGMOD2013]
  - [Ooi-SIGKDD2014]

Qurk: Data Model

- Relational model + a few “twists”
- Qurk allows multi-valued attributes
  - Different responses to the same HIT
- Qurk provides functions to convert multi-valued attributes to usable single-valued fields
  - E.g, majorityVote

Qurk: Query Language

- SQL-based query language with user-defined functions (UDFs)
- Predefined task templates for UDF:
  - Filter: produces tuples that satisfy the conditions specified in the UDF
  - Sort: ranks the input tuples according to the UDFs specified in the ORDER BY clause
  - Join: compares input tuples and performs join according to the UDF; extends syntax with a POSSIBLY keyword
  - Generative: allows workers to generate data

Qurk: Filter Example

```
SELECT image
FROM car_image
WHERE isBlack(image)

TASK isBlack(field) TYPE Filter:
  Prompt: "<table><tr> \
       <td><img src='%s'></td> \
       <td>Is the car in the image in black color?</td>
       </tr></table>", tuple[field]
  YesText: "Yes"
  NoText: "No"
  Combiner: MajorityVote
```

Qurk: Sort Example

```
SELECT squares.label
FROM squares
ORDER BY squareSort(img)

TASK squareSort(field) TYPE Rank:
  SingularName: "square"
  PluralName: "squares"
  OrderDimensionName: "area"
  LeastName: "smallest"
  MostName: "largest"
  Html: "<img src='%s' class=lgImg>",tuple[field]
```

Adam Marcus, Eugene Wu, David R. Karger, Samuel Madden, Robert C. Miller: Human-powered Sorts and Joins. PVLDB 5(1): 13-24 (2012)

Qurk: Join Example

```
SELECT c.name
FROM celeb c JOIN photos p
ON samePerson(c.img,p.img)

TASK samePerson(f1, f2) TYPE EquiJoin:
  SingluarName: "celebrity"
  PluralName: "celebrities"
  LeftPreview: "<img src='%s' class=smImg>",tuple1[f1]
  LeftNormal: "<img src='%s' class=lgImg>",tuple1[f1]
  RightPreview: "<img src='%s'class=smImg>",tuple2[f2]
  RightNormal: "<img src='%s' class=lgImg>",tuple2[f2]
  Combiner: MajorityVote
```

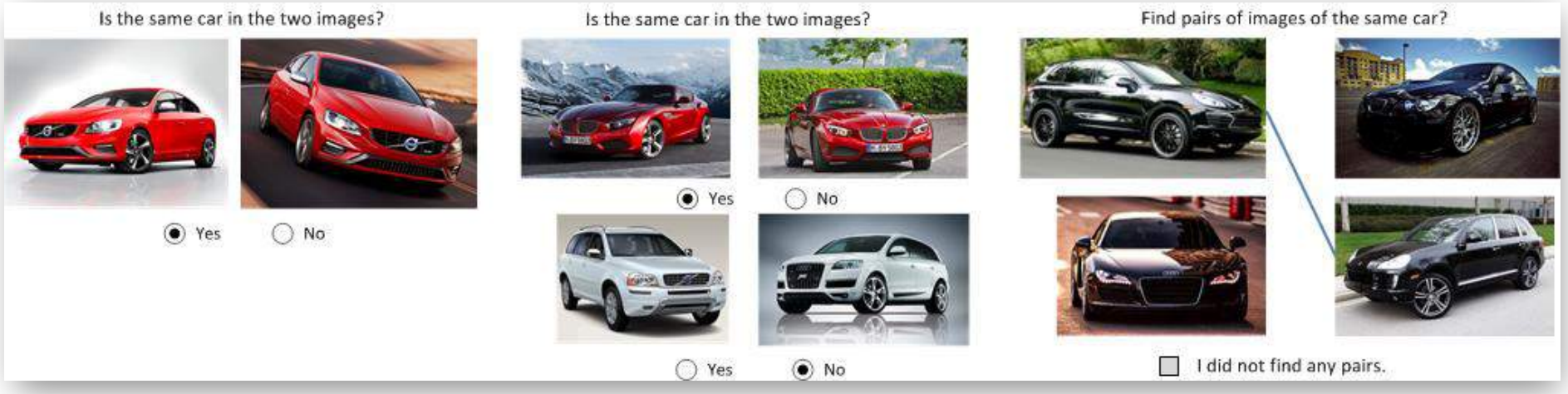
Adam Marcus, Eugene Wu, David R. Karger, Samuel Madden, Robert C. Miller: Human-powered Sorts and Joins. PVLDB 5(1): 13-24 (2012)

Qurk: Generative Example

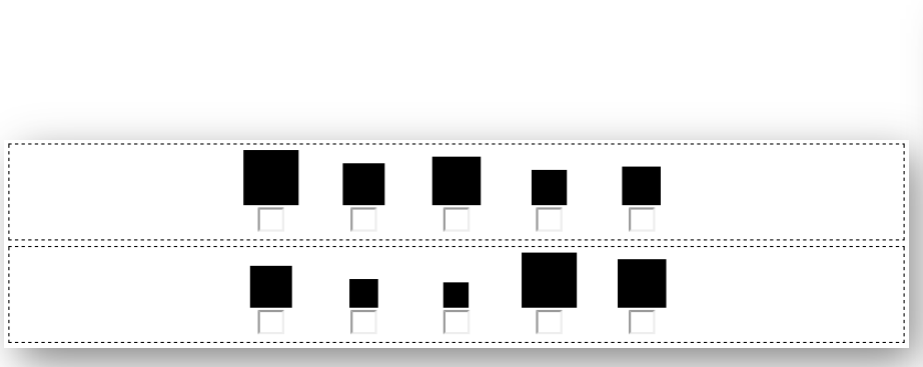
```
SELECT c.name
FROM celeb c JOIN photos p
ON samePerson(c.img,p.img)
AND POSSIBLY gender(c.img) = gender(p.img)
AND POSSIBLY hairColor(c.img) = hairColor(p.img)
AND POSSIBLY skinColor(c.img) = skinColor(p.img)

TASK gender(field) TYPE Generative:
  Prompt: "<table><tr> \
                    <td><img src='%s'> \
                    <td>What is this person's gender? \
            </table>", tuple[field]
  Response: Radio("Gender",["Male","Female",UNKNOWN])
```

Adam Marcus, Eugene Wu, David R. Karger, Samuel Madden, Robert C. Miller: Human-powered Sorts and Joins. PVLDB 5(1): 13-24 (2012)

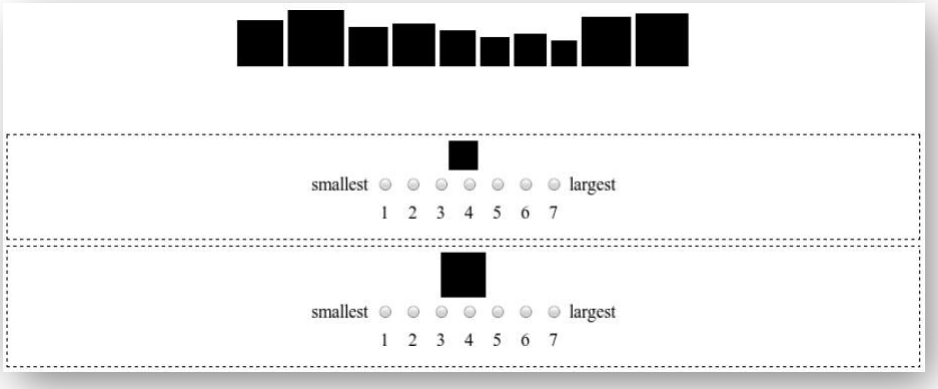
Qurk: Operators



Simple Join **Naïve batching Join** **Smart Batching Join**



Comparison-based Sort



Rating-based Sort

Adam Marcus, Eugene Wu, David R. Karger, Samuel Madden, Robert C. Miller: Human-powered Sorts and Joins. PVLDB 5(1): 13-24 (2012)

Qurk: Optimizations

- Runtime pricing
- Input sampling
- Batch predicates
- Operator implementations
- Join heuristics
- Task result cache
- Model training

Crowdsourcing DB Systems

- **Qurk – MIT**
  - [Marcus-CIDR2011]
  - [Marcus-SIGMOD2011]
  - [Marcus-VLDB2012]
- **CrowdDB – Berkeley and ETH**
  - [Feng-VLDB2011]
  - [Franklin-SIGMOD2011]
- **Deco – Stanford and UCSC**
  - [Parameswaran-CIKM2012]
  - [Park-VLDB2013]
- **CDAS – NUS**
  - [Liu-VLDB2012]
  - [Gao-SIGMOD2013]
  - [Ooi-SIGKDD2014]

CrowdDB: Query Language

- CrowdSQL: a new keyword CROWD
- Crowdsourced column:

```
CREATE TABLE car_review (
  review STRING,
  make CROWD STRING,
  model CROWD STRING,
  sentiment CROWD STRING);
```

- Crowdsourced table:

```
CREATE CROWD TABLE car (
  make STRING,
  model STRING,
  color STRING,
  style STRING,
  PRIMARY KEY (make, model));
```

CrowdDB: Query Language

- CrowdSQL: two new built-in functions

- CROWDEQUAL: ~= symbol

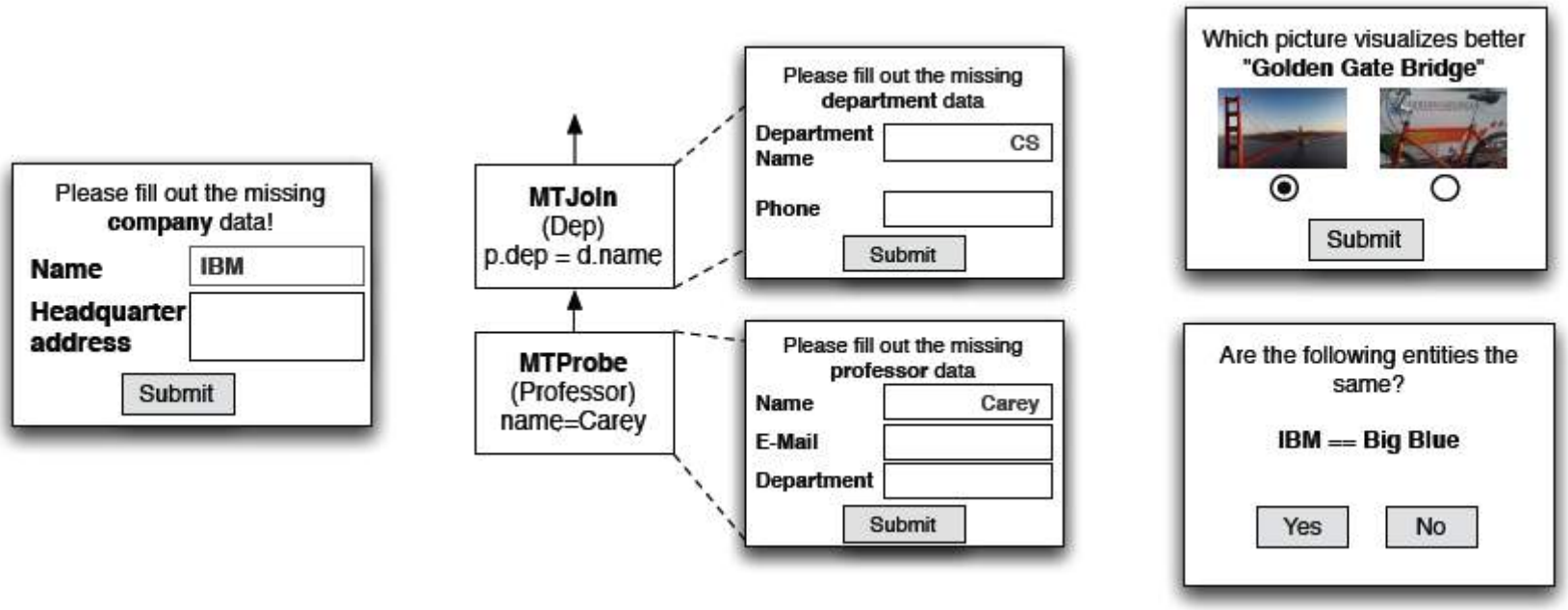
```
SELECT review FROM car_review
WHERE sentiment ~= "pos";
```

- CROWDORDER:

```
SELECT image i FROM car_image
WHERE subject = "Volvo S60"
ORDER BY CROWDORDER(i, "Which image visualizes
better %subject");
```

CrowdDB: Operators

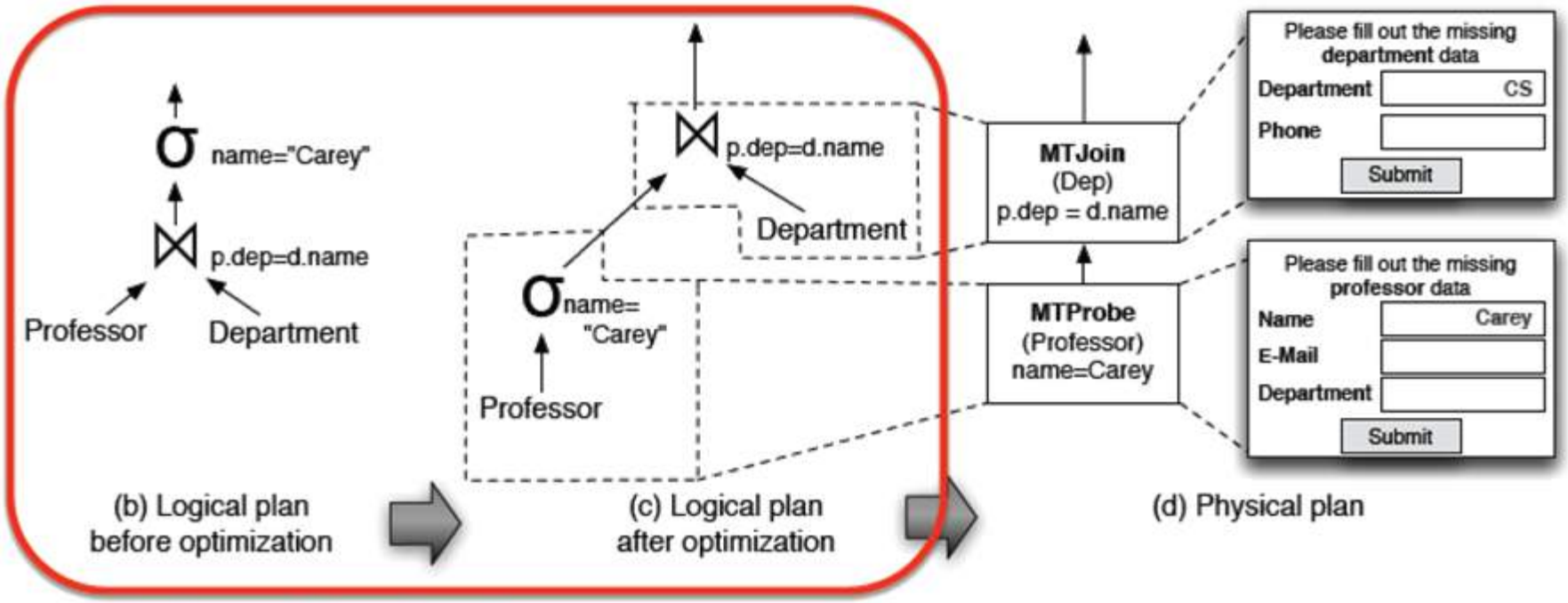
- CrowdProbe
  - Collects missing information
- CrowdJoin
  - Inner relation is a CROWD table
- CrowdCompare
  - Implements CROWDEQUAL and CROWDORDER



CrowdDB: Example

```
SELECT *
FROM PROFESSOR p, DEPARTMENT d
WHERE d.name = p.dep
AND p.name ="Michael J. Carey"
```

```
CREATE CROWD TABLE professor (
    name STRING PRIMARY KEY
    e-mail STRING
    dep STRING
      REF department(name)
);
CREATE CROWD TABLE department (
    name STRING PRIMARY KEY
    phone_no STRING);
```



Rule based optimizer

Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, Reynold Xin: CrowdDB: answering queries with crowdsourcing. SIGMOD Conference 2011: 61-72

CrowdDB: Optimizations

- Rule-based optimizer
  - Predicate push-down
  - Join ordering
  - Bounded plan
- Simple heuristics to set the crowd parameters
  - Price per HIT
  - Batch size
  - Replication factor

Crowdsourcing DB Systems

- **Qurk – MIT**
  - [Marcus-CIDR2011]
  - [Marcus-SIGMOD2011]
  - [Marcus-VLDB2012]
- **CrowdDB – Berkeley and ETH**
  - [Feng-VLDB2011]
  - [Franklin-SIGMOD2011]
- **Deco – Stanford and UCSC**
  - [Parameswaran-CIKM2012]
  - [Park-VLDB2013]
- **CDAS – NUS**
  - [Liu-VLDB2012]
  - [Gao-SIGMOD2013]
  - [Ooi-SIGKDD2014]

Deco: Data Model

- Conceptual relation: user view
  - Anchor attributes (entities)
  - Dependent attributes (properties of entities)
    - Country(country, [language], [capital])
- Raw schema: system view
  - RDBMS as back-end
  - Anchor table containing anchor attributes
  - Dependent table for each dependent attribute
    - CountryA(country),
    - CountryD1(country, language)
    - CountryD2(country, captital)

Deco: Rules

- Fetch rules $A\downarrow 1 \Rightarrow A\downarrow 2 : P$

```
∅ ⇒ country: Ask for a new country name
country ⇒ capital: Ask for the captital given a country name
```

- Resolution rules

```
∅ ⇒ country: dupElim
country ⇒ capital: majority-of-3
```

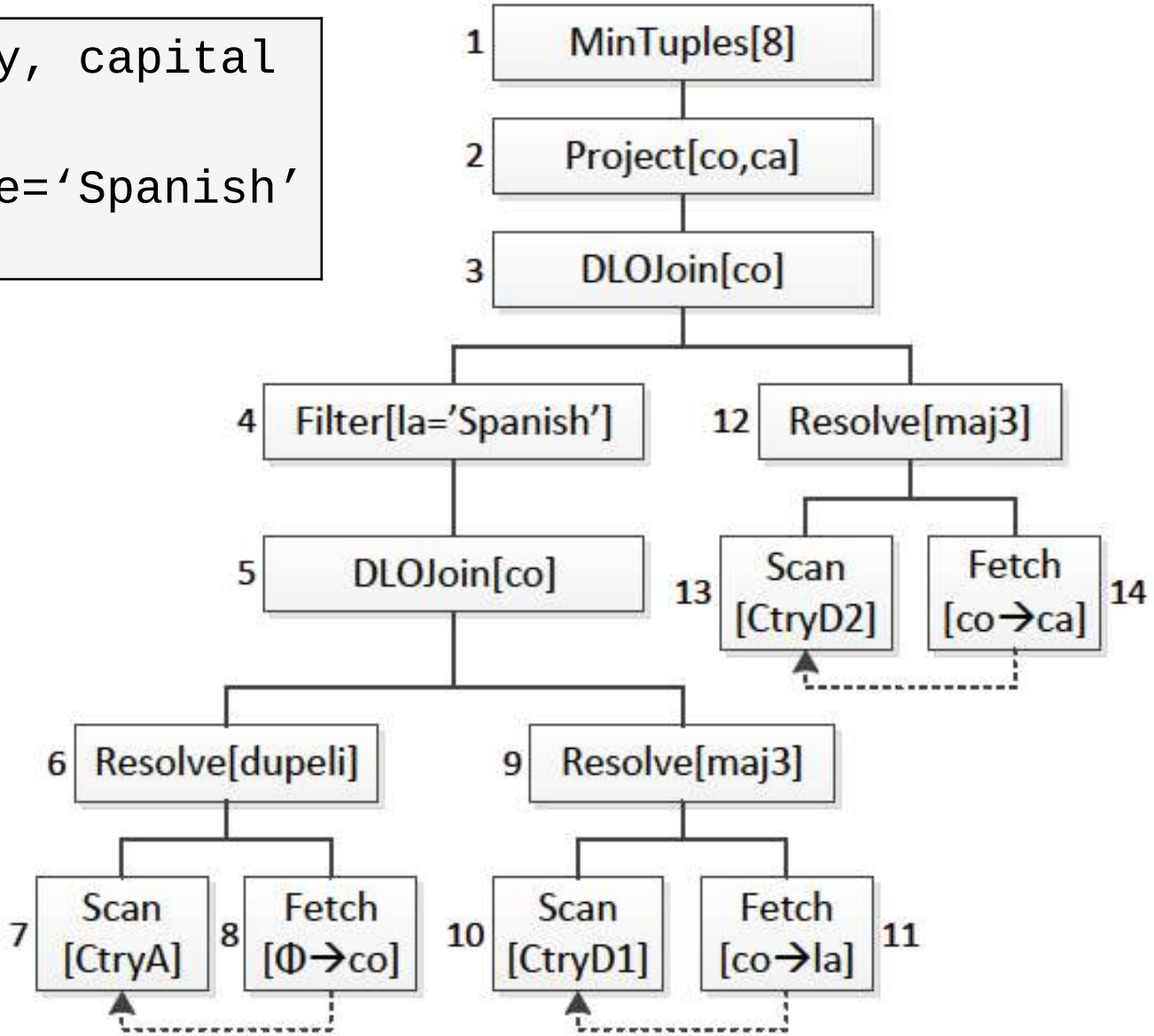
- “MinTuples n” to avoid empty-answer
  - MaxCost c
  - MaxTime t

Deco: Operators

- Scan
- Project
- Filter
- Dependent left outer join
- Fetch
- Resolve
- MinTuples

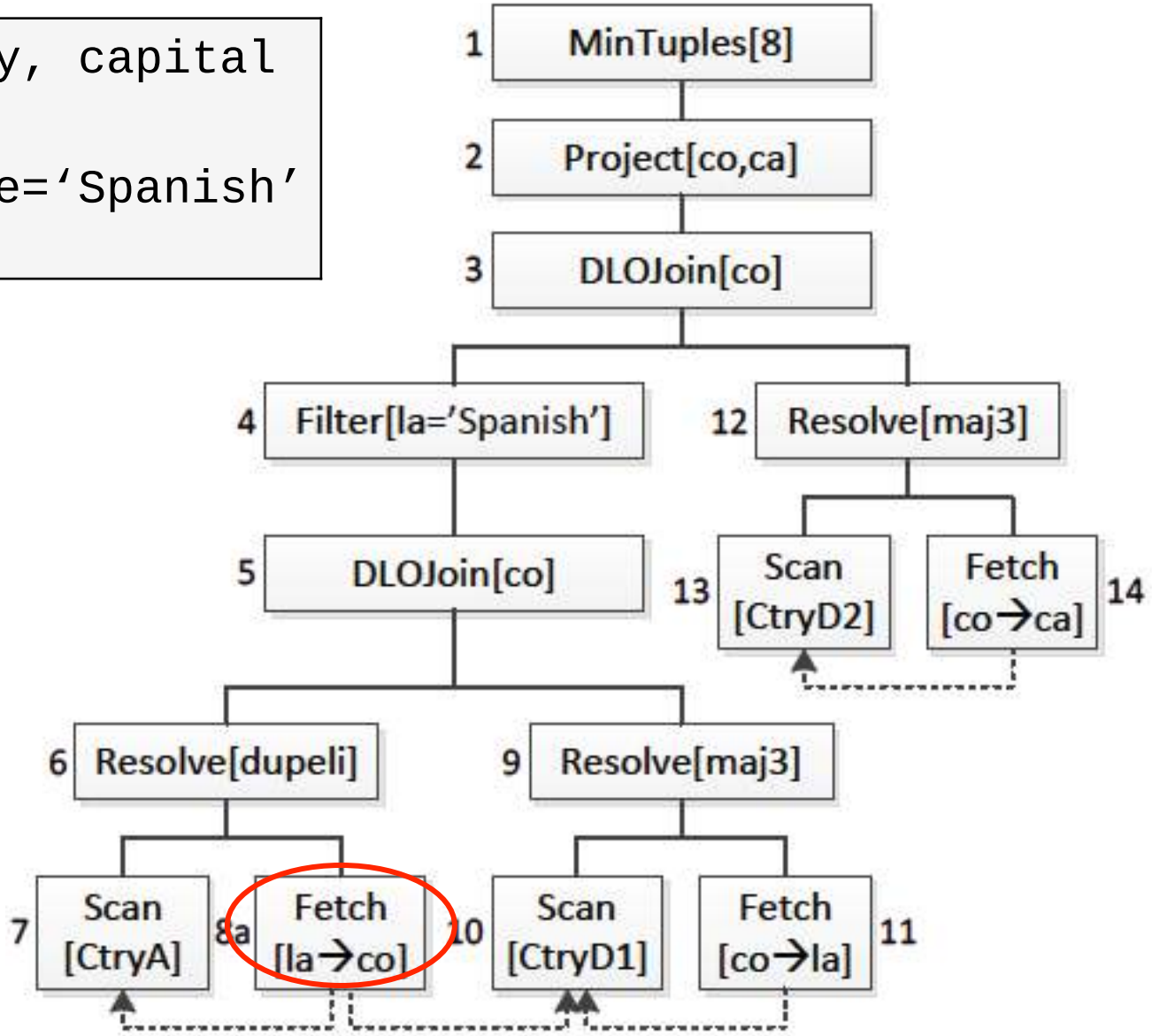
Deco: Example

```
SELECT country, capital
FROM Country
WHERE language='Spanish'
MINTUPLES 8
```



Deco: Example

```
SELECT country, capital
FROM Country
WHERE language='Spanish'
MINTUPLES 8
```



Deco: Optimizations

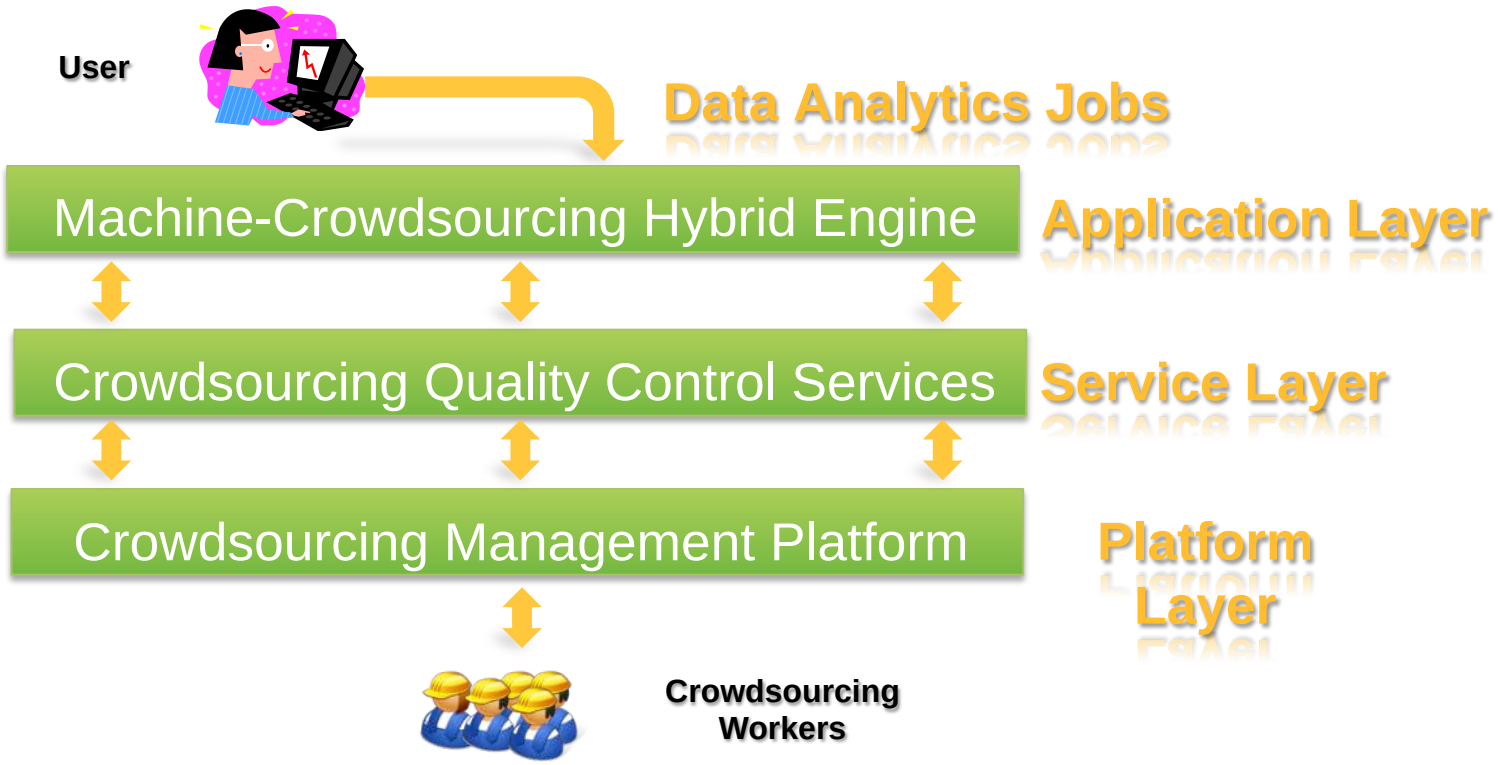
- Existing vs. new data
  - Fetch only if raw tables do not have sufficient data
- Statistical information
  - Existing data: maintained by RDBMS
  - New data: provided by schema designer/user
- Search space
  - Possible join ordering
  - Available fetch rules

Crowdsourcing DB Systems

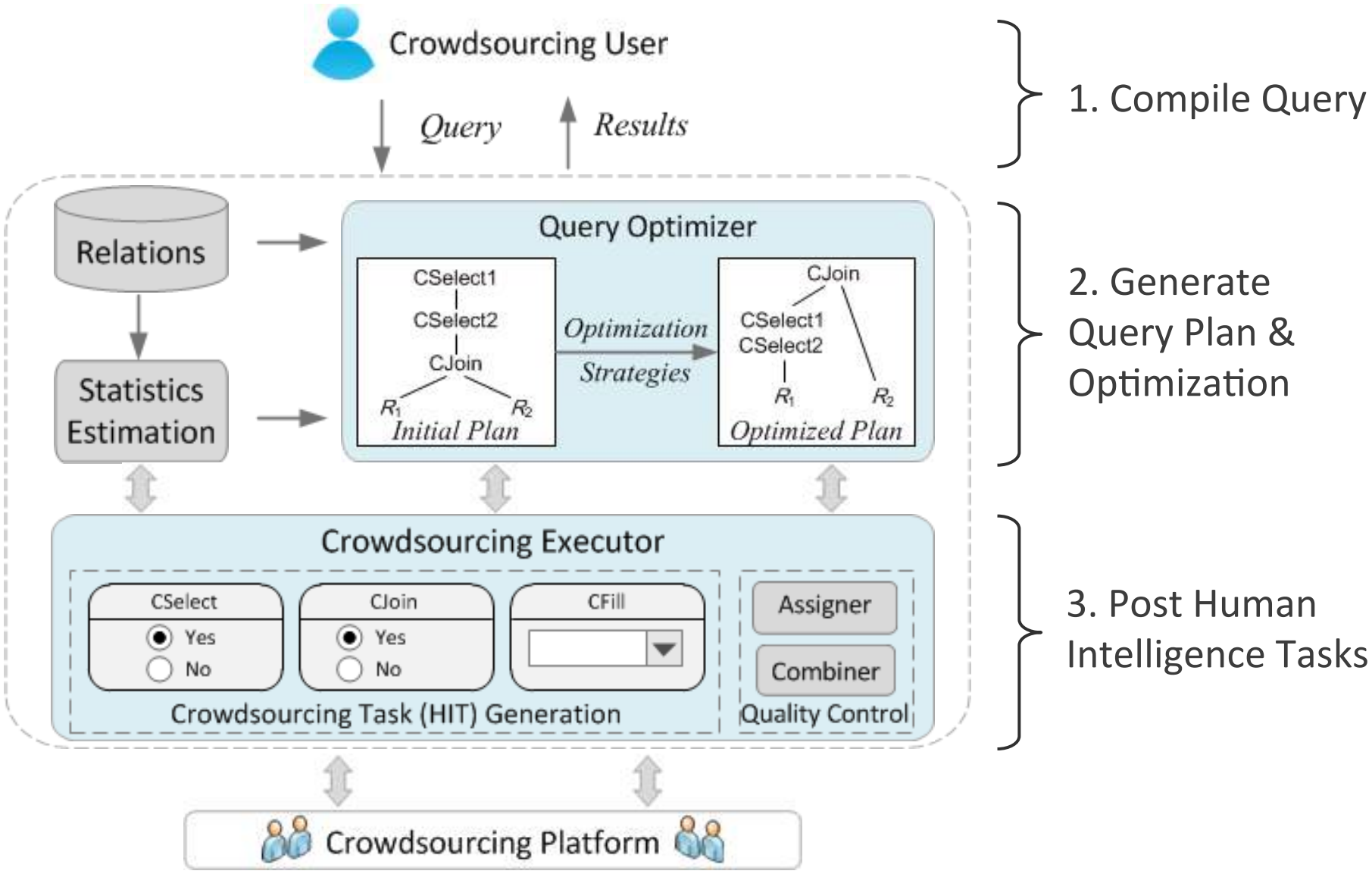
- **Qurk – MIT**
  - [Marcus-CIDR2011]
  - [Marcus-SIGMOD2011]
  - [Marcus-VLDB2012]
- **CrowdDB – Berkeley and ETH**
  - [Feng-VLDB2011]
  - [Franklin-SIGMOD2011]
- **Deco – Stanford and UCSC**
  - [Parameswaran-CIKM2012]
  - [Park-VLDB2013]
- **CDAS – NUS**
  - [Liu-VLDB2012]
  - [Gao-SIGMOD2013]
  - [Ooi-SIGKDD2014]

CDAS: Overview

- CDAS: Crowdsourcing Data Analytics System
- http://www.comp.nus.edu.sg/~cdas/
- Aim: exploit the crowd intelligence for improving the performance of data analytics jobs



CDAS: Overview



CDAS: Query Language

- Selection query

```
SELECT i.image
FROM car_image i
WHERE make = “Volvo”   AND color = “black” AND quality = “high”
```

- Join query

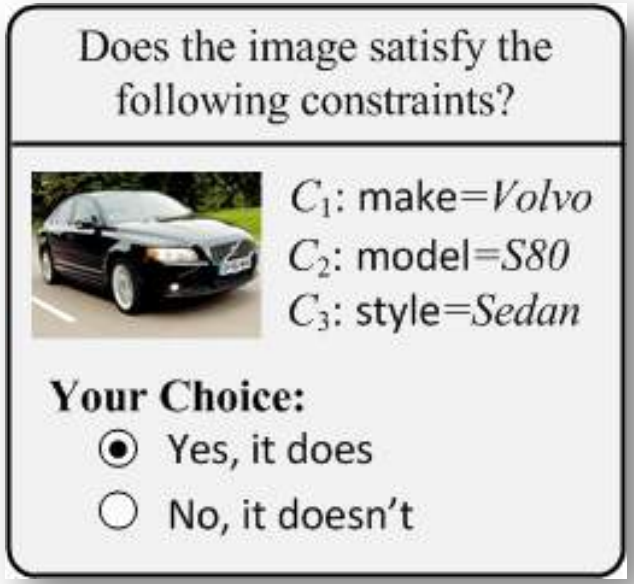
```
SELECT c.*, i.image
FROM car c, car_image i
WHERE c.make = i.make AND c.model = i.model
JoinFilter c.style = i.style
```

- Complex query

```
SELECT c.*, i.image, r.review
FROM car c, car_image i, car_review r
WHERE r.sentiment = “pos” AND i.color = “black”
AND c.make = i.make AND c.model = i.model
AND c.make = r.make AND c.model = r.model
```

CDAS: Operators

- CrowdSelect (CSelect)
  - Leverage the crowd to select the items satisfying certain constraints



– **Operator Input**
  - A set of items
  - Some constraints

– **Operator Output**
  - The subset of items satisfying the constraints

CDAS: Operators

- CrowdJoin (CJoin)
  - Solicit the crowd to match the items according to some matching criteria

– **Operator Input**
  - Two sets of items
  - Matching criteria

– **Operator Output**
  - The item pairs satisfying the criteria

CDAS: Operators

- CrowdFill (CFill)
  - Ask the crowd to fill in missing values for some item attributes

– **Operator Input**
  - A set of items
  - An attribute of interest

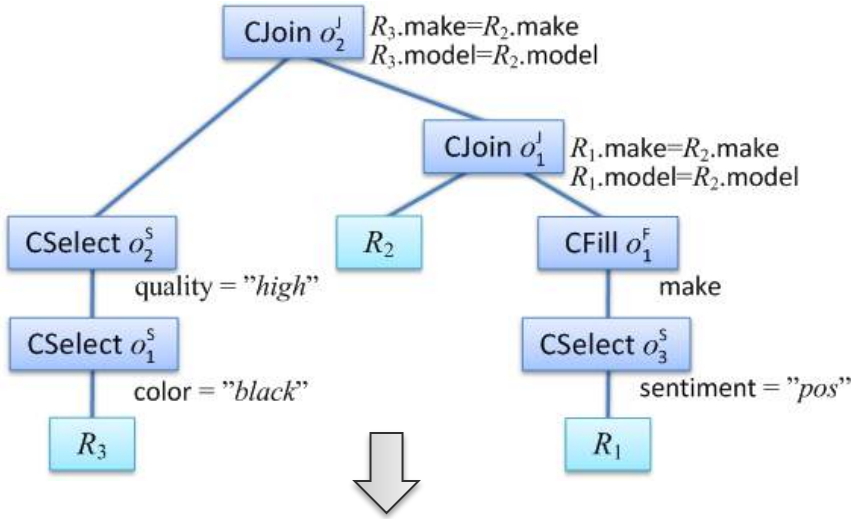
– **Operator Output**
  - The filled values for the attributes of the items

CDAS: Example

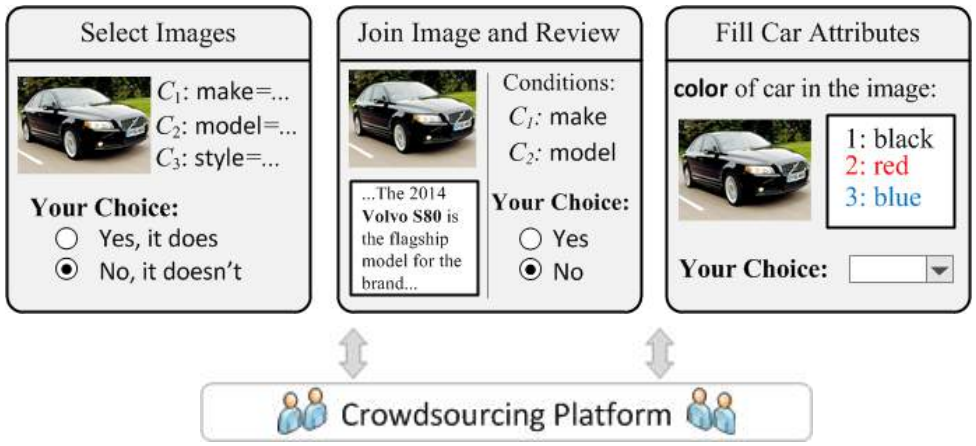
SQL Statement

```
Q1:
SELECT   R2.*, R1.review, R3.image
FROM     REVIEW R1, AUTOMOBILE R2, IMAGE R3
WHERE    R1.sentiment ="pos"
AND      R3.color ="black" AND R3.quality ="high"
AND      R1.make =R2.make AND R1.model =R2.model
AND      R2.make =R3.make AND R2.model =R3.model
```

Query Plan



Human Intelligence Tasks



Crowdsourcing Result

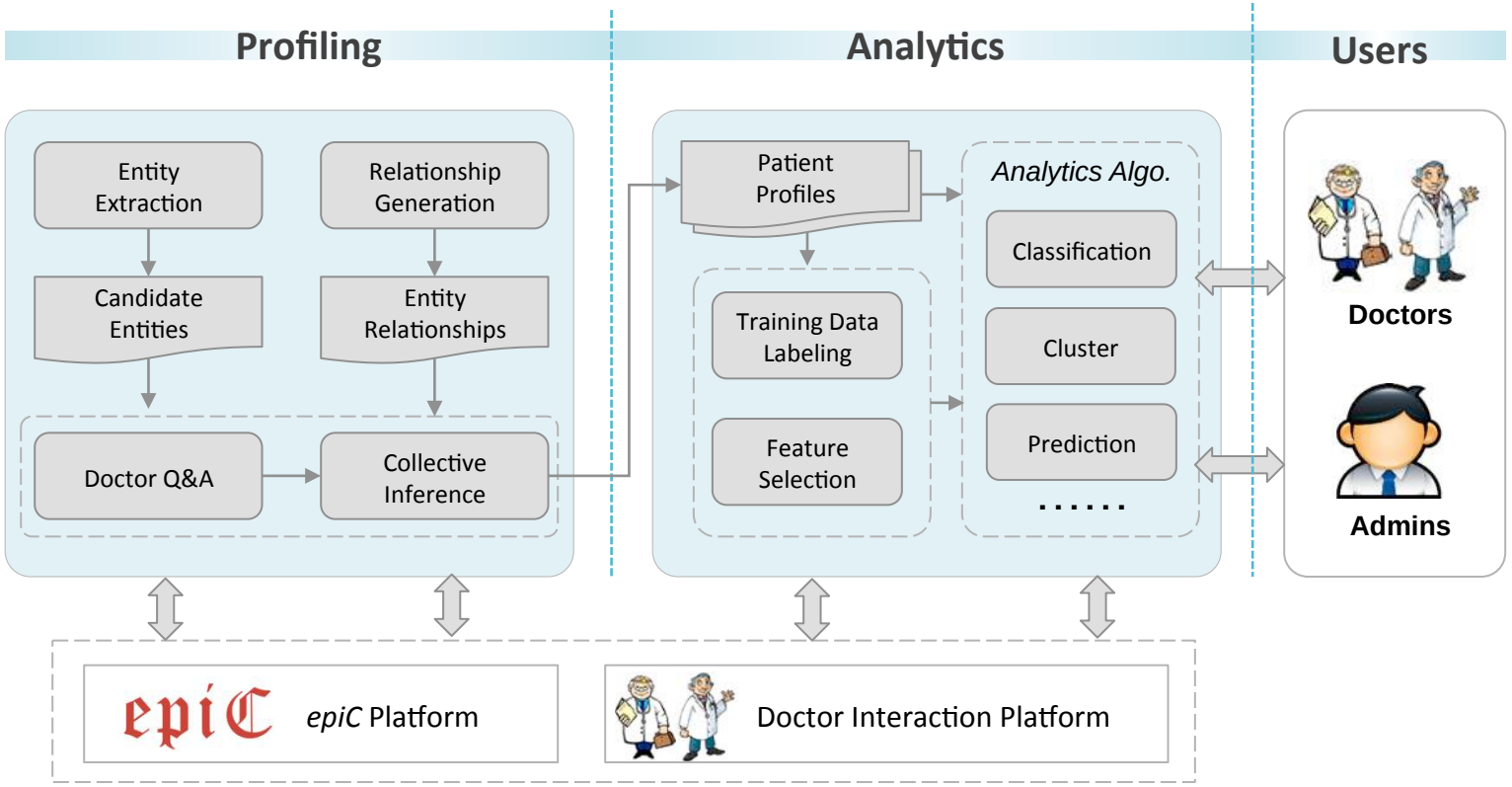
Make	Model	Review	...
Volvo	S80	...	...
...	...	...	...
...	...	...	...

CDAS: Optimizations

- Objective I: Cost Minimization
  - Find the query plan that minimizes the overall cost for evaluating a crowdsourcing query
- Objective II: Latency Bounded Cost Minimization
  - Latency cannot exceed a constraint
  - Minimizing the query plan with latency bound and minimum overall monetary cost

GEMINI: **GE**neralizable Medical **I**nformation a**N**alysis and **I**ntegration System

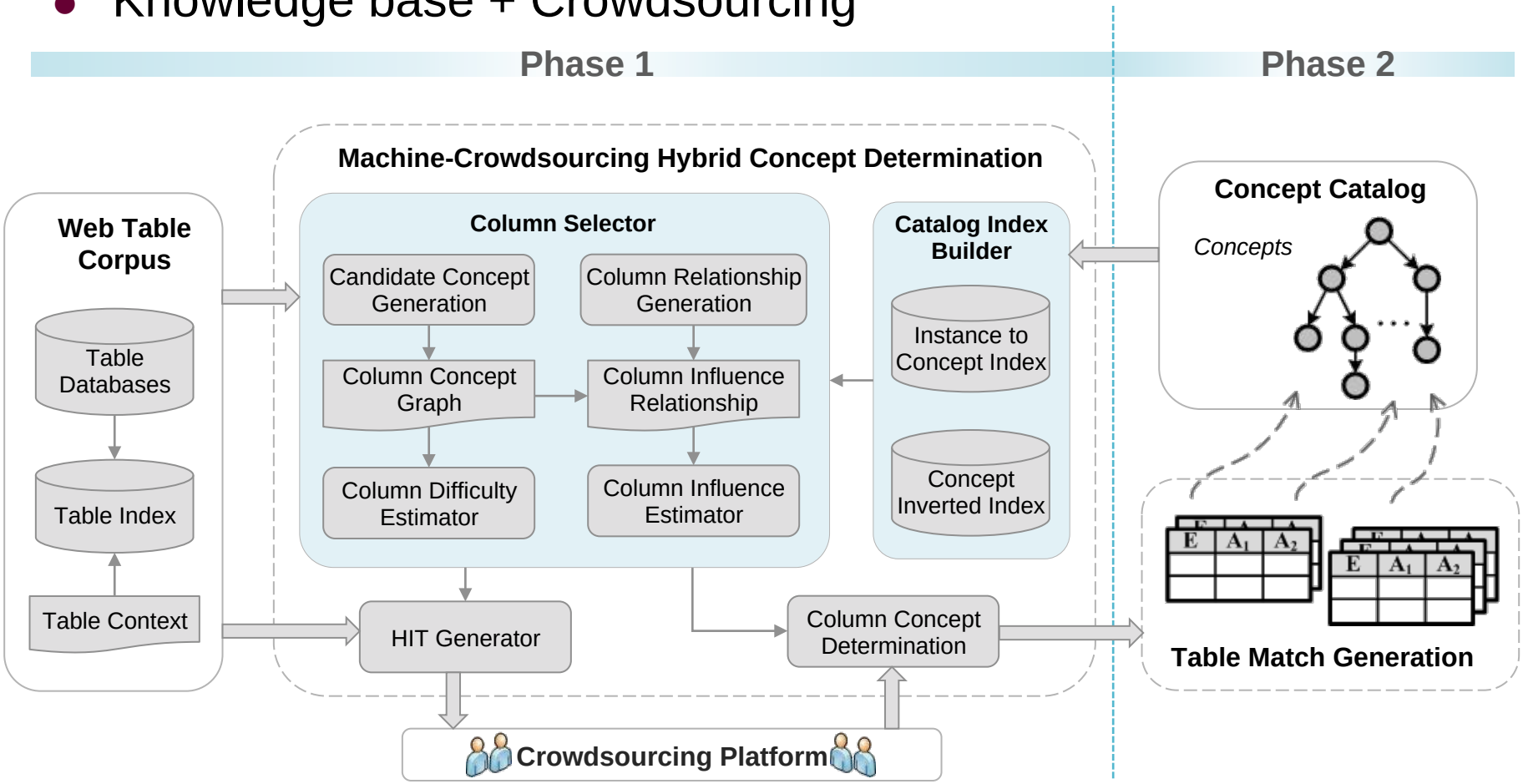
- Hybrid human-machine intelligent data management system optimized for healthcare domain
- To provide comprehensive holistic views of patients and supports real-time analytics



B. C. Ooi, K.-L. Tan, Q.T. Tran, J.W.L. Yip, G. Chen, Z.J. Ling, T. Nguyen, A.K.H. Tung, M. Zhang: Contextual Crowd Intelligence. SIGKDD Exploration, 2014 .

A Hybrid Machine-Crowdsourcing Solution for Web Table Integration

- Hybrid human-machine intelligent system optimized for web table integration
- Knowledge base + Crowdsourcing



Ju Fan, Meiyu Lu, Beng Chin Ooi, Wang-Chiew Tan, Meihui Zhang: A hybrid machine-crowdsourcing system for matching web tables. ICDE 2014: 976-987

Part 1 Summary

- Crowdsourcing is a novel paradigm to help solve DB-hard problems

- Crowdsourcing database systems
  - Qurk, CrowdDB, Deco, CDAS
  - Data model
  - Query language
  - Crowd operators
  - Optimization issues

Reference

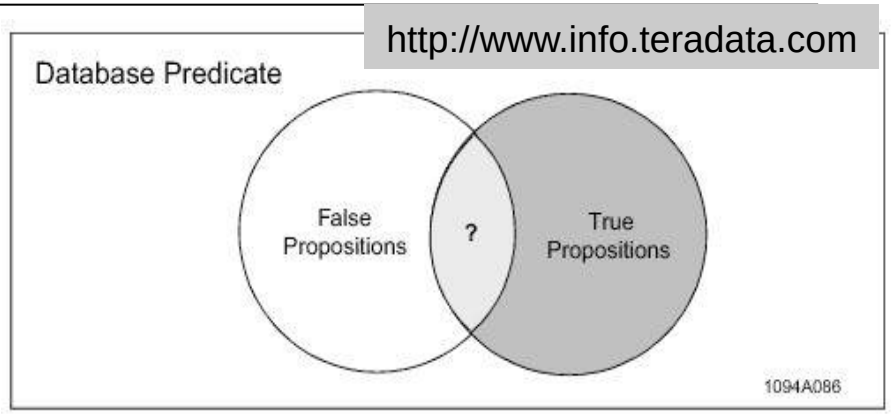
- **[Fan-ICDE2014]** *A hybrid machine-crowdsourcing system for matching web tables*, J. Fan et al., ICDE 2014
- **[Feng-VLDB2011]** *CrowdDB: Query Processing with the VLDB Crowd*, A. Feng et al. , VLDB 2011
- **[Franklin-SIGMOD11]** *CrowdDB: answering queries with crowdsourcing*, Michael J. Franklin et al, SIGMOD 2011
- **[Gao-SIGMOD2013]** *An online cost sensitive decision-making method in crowdsourci ng systems*, J. Gao et al., SIGMOD 2013
- **[Liu-VLDB2012]** *CDAS: A Crowdsourcing Data Analytics System*, X. Liu et al., VLD B 2012
- **[Marcus-CIDR2011]** *Crowdsourced Databases: Query Processing with People*, A. M arcus et al., CIDR 2011
- **[Marcus-SIGMOD2011]** *Demonstration of Qurk: a query processor for human operat ors*, A. Marcus et al., SIGMOD 2011
- **[Marcus-VLDB2012]** *Human-powered Sorts and Joins*, A. Marcus et al., VLDB 2011
- **[Parameswaran-CIKM2012]** *Deco: declarative crowdsourcing*, A. G. Parameswaran et al., CIKM 2012
- **[Park-VLDB2013]** *Query Optimization over Crowdsourced Data*, H. Park et al., VLDB 2013
- **[Ooi-SIGKDD14]** *Contextual Crowd Intelligence*, B. C. Ooi et al., SIGKDD 2014

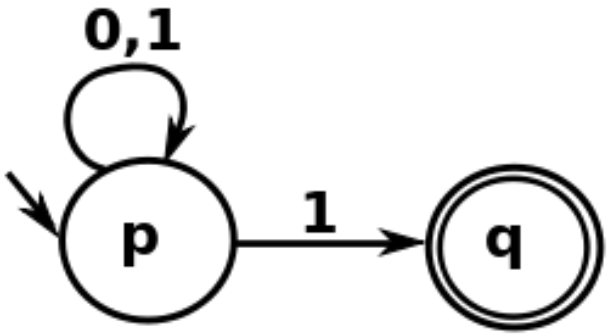
Part 2

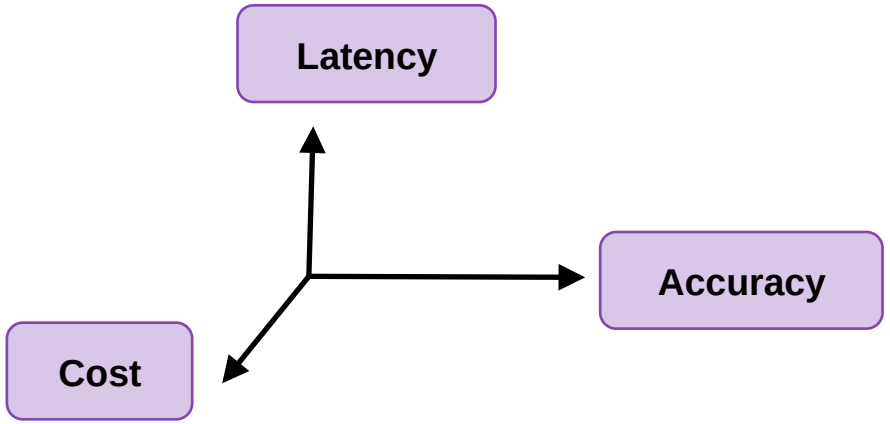
HUMAN POWERED DATABASE OPERATIONS

New Challenges

- Open-world assumption (OWA)
  - Eg, workers suggest a new relevant image
- Non-deterministic algorithmic behavior
  - Eg, different answers by the same workers
- Trade-off among cost, latency, and accuracy

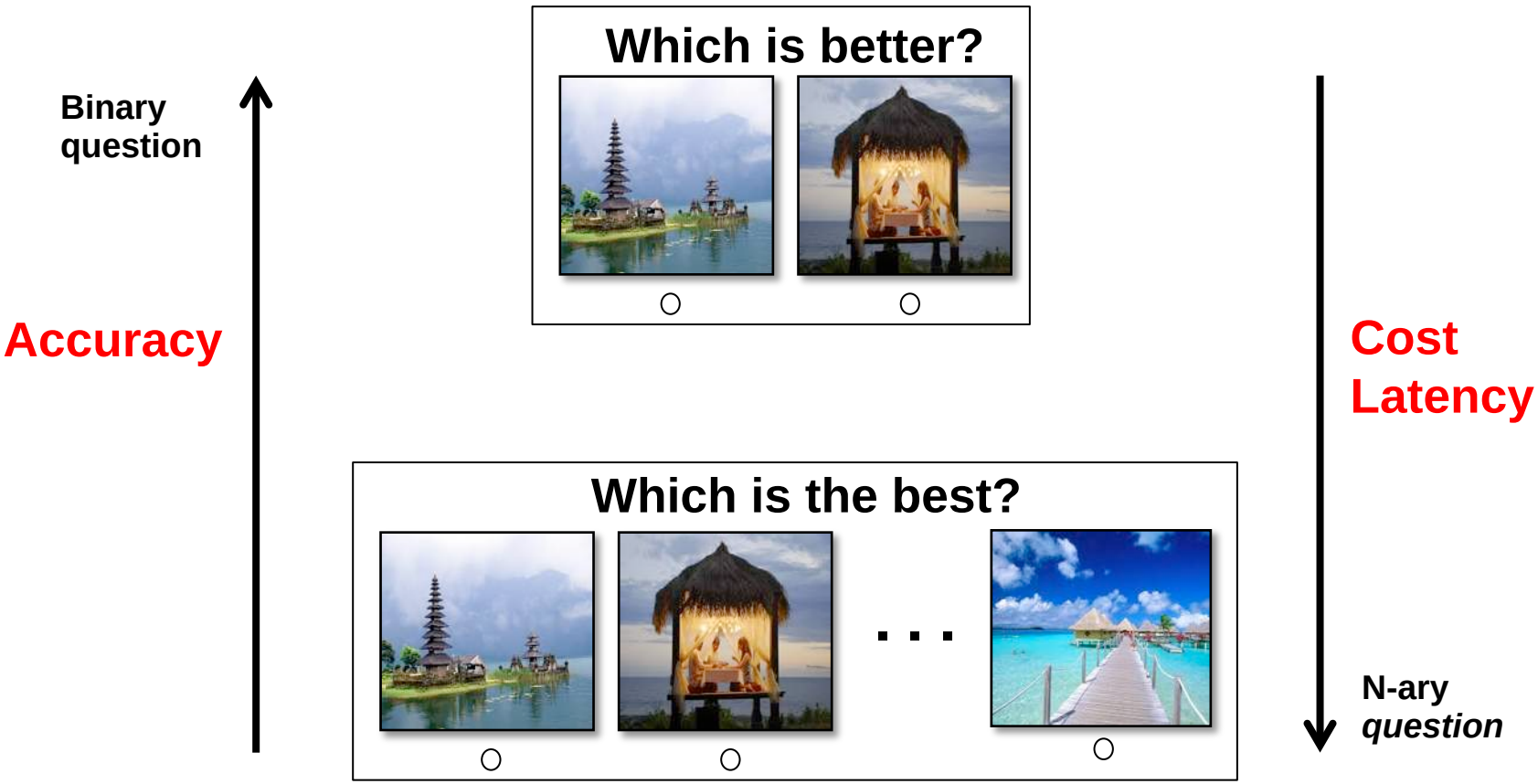






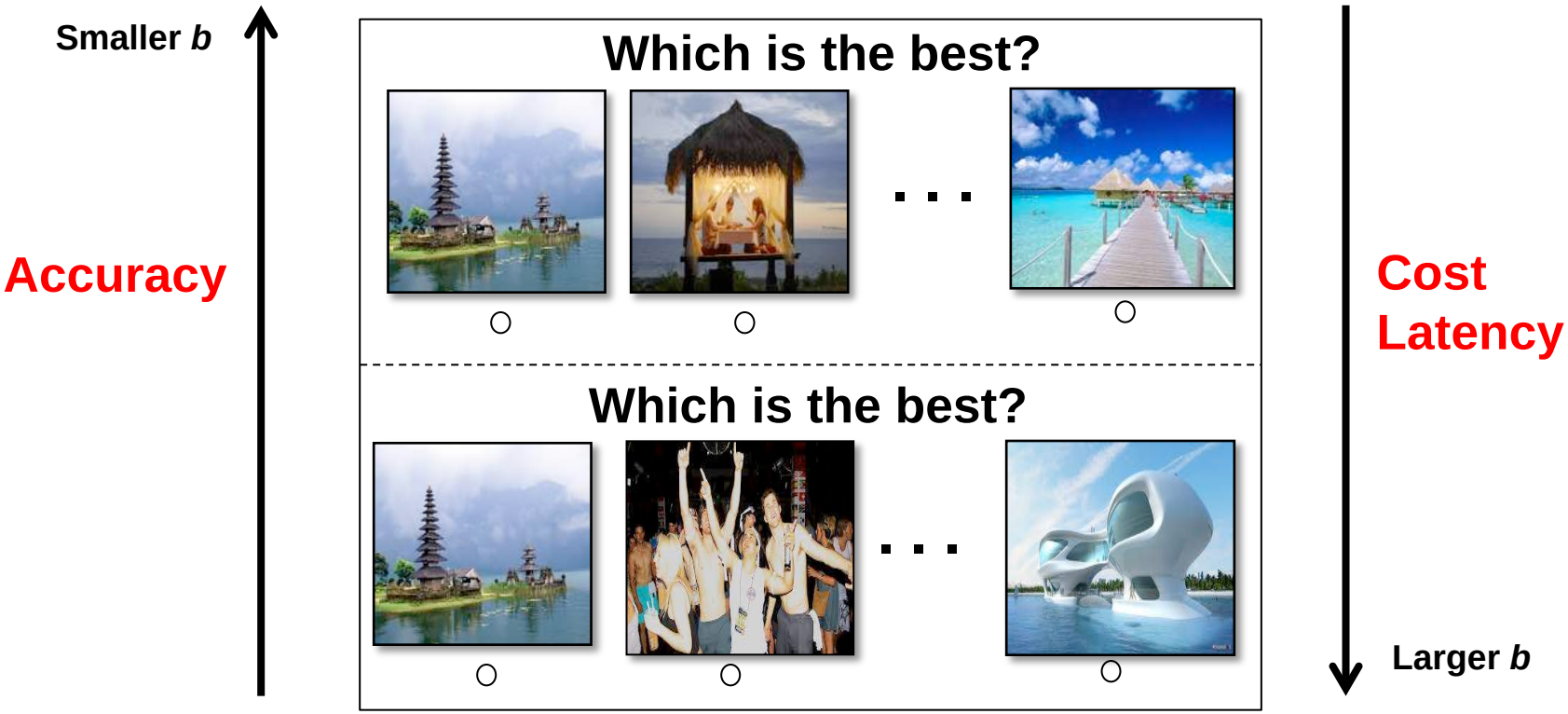
Size of Comparison

- Diverse forms of questions in a HIT
- Different sizes of comparisons in a question



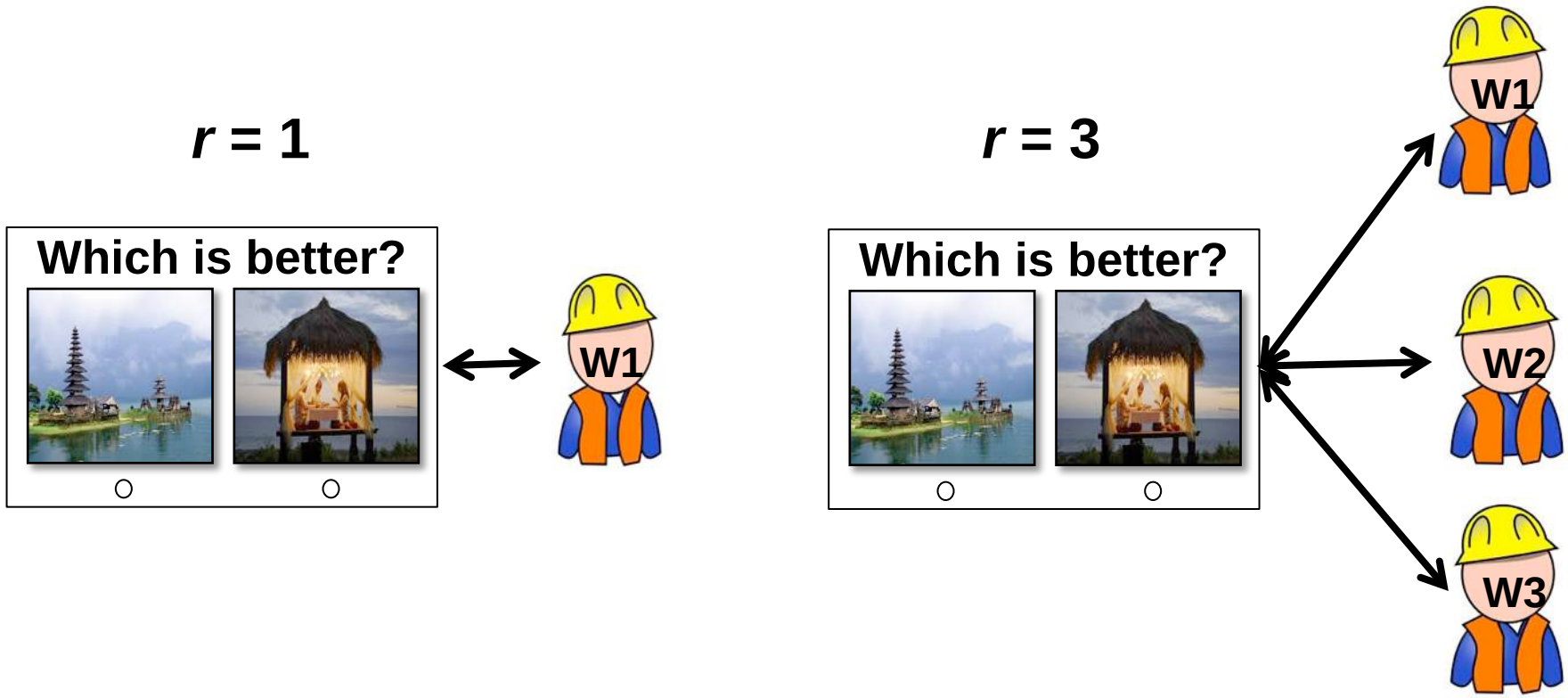
Size of Batch

- Repetitions of questions within a HIT
- Eg, two *n*-ary questions (batch factor *b*=2)



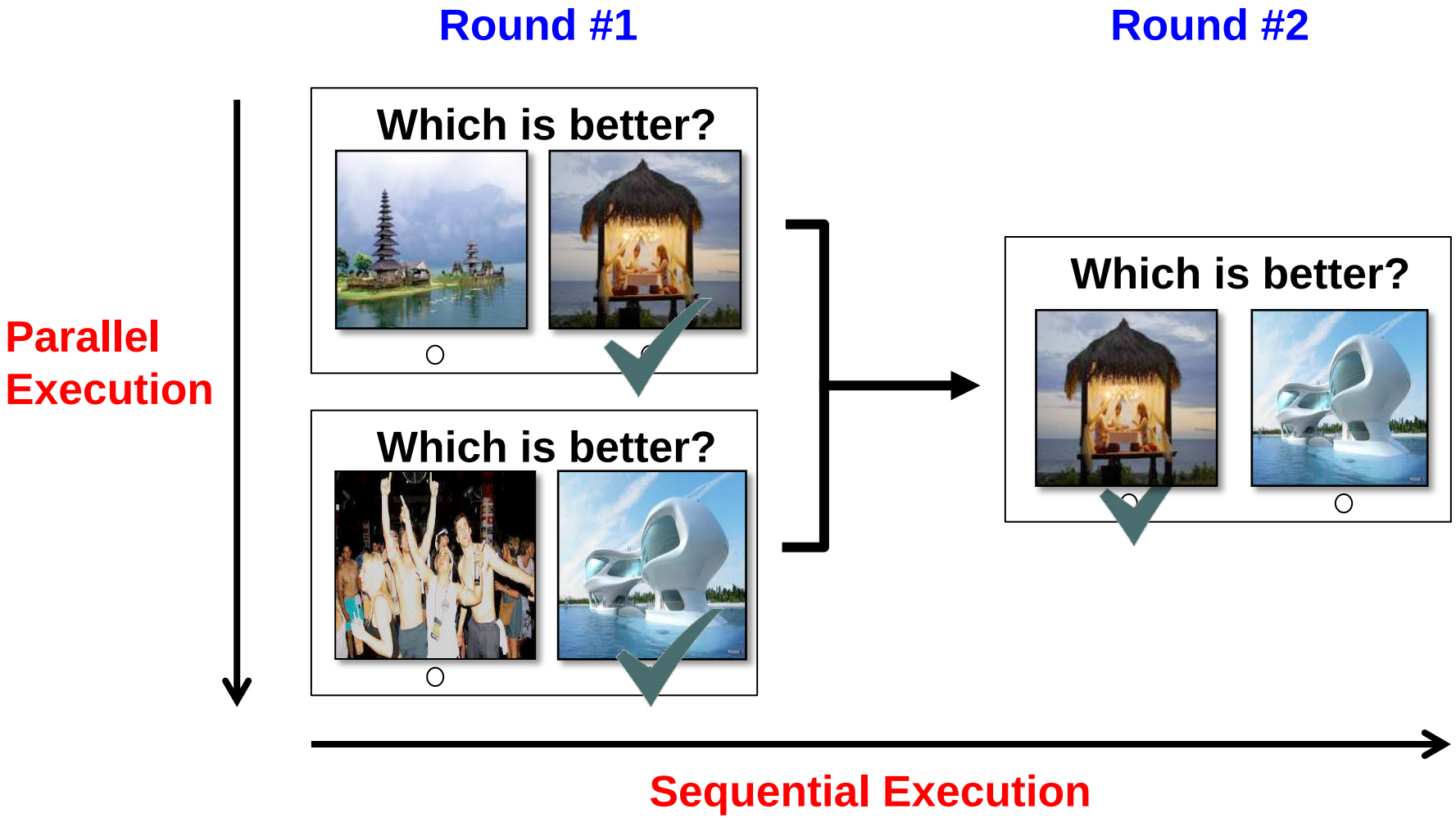
Response (*r*)

- # of human responses seeked for a HIT



Round (= Step)

- Algorithms are executed in rounds
- # of rounds ≈ latency



DB Operations

- The focus of Part 2
  - Top-1 (= Max)
  - Top-k
  - Sort
  - *Demo*
  - Select
  - Count
  - Join

Top-1 Operation

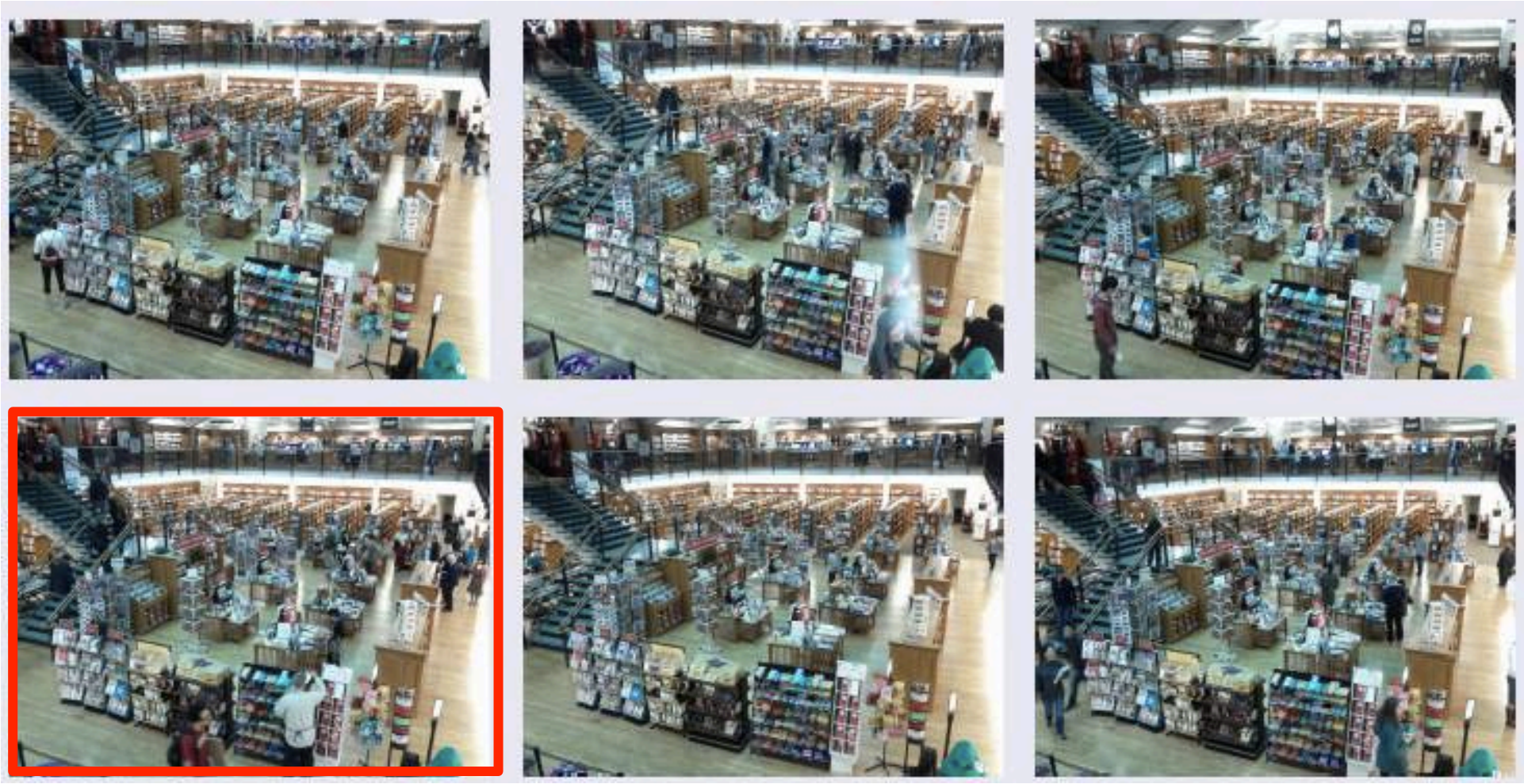
- **Find the top-1, either MAX or MIN, among *N* items w.r.t. a predicate *P***
- Often *P* is subjective, fuzzy, ambiguous, and/or difficult-for-machines-to-compute
  - Which is the most "representative" image of Shanghai?
  - Which animal is the most "dangerous"?
  - Which soccer player is the most "valuable"?
- Note
  - Avoid sorting all *N* items to find top-1

Top-1 Operation

- Examples
  - **[Venetis-WWW12]** introduces the bubble max and tournament-based max in a parameterized framework
  - **[Guo-SIGMOD12]** studies how to find max using pair-wise questions in the tournament-like setting and how to improve accuracy by asking more questions

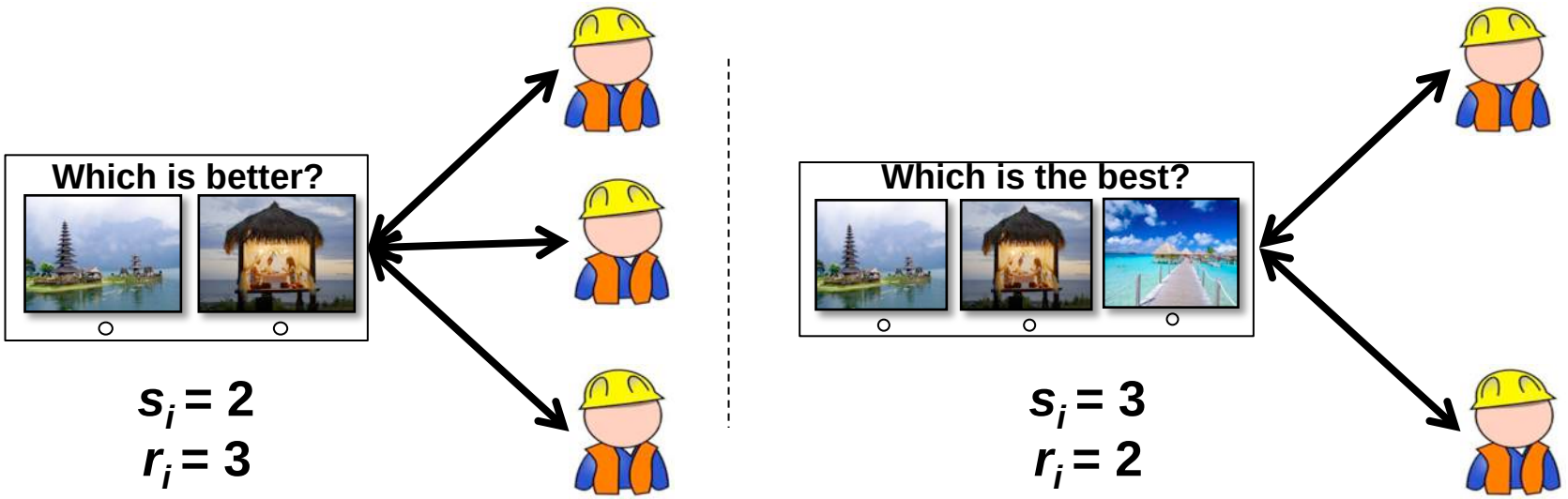
Max [Venetis-WWW12]

- Eg, Finding peak hours



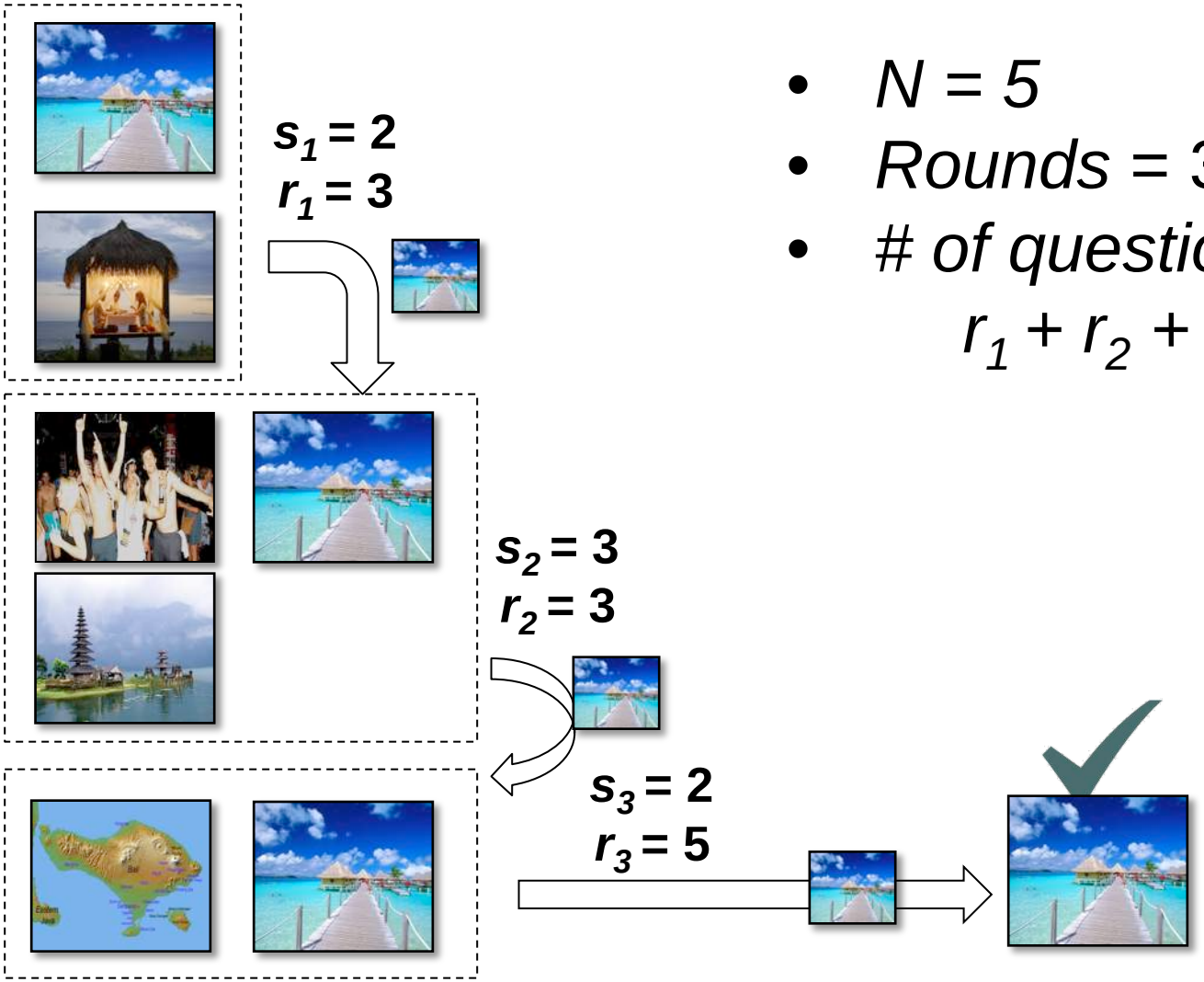
Max [Venetis-WWW12]

- Introduced two Max algorithms
  - Bubble Max
  - Tournament Max
- Parameterized framework
  - s_i: size of sets compared at the i-th round
  - r_i: # of human responses at the i-th round



Max [Venetis-WWW12]

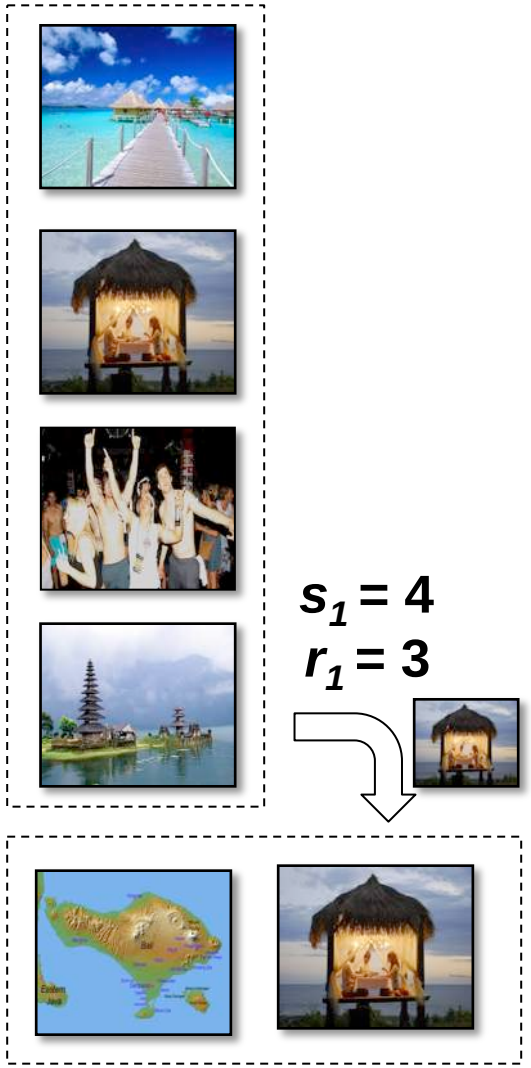
- Bubble Max Case #1



- *N = 5*
- *Rounds = 3*
- *# of questions =*
  $r_1 + r_2 + r_3 = 11$

Max [Venetis-WWW12]

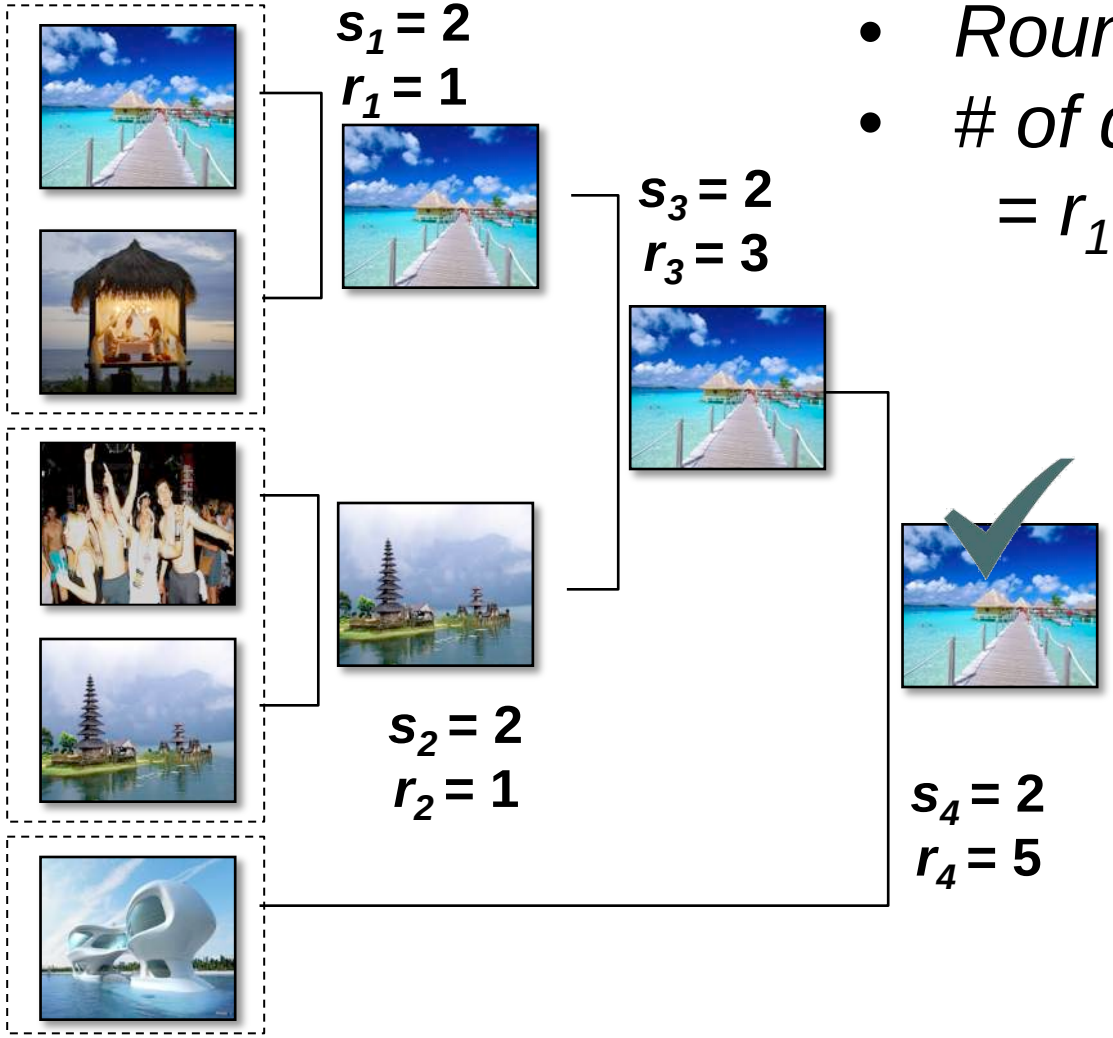
- Bubble Max Case #2



- *N = 5*
- *Rounds = 2*
- *# of questions =*
  $r_1 + r_2 = 8$

Max [Venetis-WWW12]

- Tournament Max

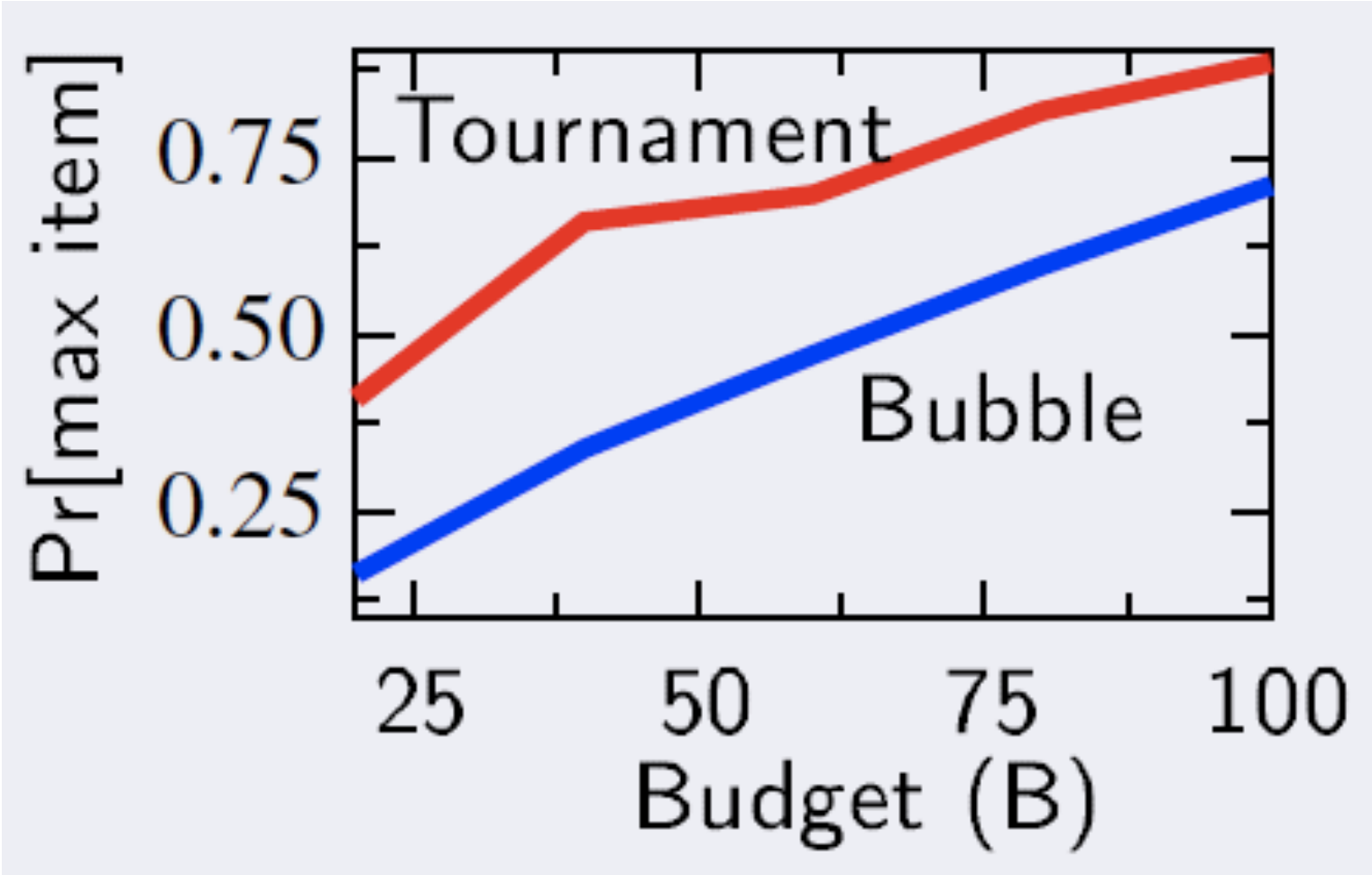


- *N = 5*
- *Rounds = 3*
- *# of questions*
  $= r_1 + r_2 + r_3 + r_4 = 10$

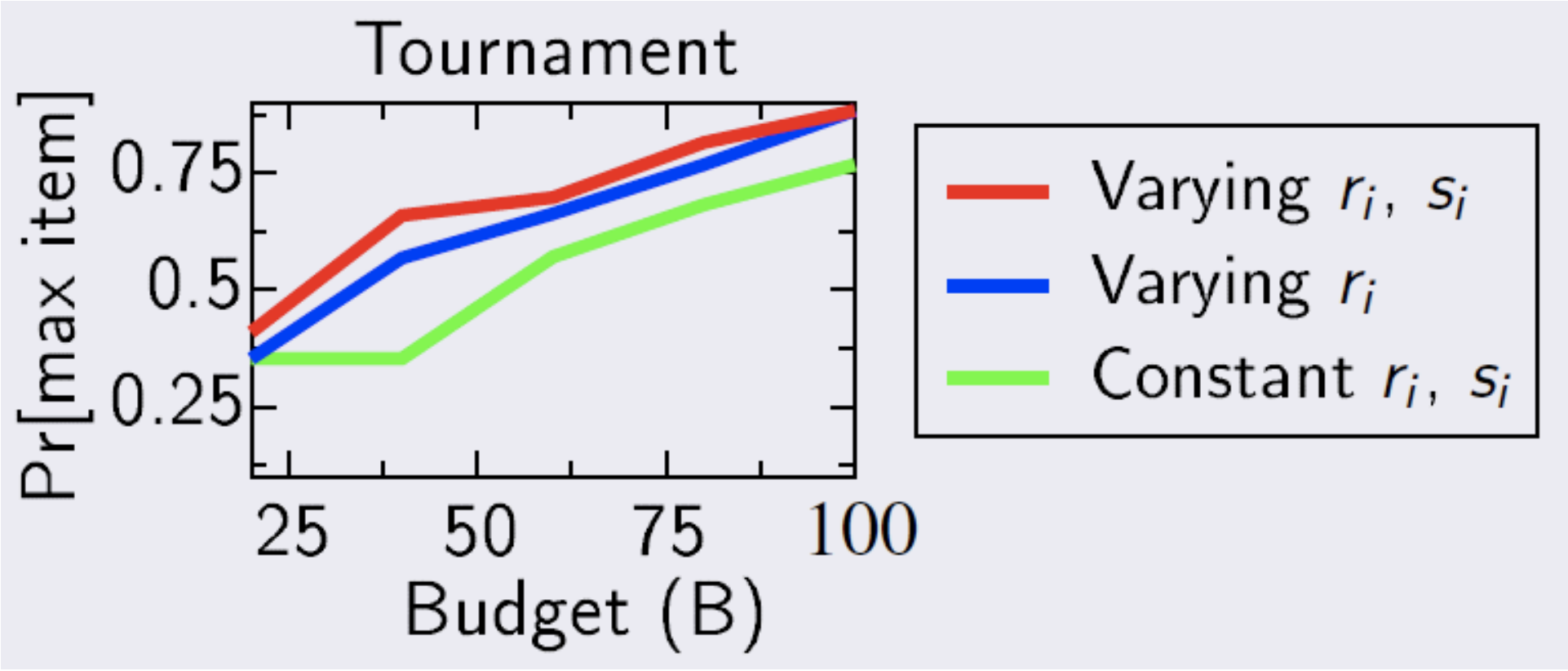
Max [Venetis-WWW12]

- How to find optimal parameters?: s_i and r_i
- Tuning Strategies (using Hill Climbing)
  - Constant s_i and r_i
  - Constant s_i and varying r_i
  - Varying s_i and r_i

Max [Venetis-WWW12]



Max [Venetis-WWW12]



Top-*k* Operation

- **Find top-*k* items among *N* items w.r.t. a predicate *P***
- Top-*k* list vs. top-*k* set
- Objective
  - Avoid sorting all *N* items to find top-*k*

Top-*k* Operation

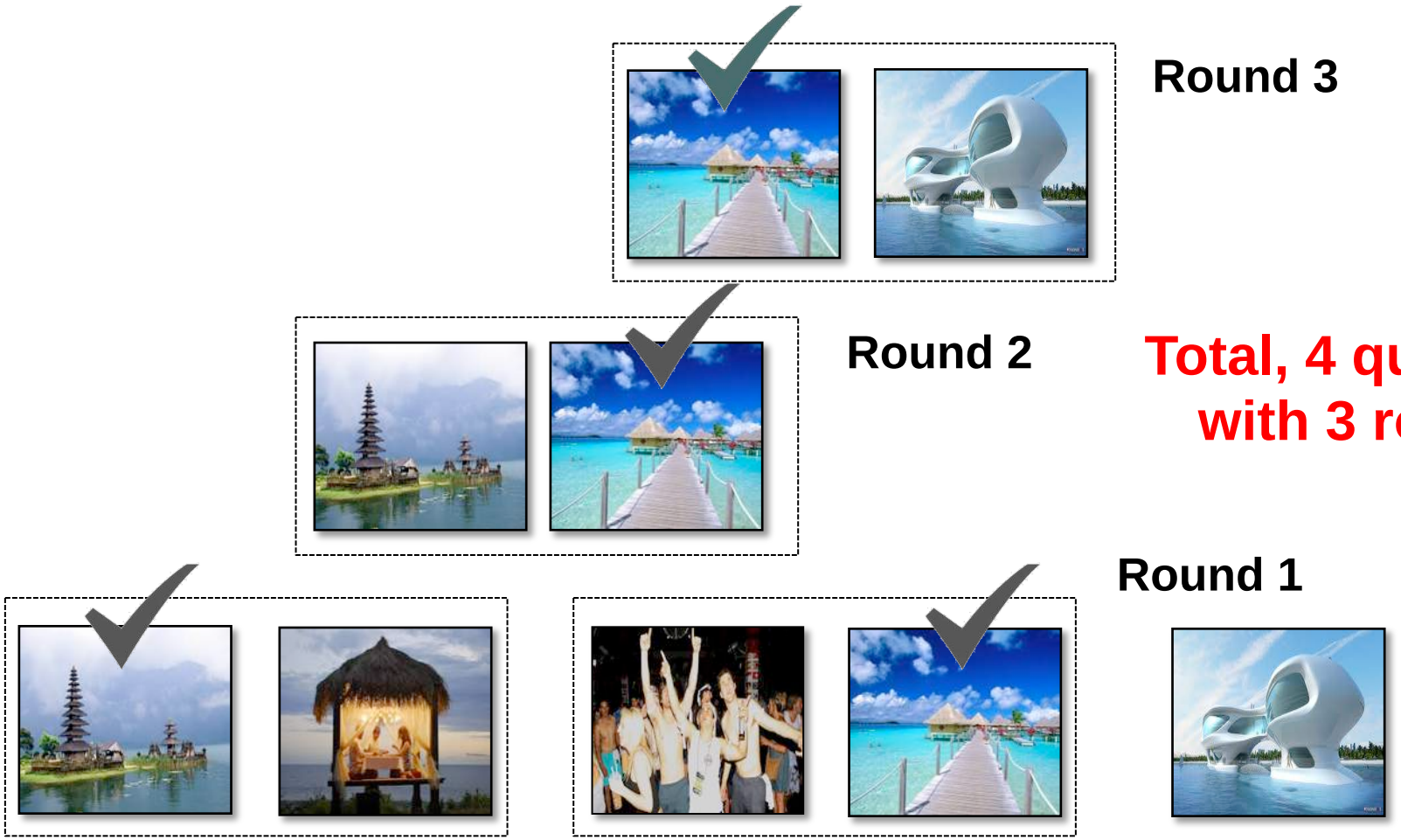
- Examples
  - **[Davidson-ICDT13]** investigates the variable user error model in solving top-*k* list problem
  - **[Polychronopoulous-WebDB13]** proposes tournament-based top-*k* set solution

Top-*k* Operation

- Naïve solution is to "sort" *N* items and pick top-*k* items
- Eg, *N*=5, *k*=2, "Find two best Bali images?"
  - Ask $\binom{5}{2}$ = 10 pair-wise questions to get a total order
  - Pick top-2 images

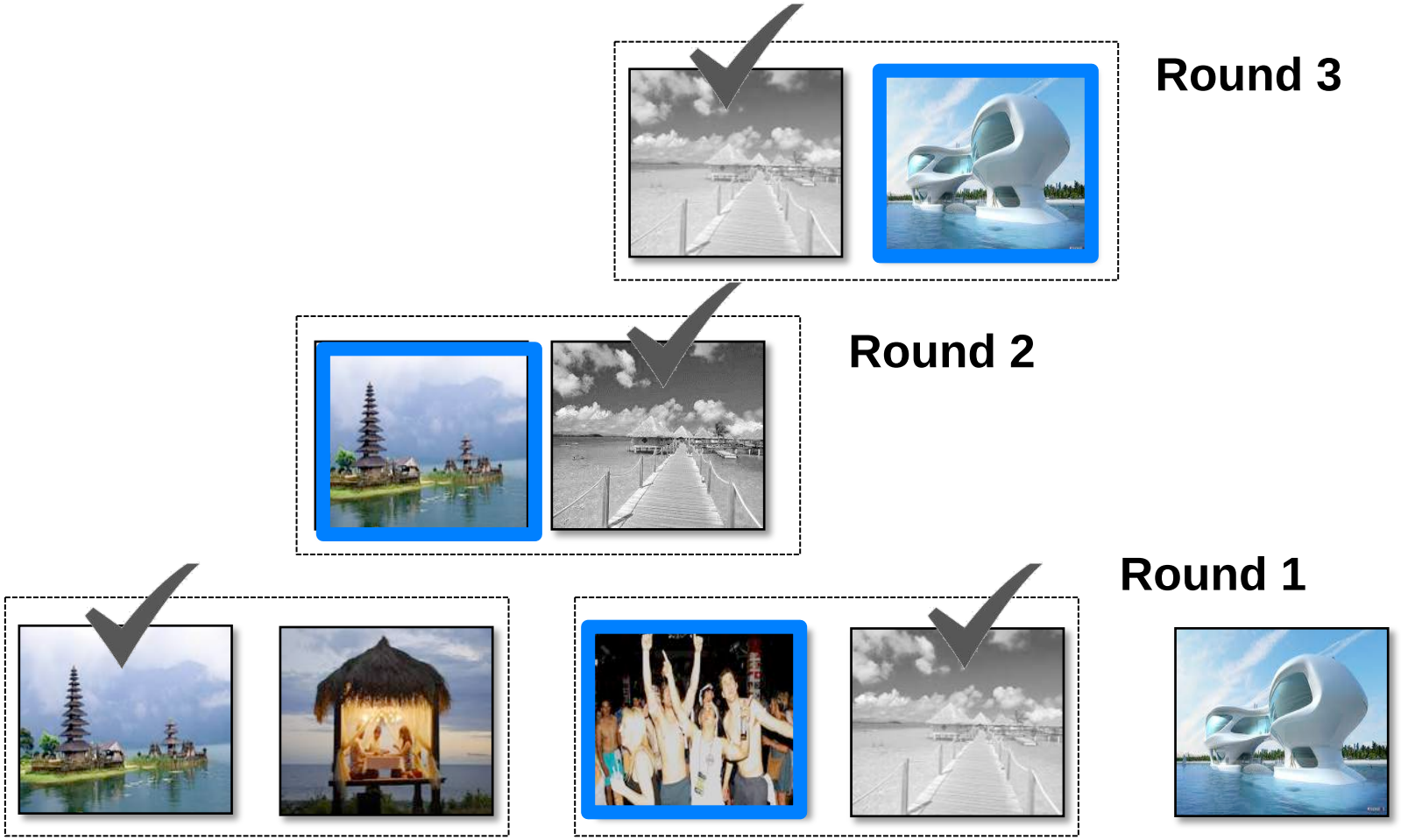
Top-*k*: Tournament Solution (*k* = 2)

- Phase 1: **Building a tournament tree**
  - For each comparison, only winners are promoted to the next round



Top-*k*: Tournament Solution (*k* = 2)

- Phase 2: **Updating a tournament tree**
  - **Iteratively** asking pair-wise questions from the bottom level



Top-*k*: Tournament Solution (*k* = 2)

- Phase 2: **Updating a tournament tree**
  - **Iteratively** asking pair-wise questions from the bottom level

Round 5

Round 4

Total, 6 questions
With 5 rounds

Top-*k*: Tournament Solution

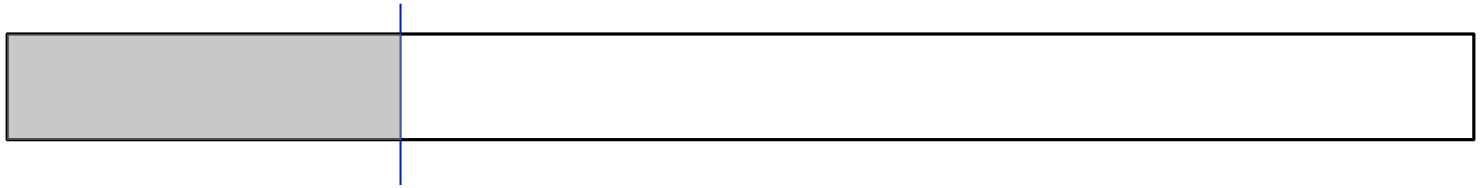
- This is a top-*k* **list** algorithm
- Analysis

	k = 1	k ≥ 2
# of questions	O(n)	$O(n + k \lceil \log_2 n \rceil)$
# of rounds	$O(\lceil \log_2 n \rceil)$	$O(k \lceil \log_2 n \rceil)$

- If there is no constraint for the number of rounds, this tournament sort based top-*k* scheme yields the optimal result

Top-*k* [Polychronopoulous-WebDB13]

- Top-*k* **set** algorithm
  - Top-k items are “better” than remaining items
  - Capture NO ranking among top-*k* items



  - Tournament-based approach
- Can become a Top-*k* **list** algorithm
  - Eg, Top-*k* **set** algorithm, followed by [Marcus-VLDB11] to sort *k* items

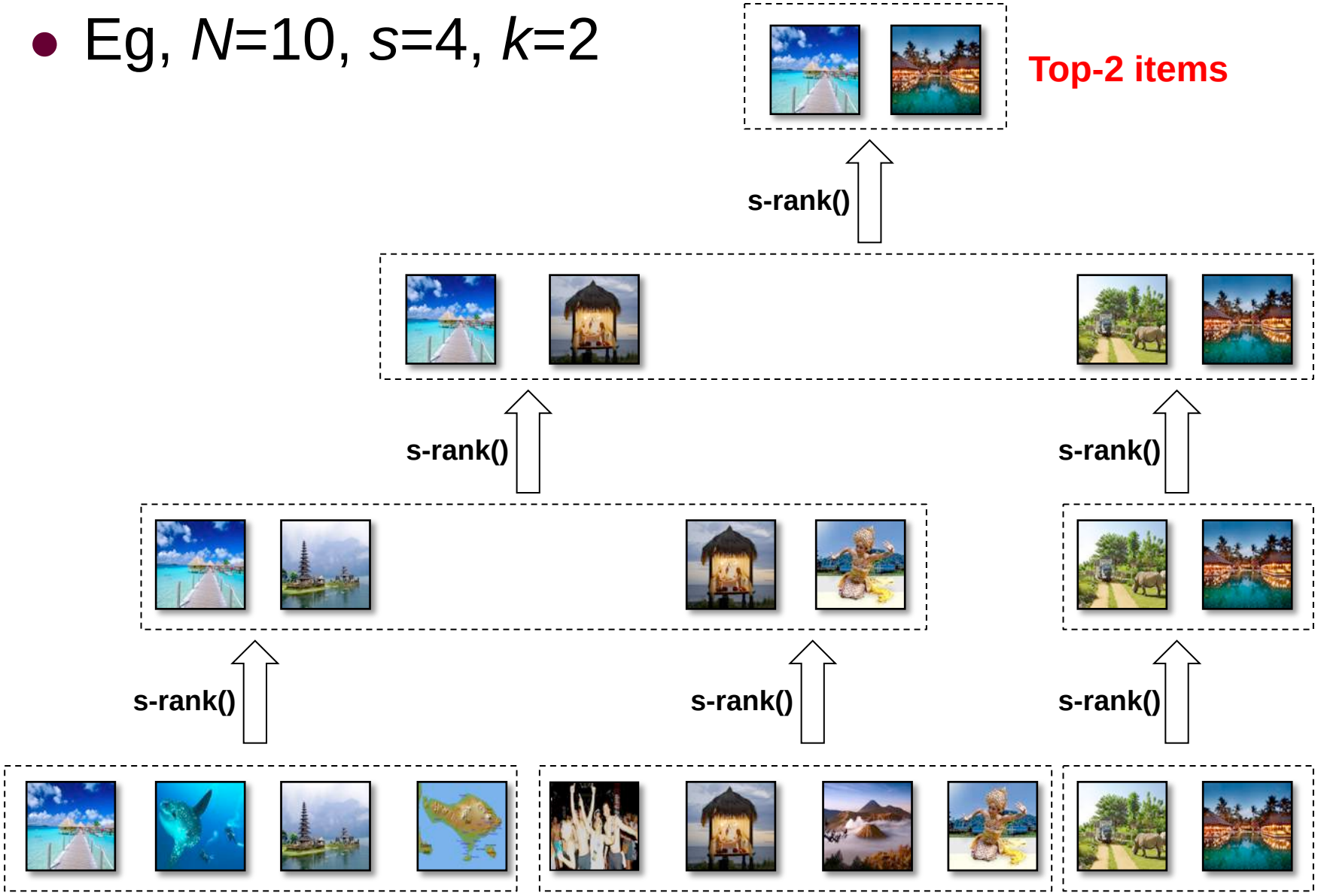
Top-*k* [Polychronopoulous-WebDB13]

- Algorithm
  - Input: *N* items, integer *k* and *s* (ie, *s* > *k*)
  - Output: top-*k* set
  - Procedure:
    - *O* ← *N* items
    - While |*O*| > *k*
      - Partition *O* into disjoint subsets of size *s*
      - Identify top-*k* items in each subset of size *s*: *s-rank(s)*
      - Merge all top-*k* items into *O*
    - Return *O*

- More effective when *s* and *k* are small
  - Eg, *s-rank*(20) with *k*=10 may give poor accuracy

Top-*k* [Polychronopoulous-WebDB13]

- Eg, *N*=10, *s*=4, *k*=2



Top-*k* [Polychronopoulous-WebDB13]

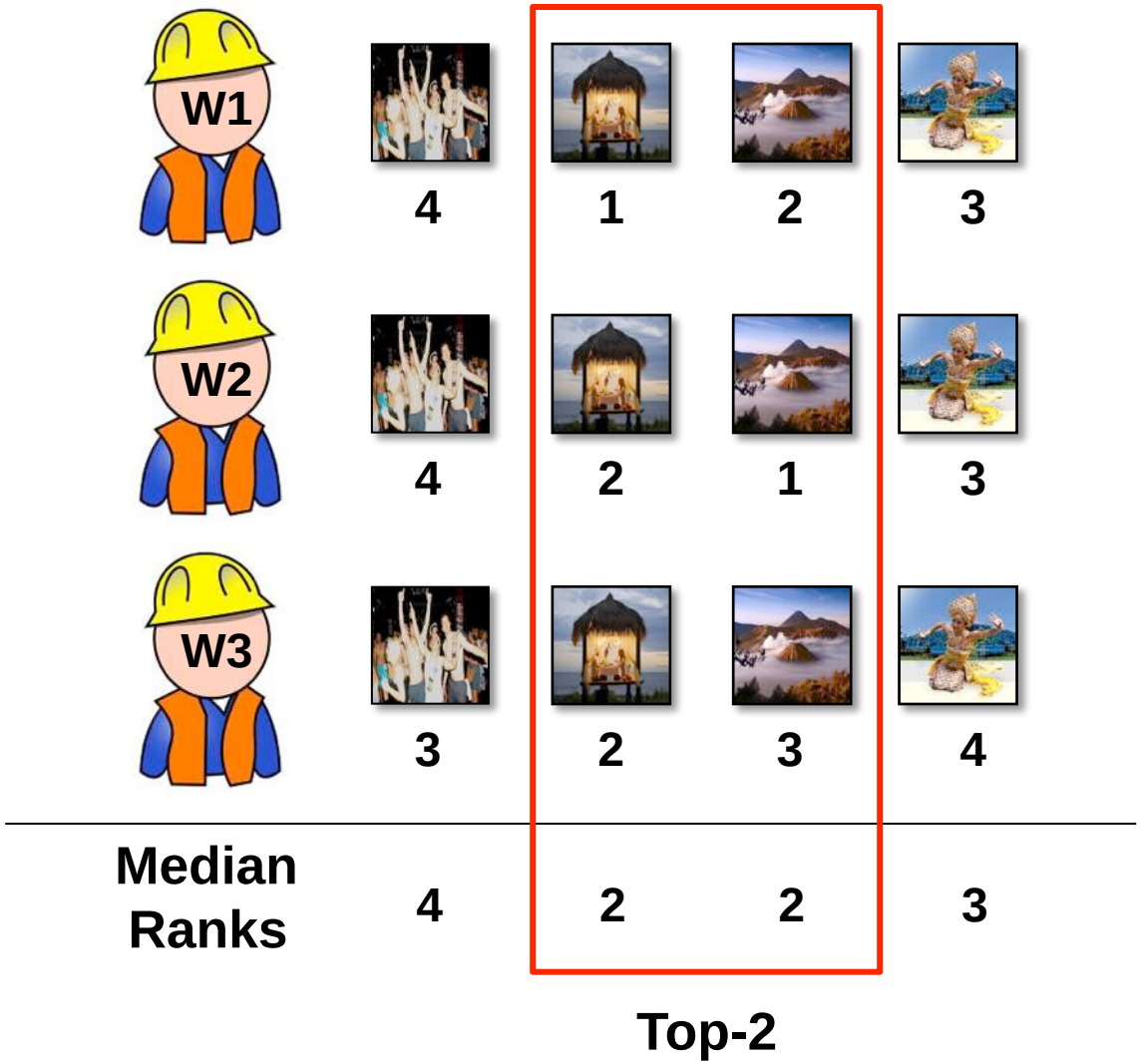
- s-rank(s)

  // workers rank *s* items and aggregate

  - Input: *s* items, integer *k* (ie, *s* > *k*), *w* workers
  - Output: top-*k* items among *s* items
  - Procedure:
    - For each of *w* workers
      - Rank *s* items ≈ comparison-based sort [Marcus-VLDB11]
    - Merge *w* rankings of *s* items into a single ranking
      - Use median-rank aggregation [Dwork-WWW01]
    - Return top-*k* item from the merged ranking of *s* items

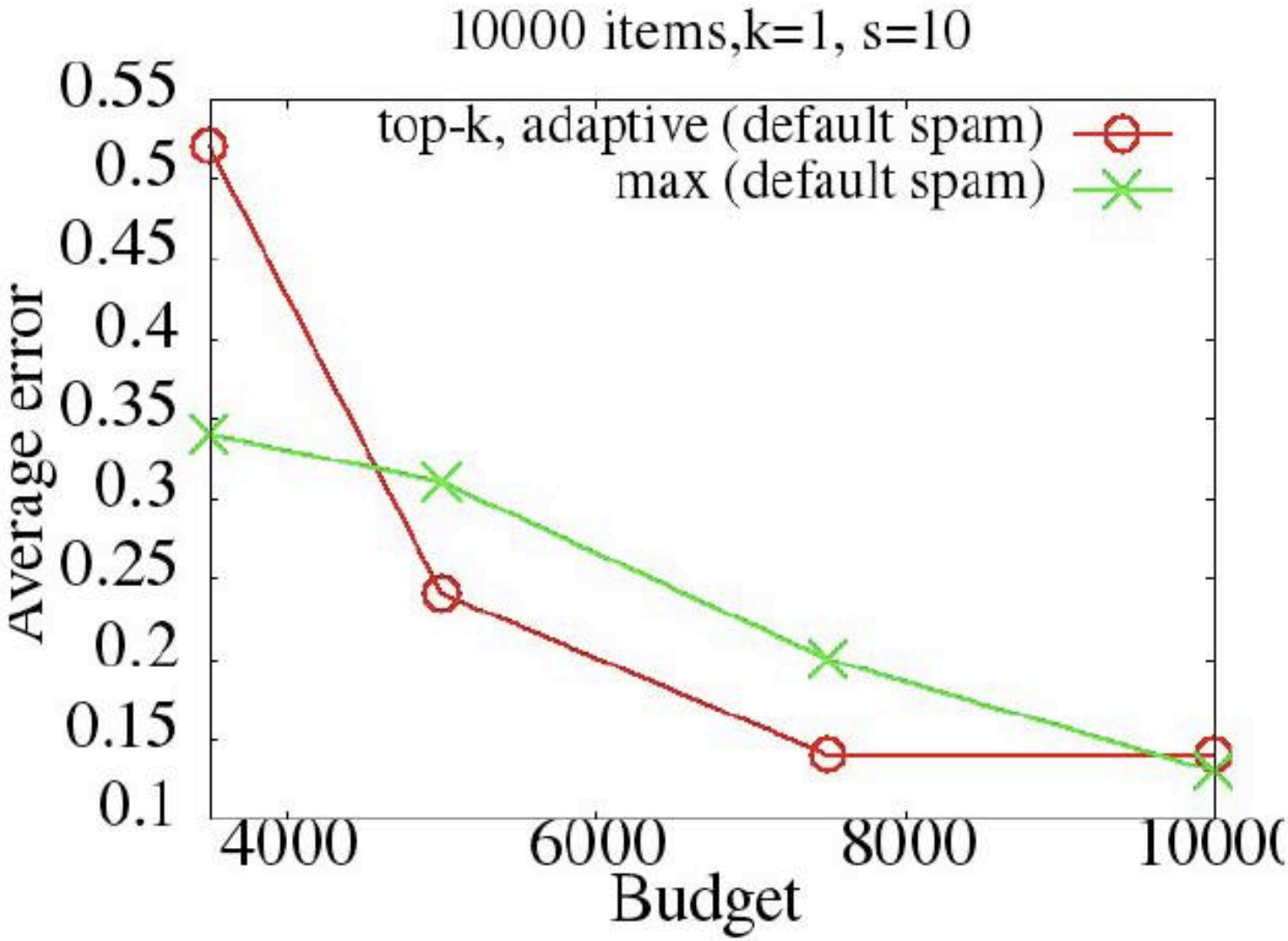
Top-*k* [Polychronopoulous-WebDB13]

- Eg, s-rank(): *s*=4, *k*=2, *w*=3



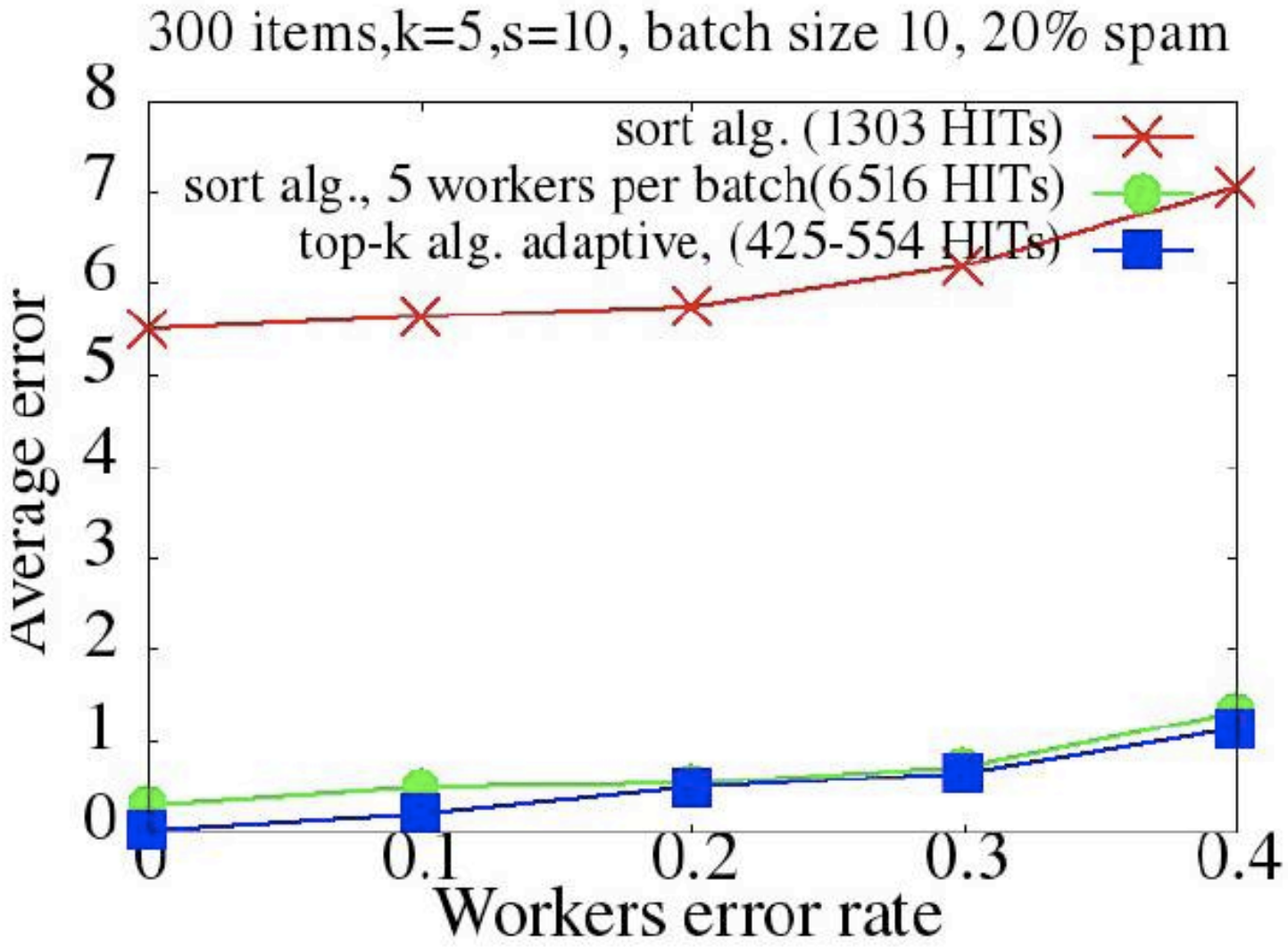
Top-*k* [Polychronopoulous-WebDB13]

- Comparison to Max [Venetis-WWW12]



Top-*k* [Polychronopoulous-WebDB13]

- Comparison to Sort [Marcus-VLDB11]



Sort Operation

- **Rank *N* items w.r.t. a predicate *P***

```
SELECT     *
FROM       SoccerPlayers AS P
WHERE      P.WorldCupYear = ‘2014’
ORDER BY   CrowdOp(‘most-valuable’)
```









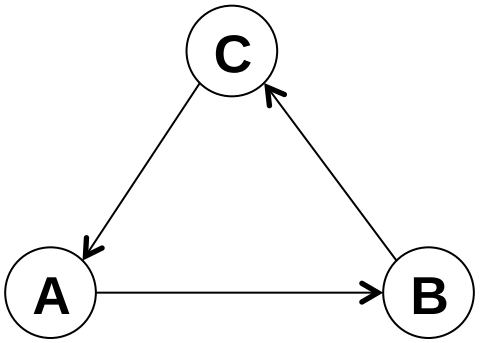
. . .

Naïve Sort

- Eg, “Which of two players is better?”
- Naïve all pair-wise comparisons takes $\binom{N}{2}$ comparisons
  - Optimal # of comparison is *O(N log N)*

Naïve Sort

- Conflicting opinions may occur
  - Cycle: A > B, B > C, and C > A



- If no cycle occurs
  - Naïve all pair-wise comparisons takes $\binom{N}{2}$ comparisons
- If cycle exists
  - More comparisons from workers
  - Break cycle

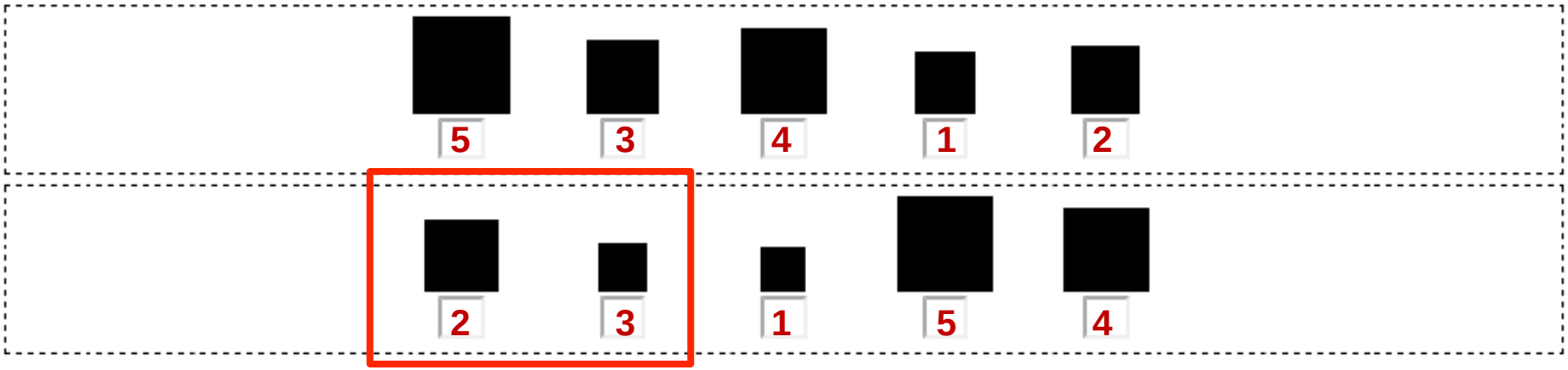
Sort [Marcus-VLDB11]

- Proposed 3 crowdsourced sort algorithms
- #1: Comparison-based Sort
  - Workers rank *S* items ($S \subset N$) per HIT
  - Each HIT yields $\binom{S}{2}$ pair-wise comparisons
  - Build a directed graph using all pair-wise comparisons from all workers
    - If *i* > *j*, then add an edge from *i* to *j*
  - Break a cycle in the graph: “head-to-head”
    - Eg, If *i* > *j* occurs 3 times and *i* < *j* occurs 2 times, keep only *i* > *j*
  - Perform a topological sort in the DAG

Sort [Marcus-VLDB11]

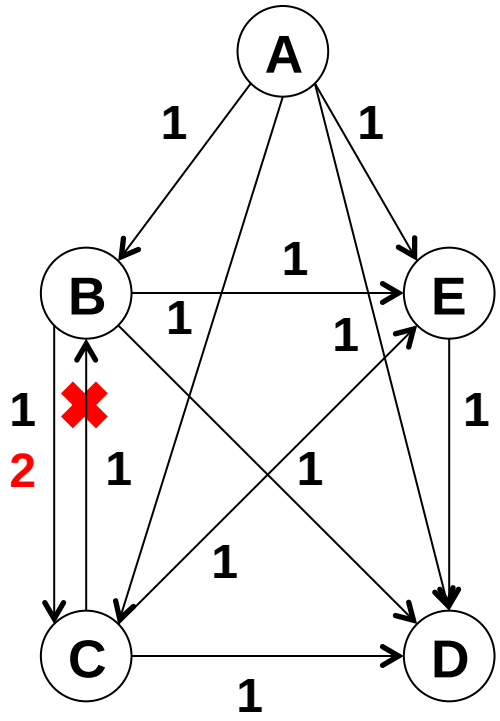
There are 2 groups of squares. We want to order the squares in each group from smallest to largest.

- Each group is surrounded by a dotted line. Only compare the squares within a group.
- Within each group, assign a number from 1 to 7 to each square, so that:
  - 1 represents the smallest square, and 7 represents the largest.
  - We do not care about the specific value of each square, only the relative order of the squares.
  - Some groups may have less than 7 squares. That is OK: use less than 7 numbers, and make sure they are ordered according to size.
  - If two squares in a group are the same size, you should assign them the same number.



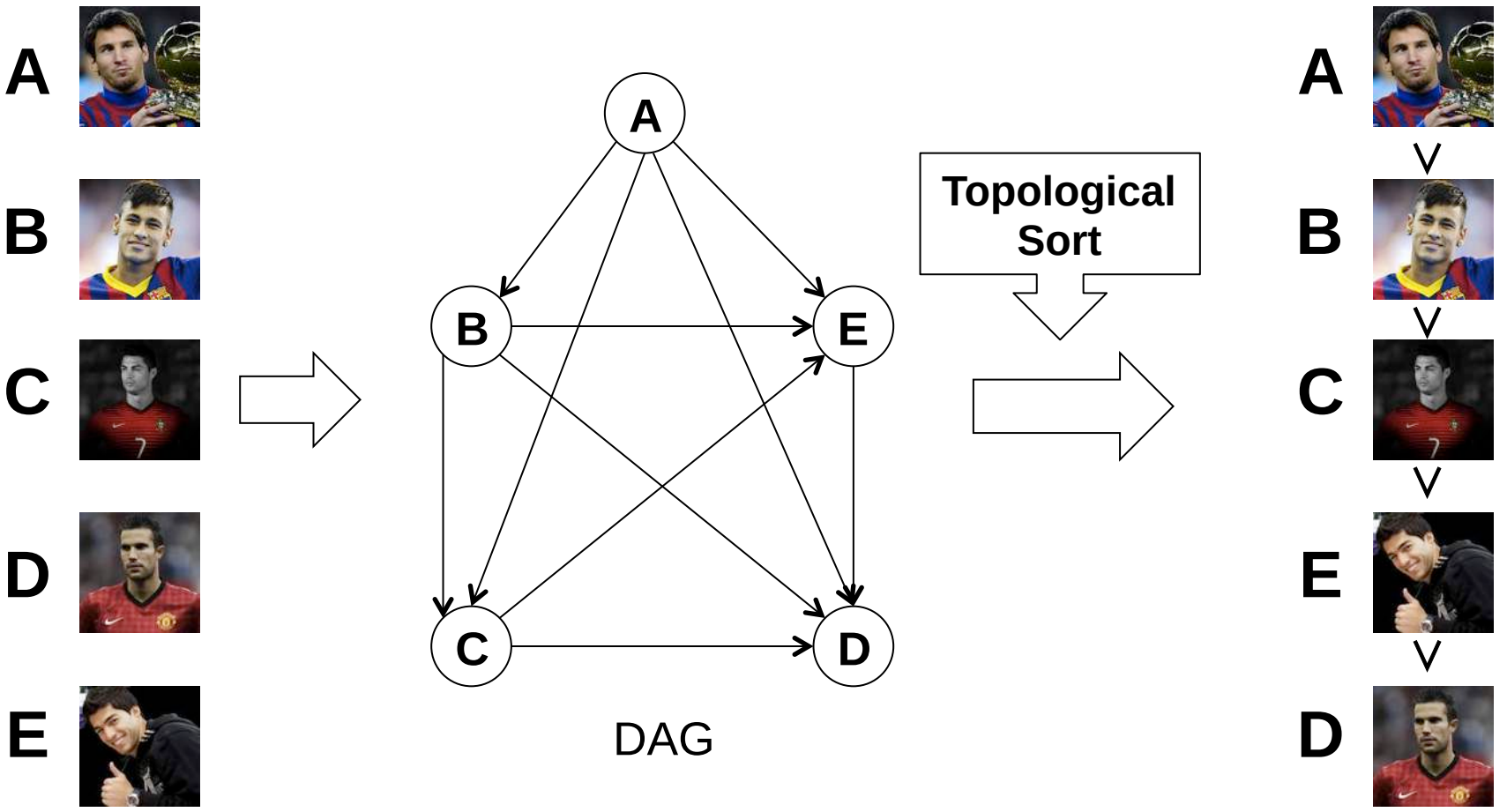
Sort [Marcus-VLDB11]

- N=5, S=3



Sort [Marcus-VLDB11]

- N=5, S=3



Sort [Marcus-VLDB11]

- #2: Rating-based Sort
  - *W* workers rate each item along a numerical scale
  - Compute the mean of *W* ratings of each item
  - Sort all items using their means
  - Requires *W*N* HITs: *O(N)*

Sort [Marcus-VLDB11]

There are 2 squares below. We want to rate squares by their size.

- For each square, assign it a number from 1 (smallest) to 7 (largest) indicating its size.
- For perspective, here is a small number of other randomly picked squares:

smallest ○ ○ ✓ ○ ○ ○ ○ largest

1 2 3 4 5 6 7

smallest ○ ○ ○ ○ ✓ ○ ○ largest

1 2 3 4 5 6 7

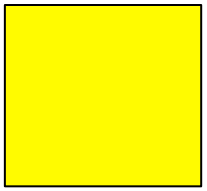
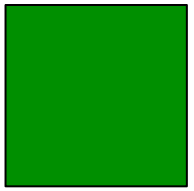
Submit

Sort [Marcus-VLDB11]

- #3: Hybrid Sort
  - First, do rating-based sort → sorted list *L*
  - Second, do comparison-based sort on *S* ($S \subset L$)
    - *S* may not be accurately sorted
  - How to select the size of S
    - Random
    - Confidence-based
    - Sliding window

Sort [Marcus-VLDB11]

- Q1: squares by size
- Q2: adult size
- Q3: dangerousness
- Q4: how much animal belongs to Saturn
  - Non-sensical question
- Q5: random response


vs.



vs.



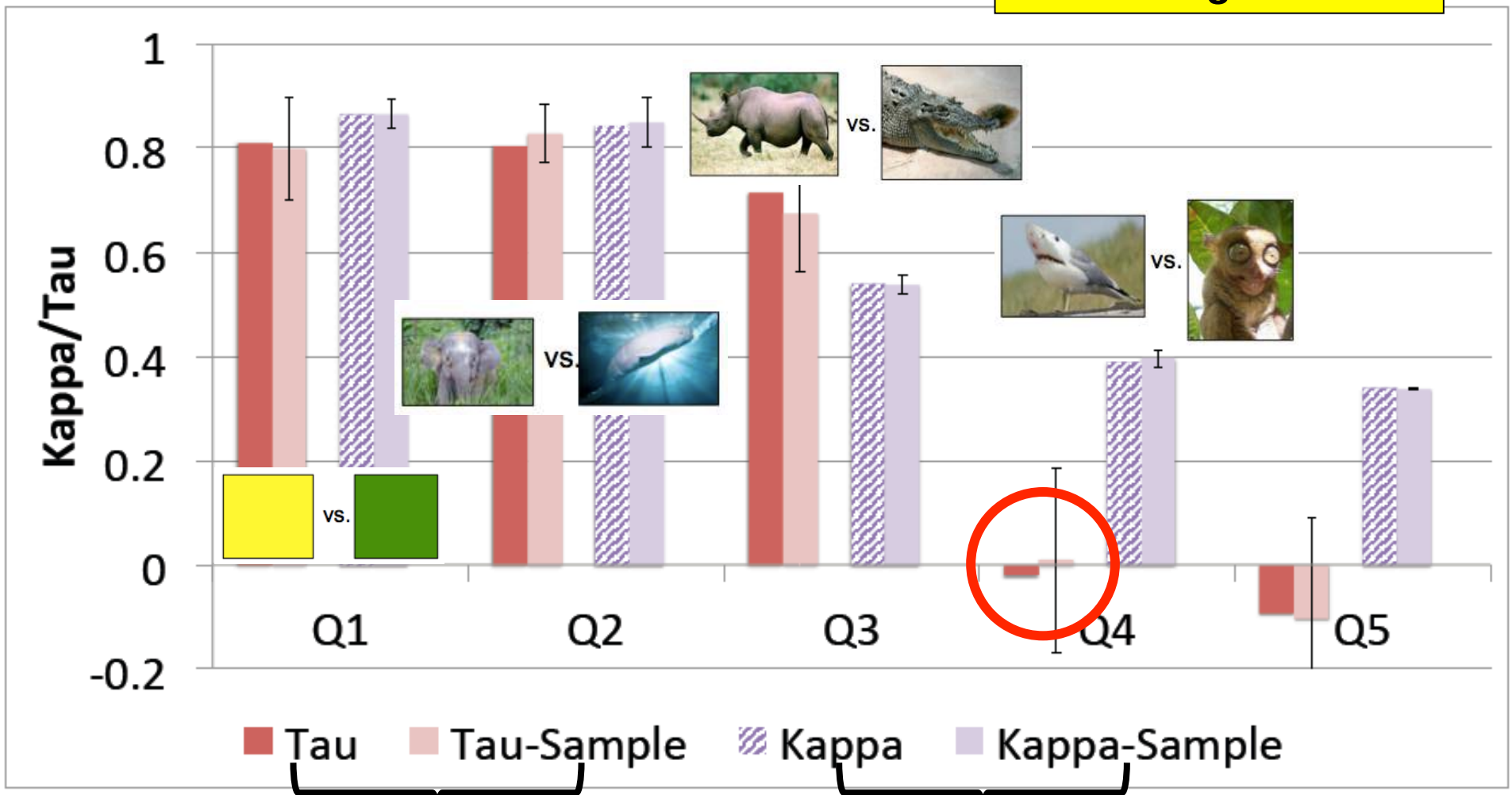
vs.



vs.


Sort [Marcus-VLDB11]

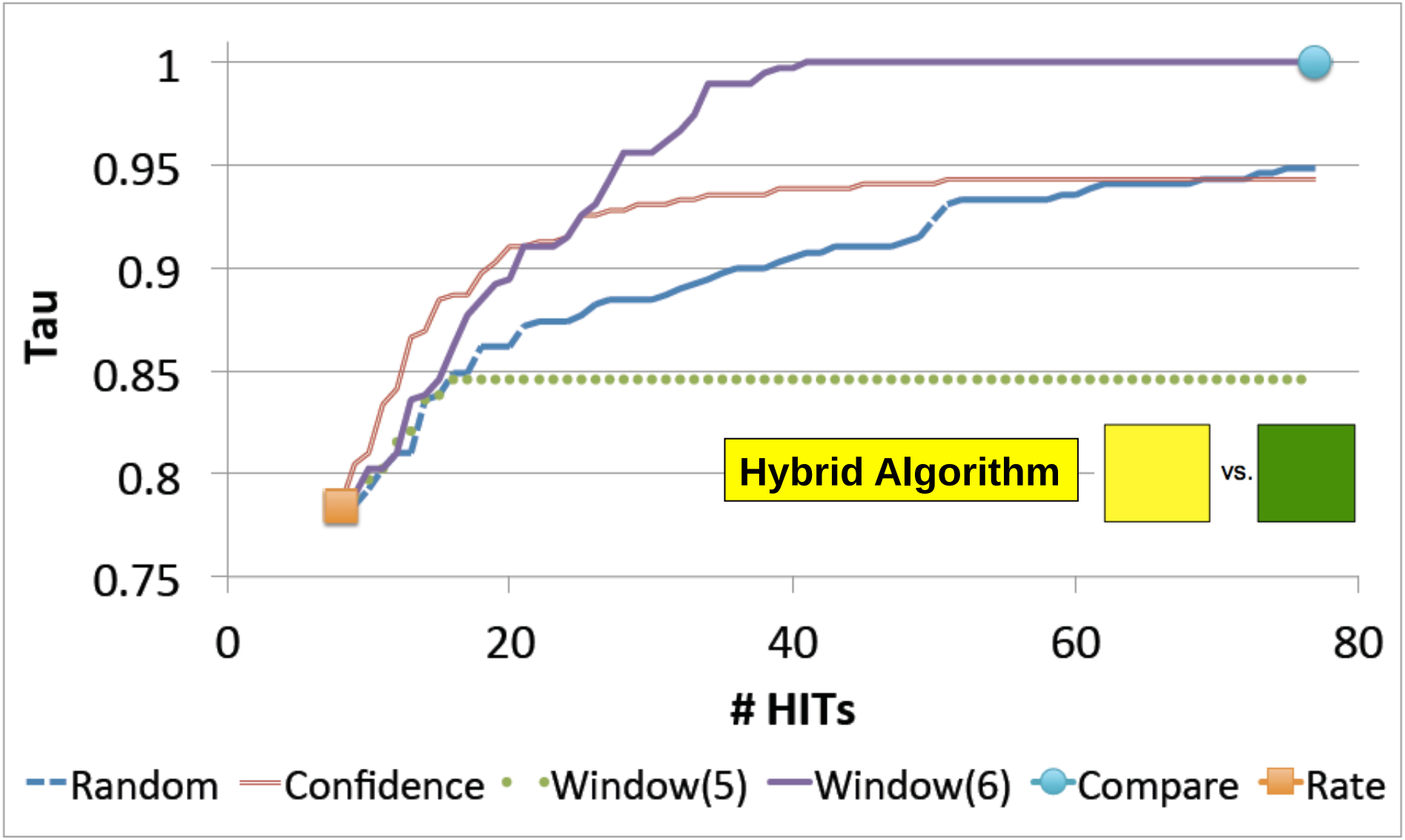
Finds that in general comparison-sort > rating-sort



Rank correlation btw. Comparison vs. rating

Worker agreement

Sort [Marcus-VLDB11]



Demo: Human-Powered Sorting

- From your smartphone or laptop, access the following URL or QR code:

http://is.gd/Eju2nU

Select Operation

- **Given *N* items, select *m* items that satisfy a predicate *P***

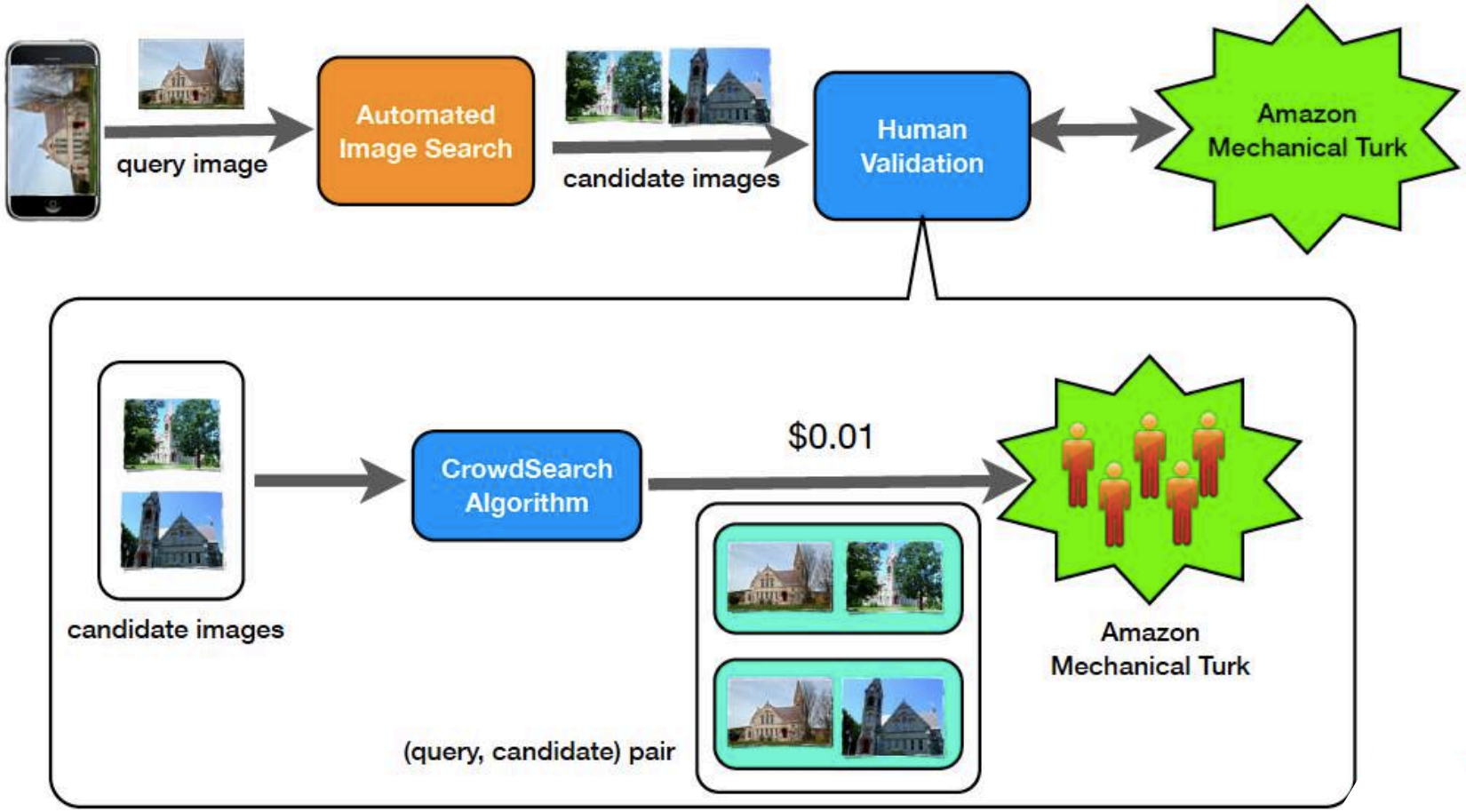
- ≈ Filter, Find, Screen, Search

Select Operation

- Examples
  - **[Yan-MobiSys10]** uses crowds to search an image relevant to a query
  - **[Parameswaran-SIGMOD12]** develops human-powered filtering algorithms
  - **[Franklin-ICDE13]** efficiently enumerates items satisfying conditions via crowdsourcing
  - **[Sarma-ICDE14]** finds a bounded number of items satisfying predicates using the optimal solution by the skyline of cost and time

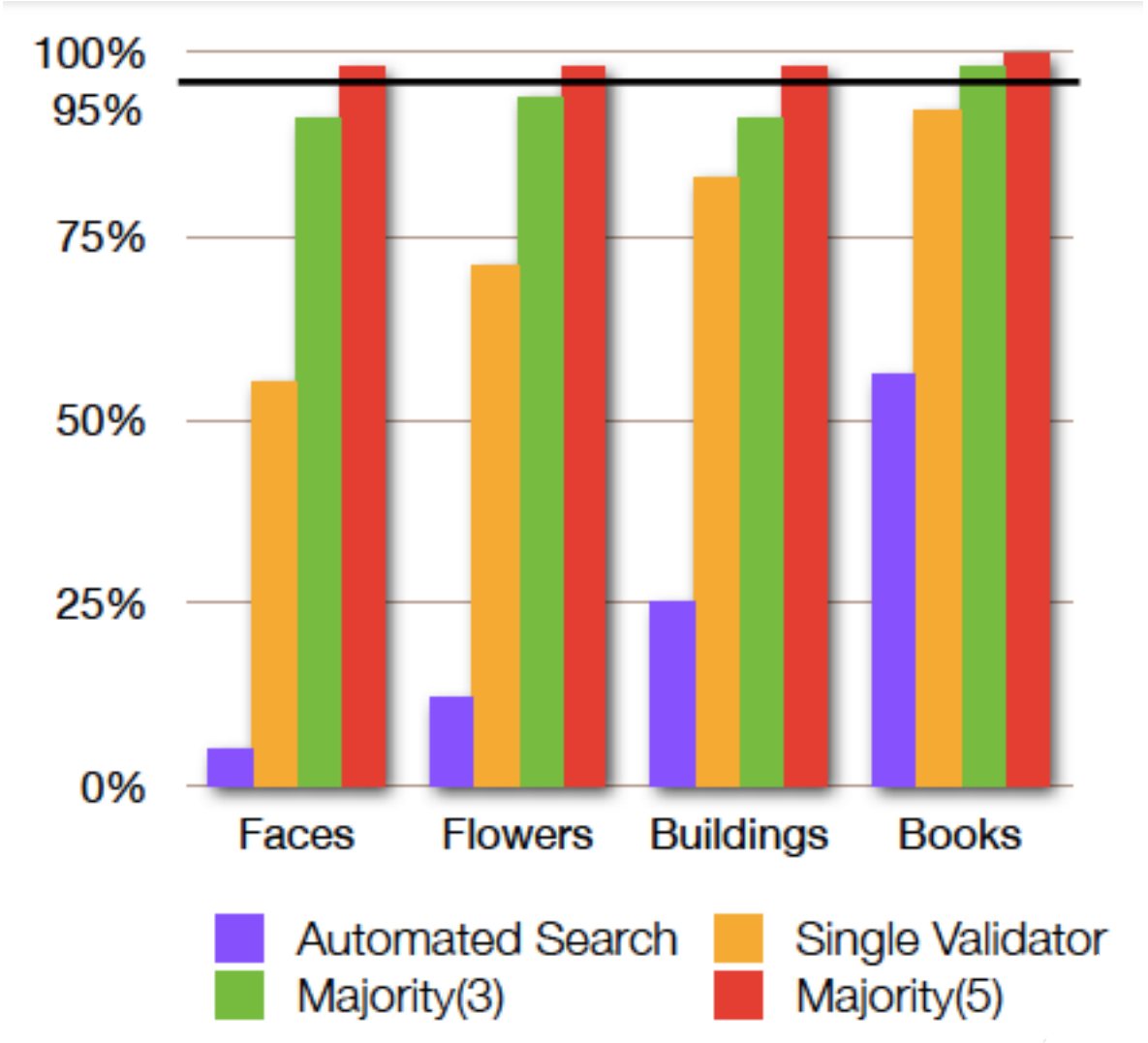
Select [Yan-MobiSys10]

- Improving mobile image search using crowdsourcing



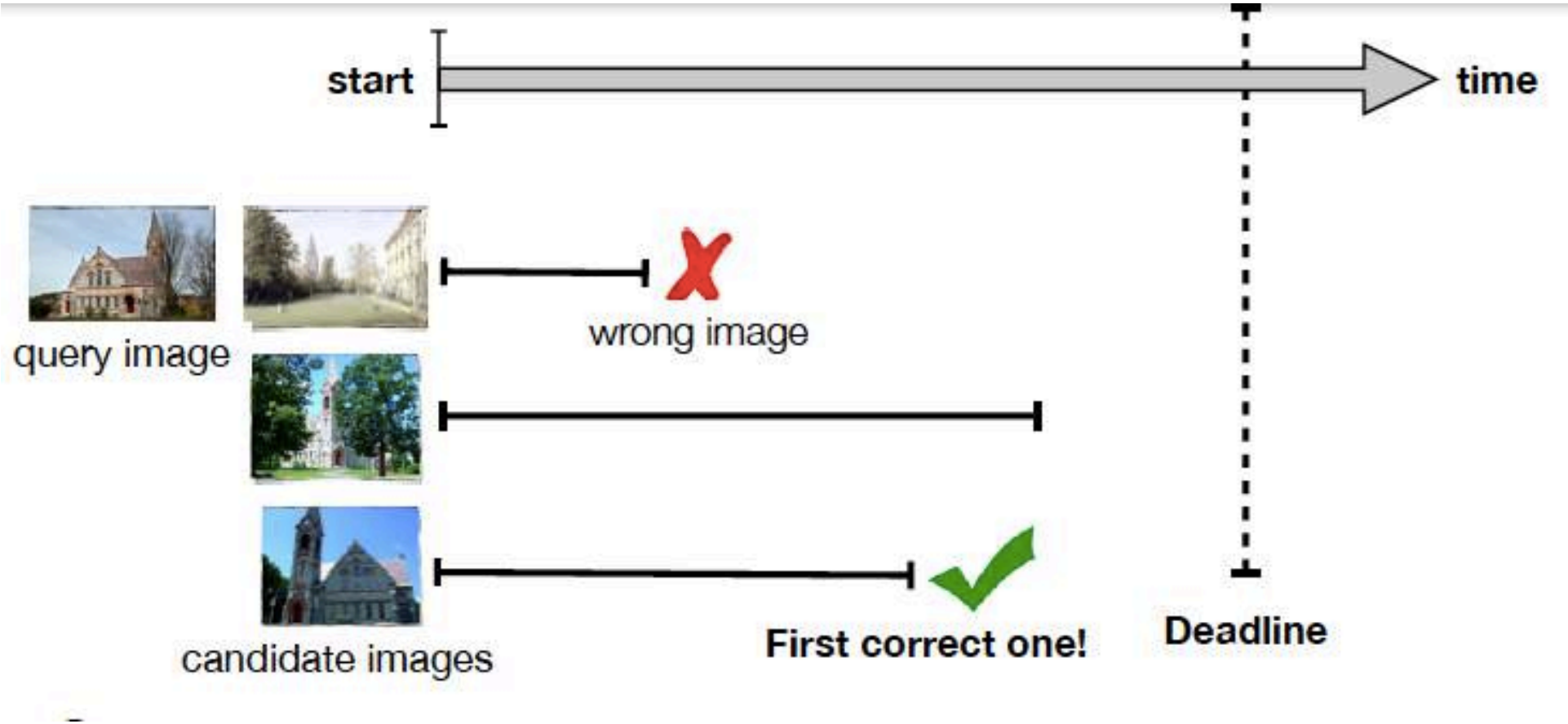
Select [Yan-MobiSys10]

- Ensuring accuracy with majority voting
- Given accuracy, optimize cost and latency
- Deadline as latency in mobile phones



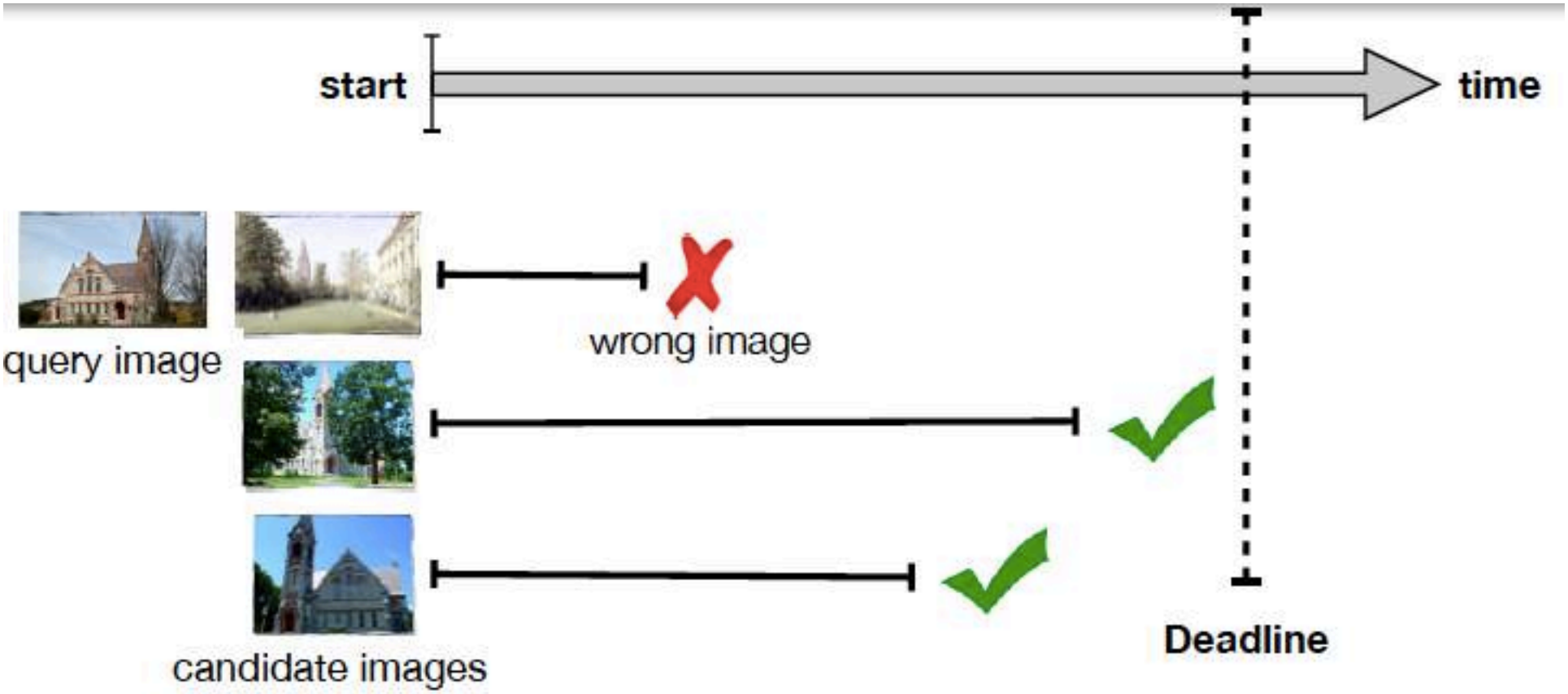
Select [Yan-MobiSys10]

- Goal: For a query image *Q*, find the first relevant image *I* with **min cost** before the **deadline**



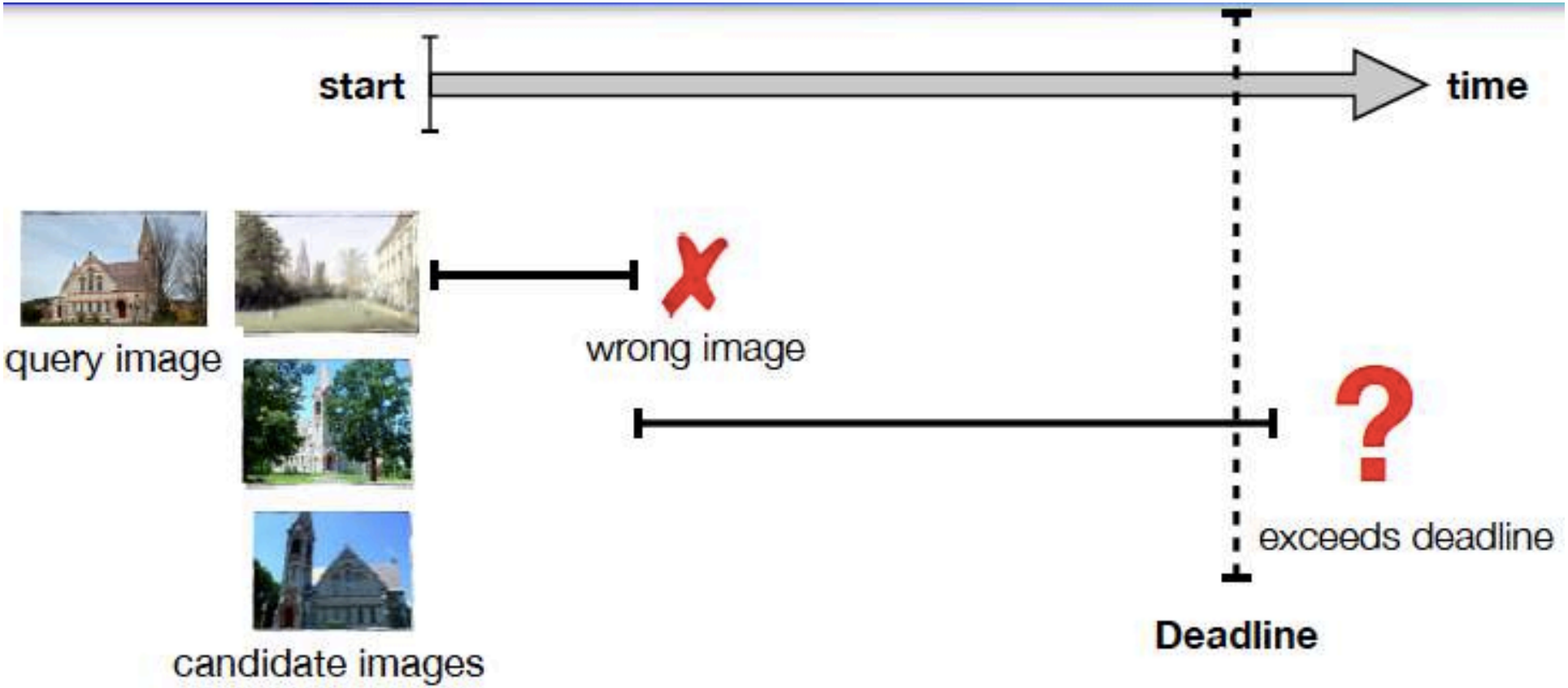
Select [Yan-MobiSys10]

- Parallel crowdsourced validation



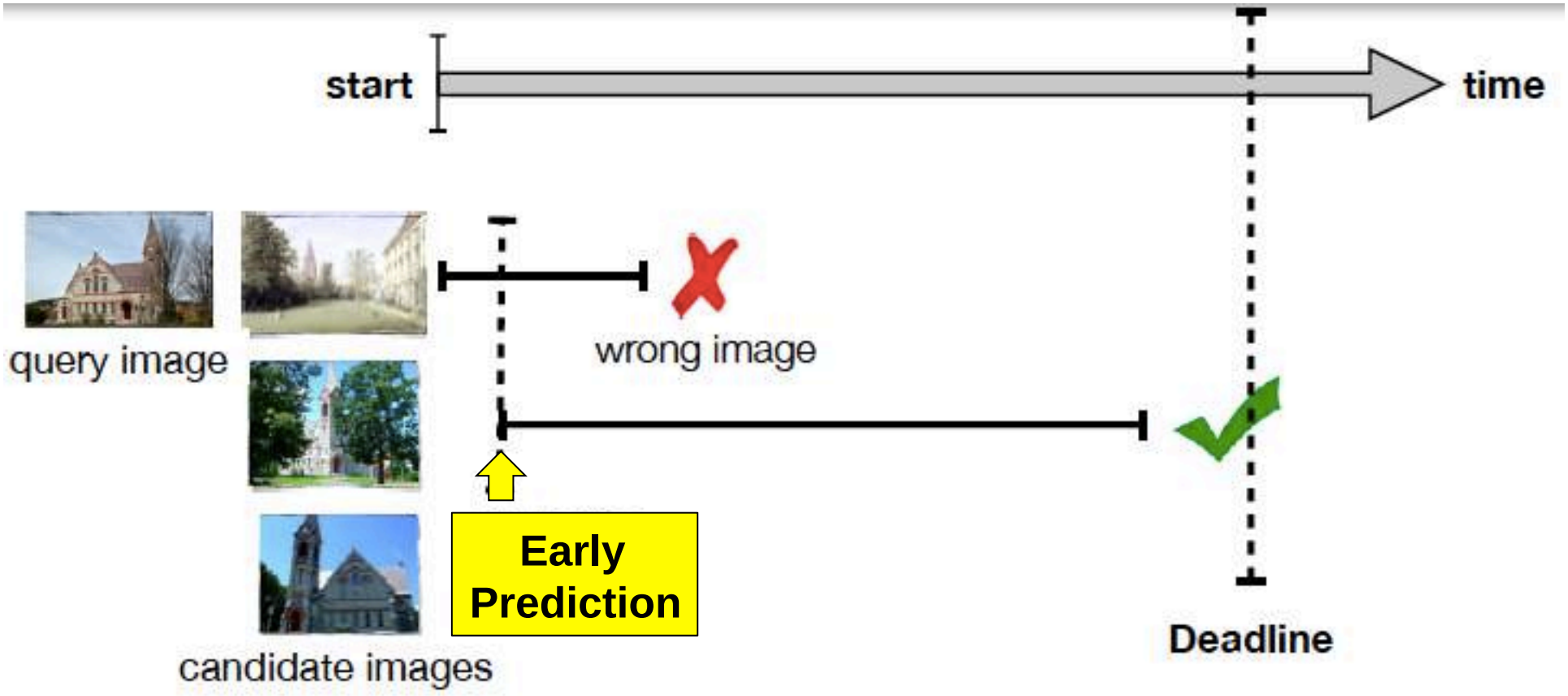
Select [Yan-MobiSys10]

- Sequential crowdsourced validation



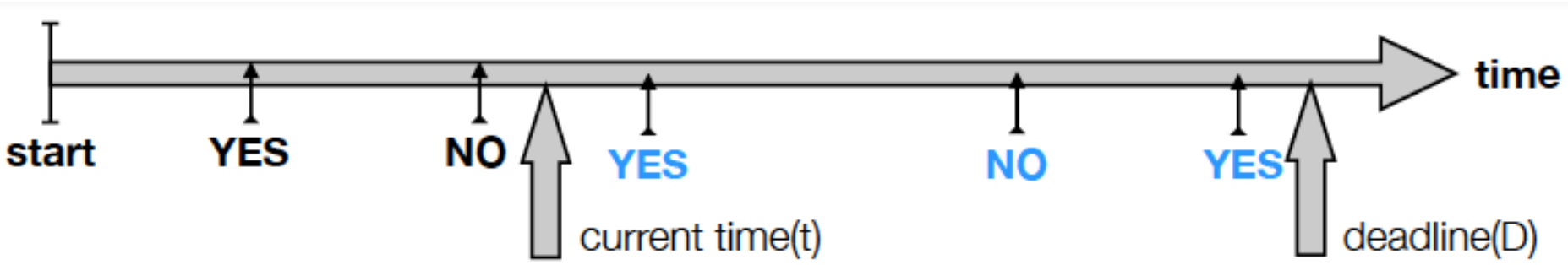
Select [Yan-MobiSys10]

- CrowdSearch: using early prediction on the delay and outcome to start the validation of next candidate early

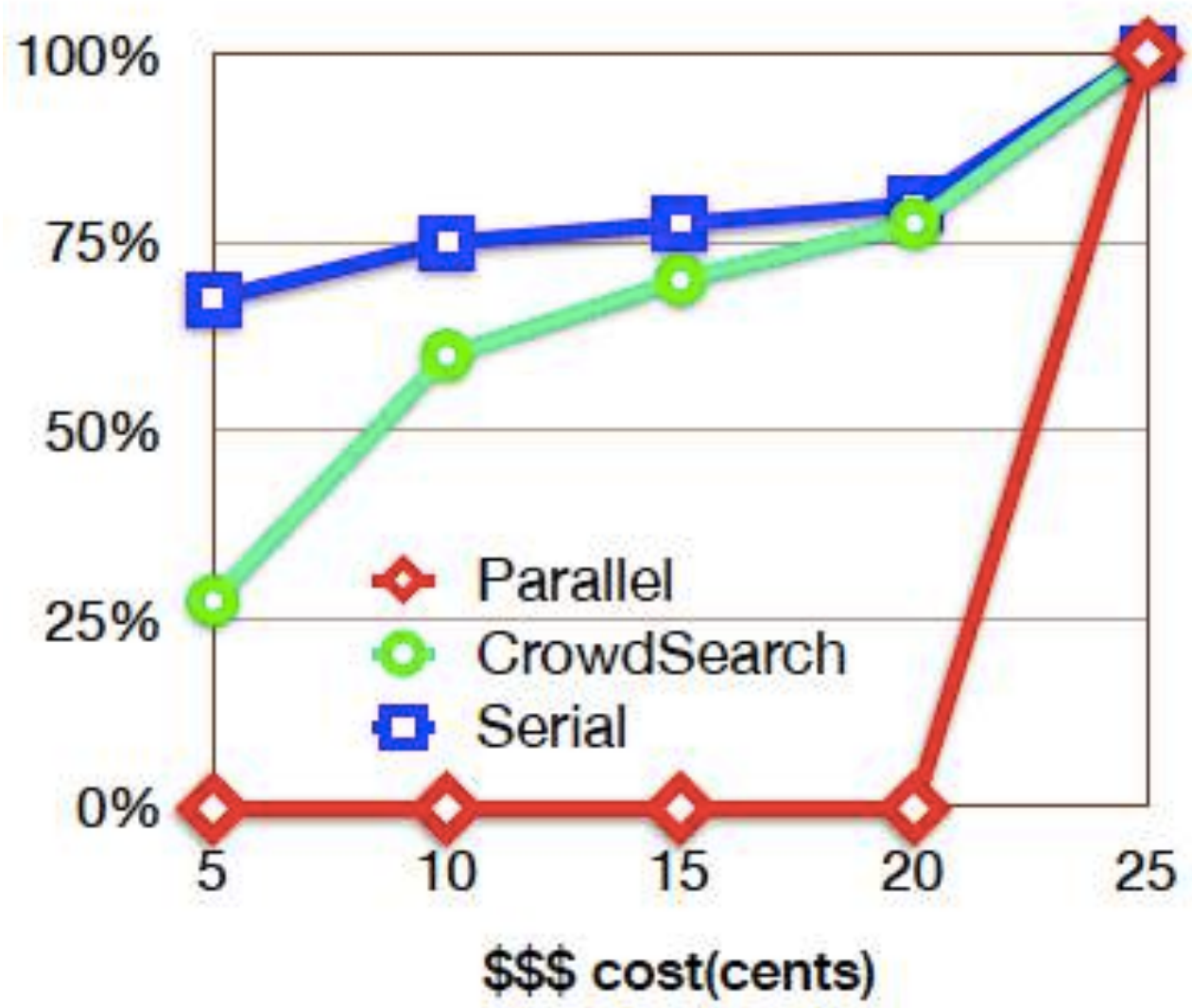


Select [Yan-MobiSys10]

- Predicting accuracy
- Eg, at time *t*
  - 2 responses so far (1 Yes, and 1 No)
  - From training data, list all majority-vote(5)=Yes
  - Determine probability



Select [Yan-MobiSys10]



Count Operation

- **Given *N* items, estimate the number of *m* items that satisfy a predicate *P***

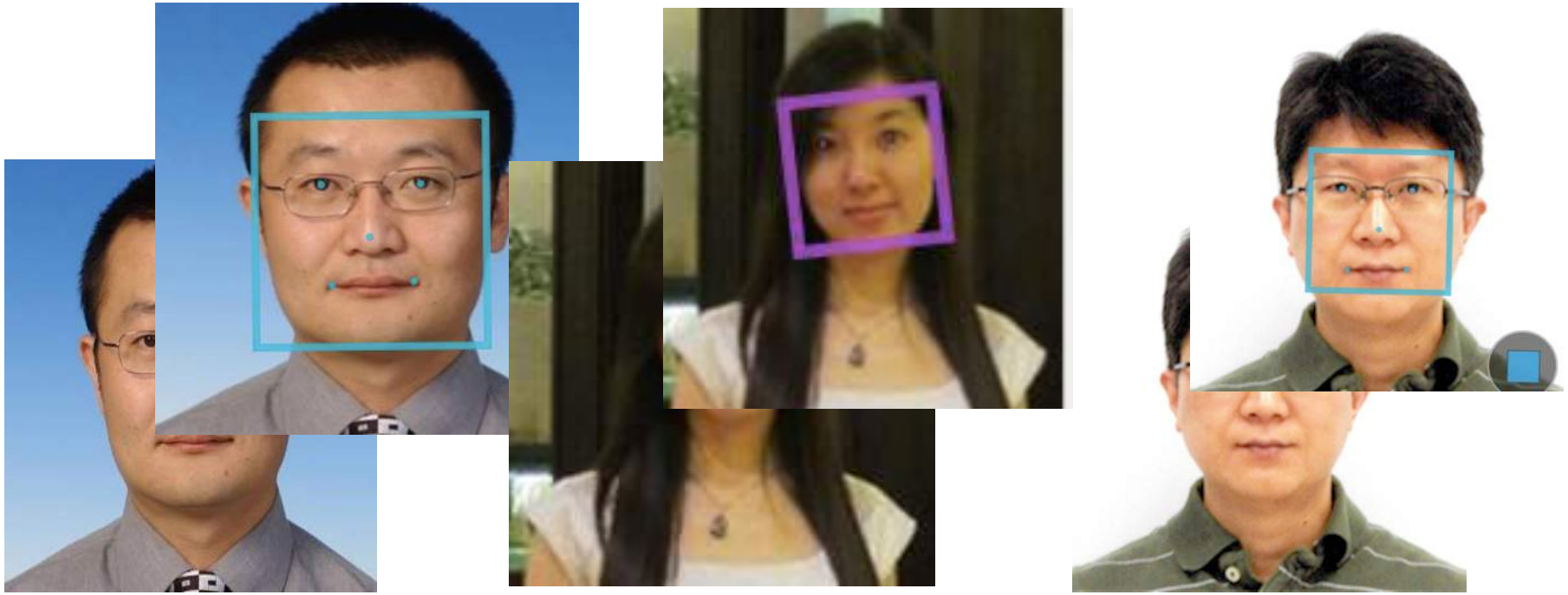
- Selectivity estimation in DB → crowd-powered query optimizers
- Evaluating queries with GROUP BY + COUNT/AVG/SUM operators
- Eg, “Find photos of females with red hairs”
  - Selectivity(“female”) ≈ 50%
  - Selectivity(“red hair”) ≈ 2%
  - Better to process predicate(“red hair”) first

Count Operation

- Q: “How many teens are participating in the Hong Kong demonstration?”

Count Operation

- Using Face++, guess the age of a person



7 - 42

~~**5 - 17**~~

~~**9 - 28**~~

http://www.faceplusplus.com/demo-detect/

Count [Marcus-VLDB13]

- Hypothesis: Humans can estimate the frequency of objects' properties in a batch without having to explicitly label each item

- Two approaches
  - #1: Label Count
    - Sampling based
    - Have workers label samples explicitly
  - #2: Batch Count
    - Have workers estimate the frequency in a batch

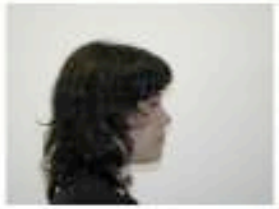
Count [Marcus-VLDB13]

- Label Count (via sampling)

There are 2 people below. Please identify the gender of each.

What is the gender of this person?

male female



What is the gender of this person?

male female

Submit

Count [Marcus-VLDB13]

- Batch Count

There are 10 people below. Please provide rough estimates for how many of the people have various properties.

About how many of the 10 people are male? 4

About how many of the 10 people are female?

Submit

Count [Marcus-VLDB13]

- Findings on accuracy
  - Images: Batch count > Label count
  - Texts: Batch count < Label count

- Further Contributions
  - Detecting spammers
  - Avoiding coordinated attacks

Join Operation

- **Identify matching records or entities within or across tables**
  - ≈ similarity join, entity resolution (ER), record linkage, de-duplication, …
  - Beyond the exact matching
- [Chaudhuri-ICDE06] similarity join
  - R JOIN$_p$ S, where p=$sim(R.A, S.A) > t$
  - *sim()* can be implemented as UDFs in SQL
  - Often, the evaluation is expensive
    - DB applies UDF-based join predicate after Cartesian product of R and S

Join Operation

- Examples
  - **[Marcus-VLDB11]** proposes 3 types of joins
  - **[Wang-VLDB12]** generates near-optimal cluster-based HIT design to reduce join cost
  - **[Wang-SIGMOD13]** reduces join cost further by exploiting transitivity among items
  - **[Whang-VLDB13]** selects right questions to ask to crowds to improve join accuracy
  - **[Gokhale-SIGMOD14]** proposes the hands-off crowdsourcing for join workflow

Join [Marcus-VLDB11]

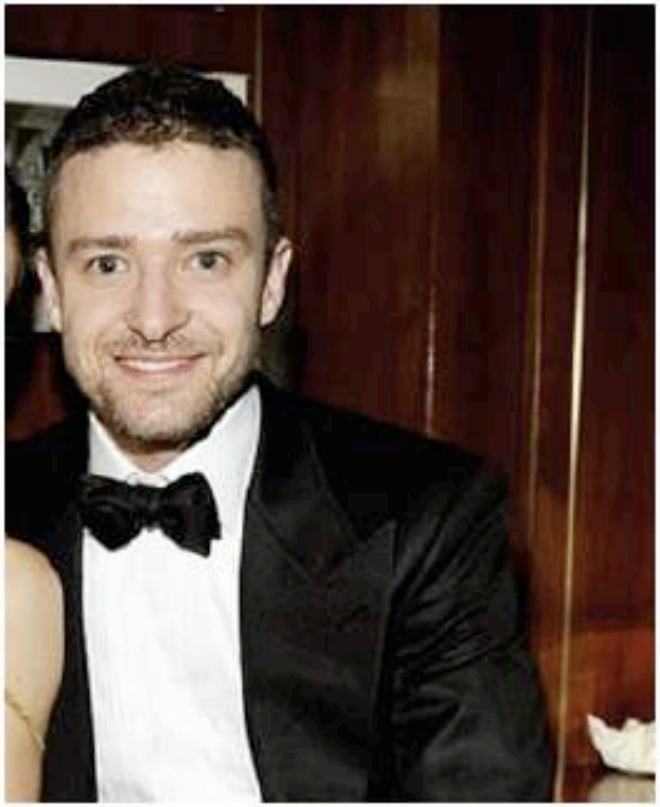
- To join tables *R* and *S*
- #1: Simple Join
  - Pair-wise comparison HIT
  - |*R*||*S*| HITs needed
- #2: Naïve Batching Join
  - Repetition of #1 with a batch factor *b*
  - |*R*||*S*|/*b* HITs needed
- #3: Smart Batching Join
  - Show *r* and *s* images from *R* and *S*
  - Workers pair them up
  - |*R*||*S*|/*rs* HITs needed

Join [Marcus-VLDB11]

Is the same celebrity in the image on the left and the image on the right?

#1 Simple Join

Yes No

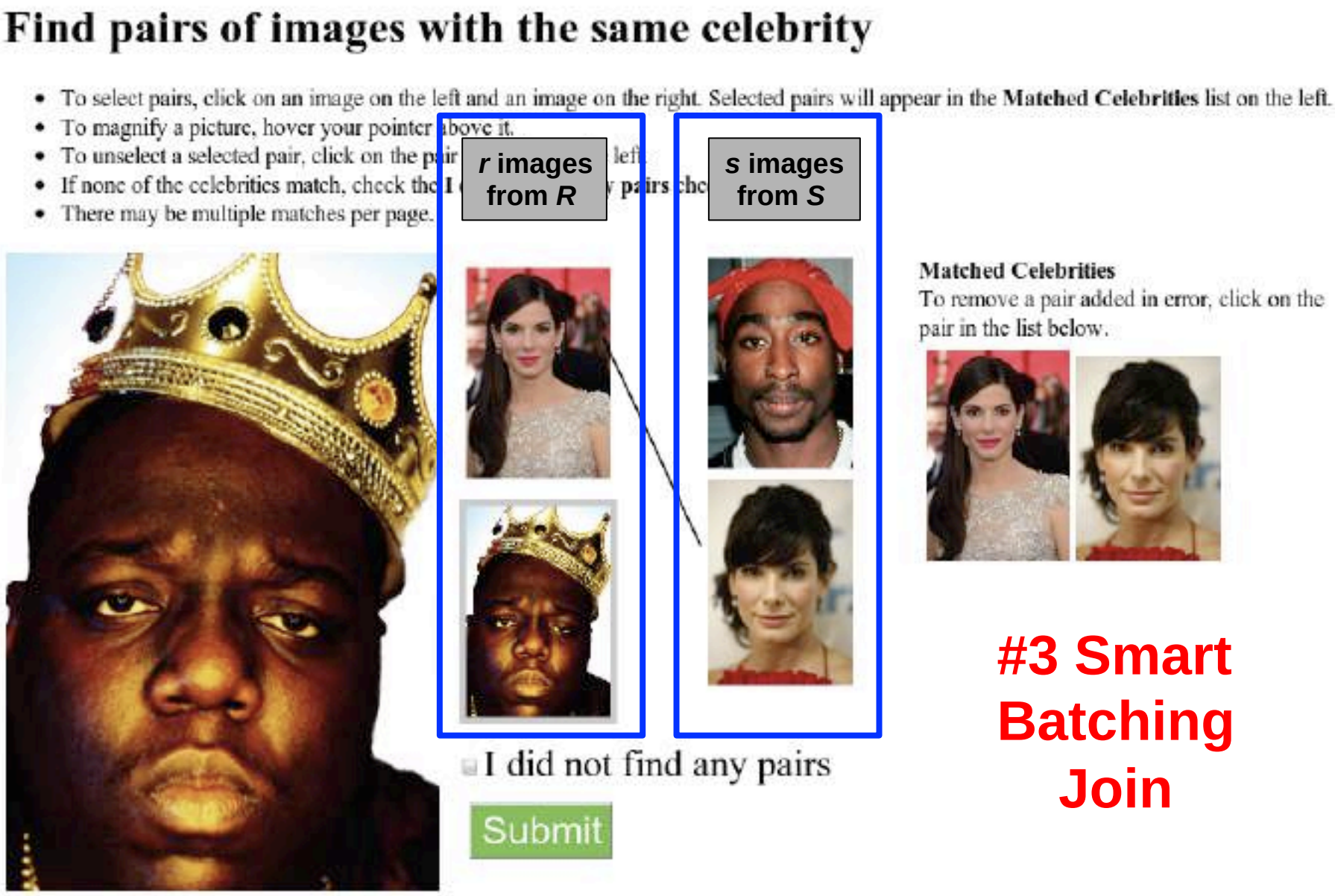


Join [Marcus-VLDB11]

#2 Naïve Batching Join

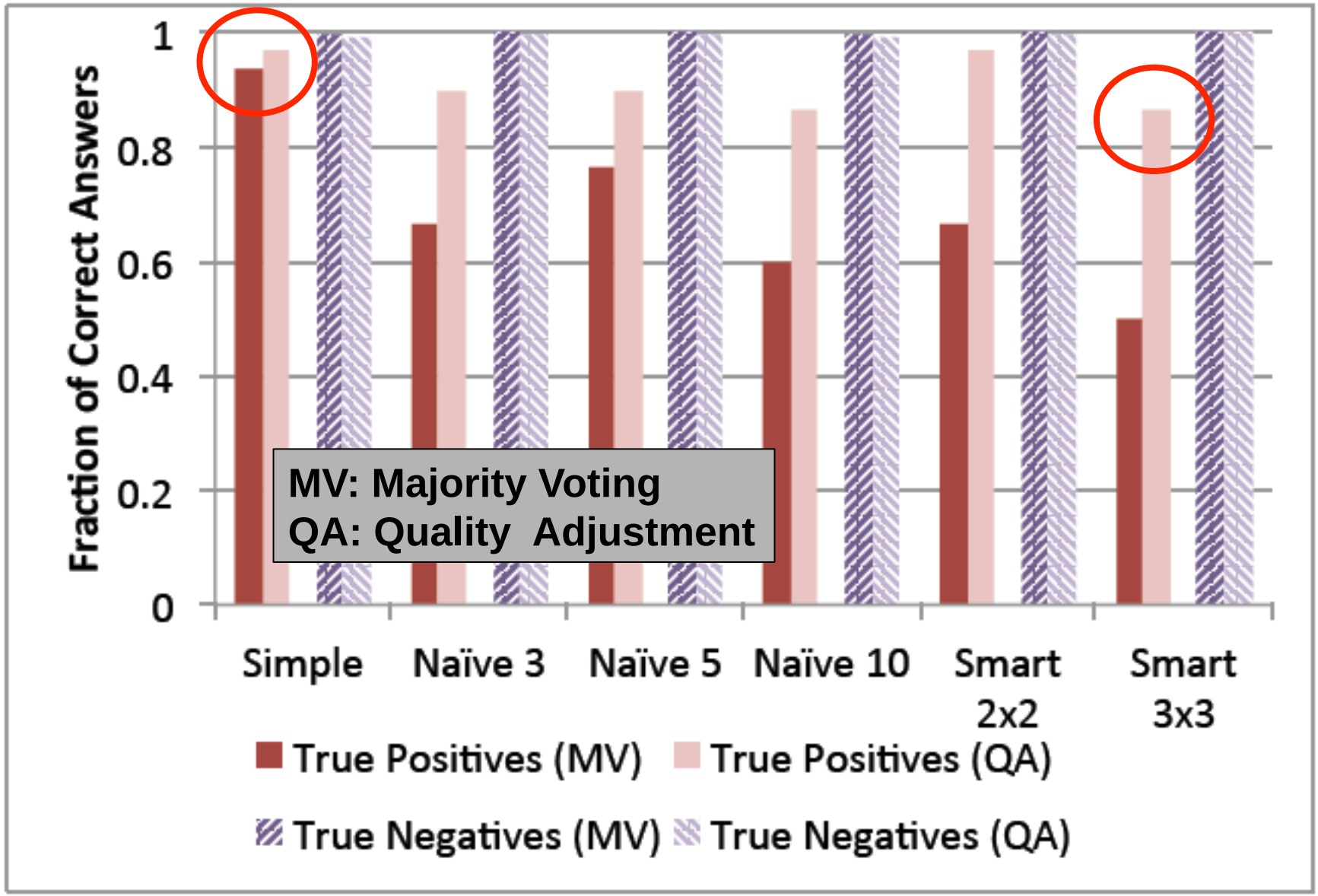
Batch factor
***b* = 2**

Join [Marcus-VLDB11]



#3 Smart Batching Join

Join [Marcus-VLDB11]



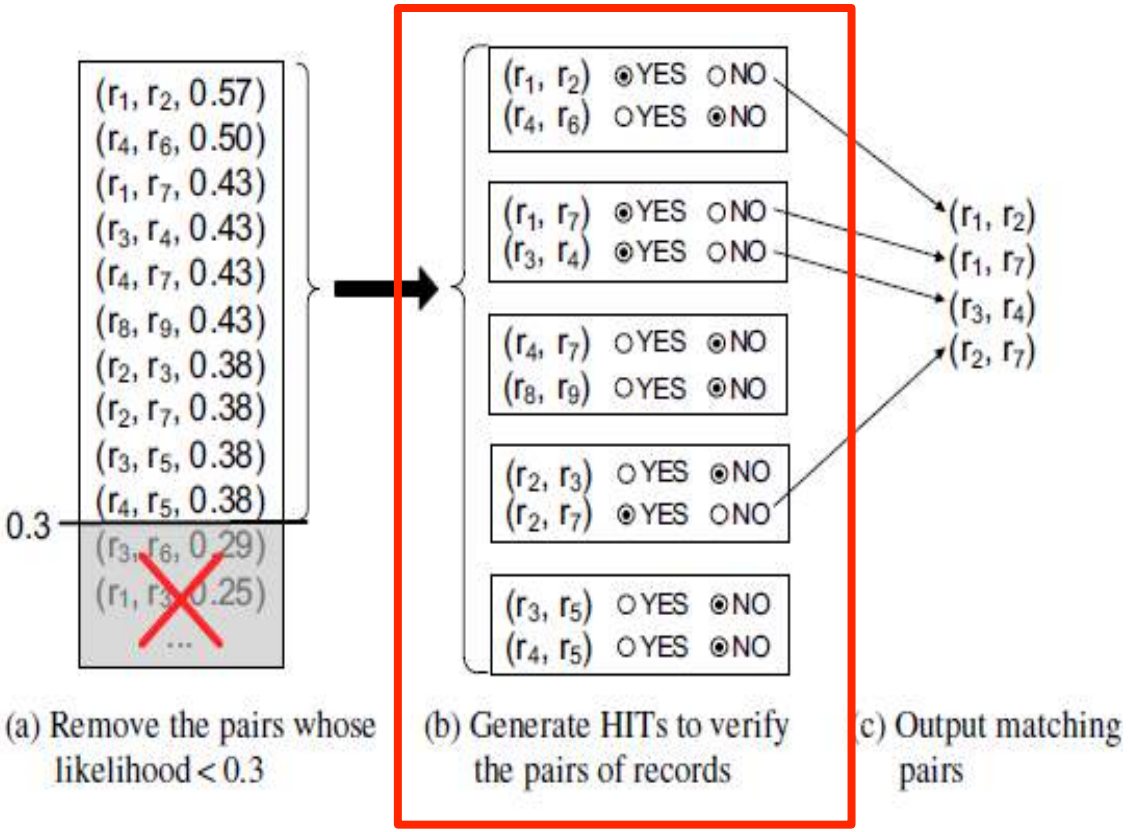
Join [Wang-VLDB12]

- [Marcus-VLDB11] proposed two batch joins
  - More efficient smart batch join still generates $|R||S|/rs$ # of HITs
  - Eg, (10,000 X 10,000) / (20 x 20) = 250,000 HITs
    → Still too many !
- [Wang-VLDB12] contributes CrowdER:
  1. A hybrid human-machine join
     - #1 machine-join prunes obvious non-matches
     - #2 human-join examines likely matching cases
       - Eg, candidate pairs with high similarity scores
  2. Algorithm to generate min # of HITs for step #2

Join [Wang-VLDB12]

- Hybrid idea: generate candidate pairs using existing similarity measures (eg, Jaccard)

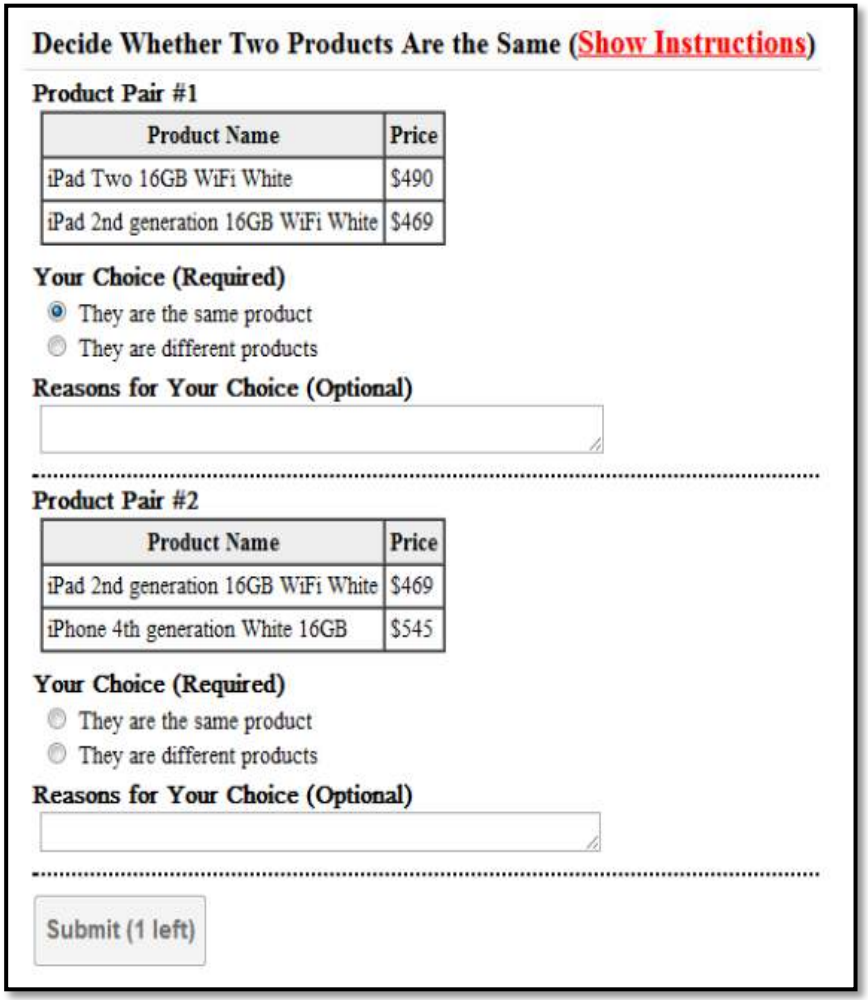
ID	Product Name	Price
r_1	iPad Two 16GB WiFi White	$490
r_2	iPad 2nd generation 16GB WiFi White	$469
r_3	iPhone 4th generation White 16GB	$545
r_4	Apple iPhone 4 16GB White	$520
r_5	Apple iPhone 3rd generation Black 16GB	$375
r_6	iPhone 4 32GB White	$599
r_7	Apple iPad2 16GB WiFi White	$499
r_8	Apple iPod shuffle 2GB Blue	$49
r_9	Apple iPod shuffle USB Cable	$19



Main Issue: HIT Generation Problem

Join [Wang-VLDB12]

Pair-based HIT Generation ≈ Naïve Batching in [Marcus-VLDB11]



Decide Whether Two Products Are the Same (Show Instructions)

Product Pair #1

Product Name	Price
iPad Two 16GB WiFi White	$490
iPad 2nd generation 16GB WiFi White	$469

Your Choice (Required)

- (•) They are the same product
- ( ) They are different products

Reasons for Your Choice (Optional)

Product Pair #2

Product Name	Price
iPad 2nd generation 16GB WiFi White	$469
iPhone 4th generation White 16GB	$545

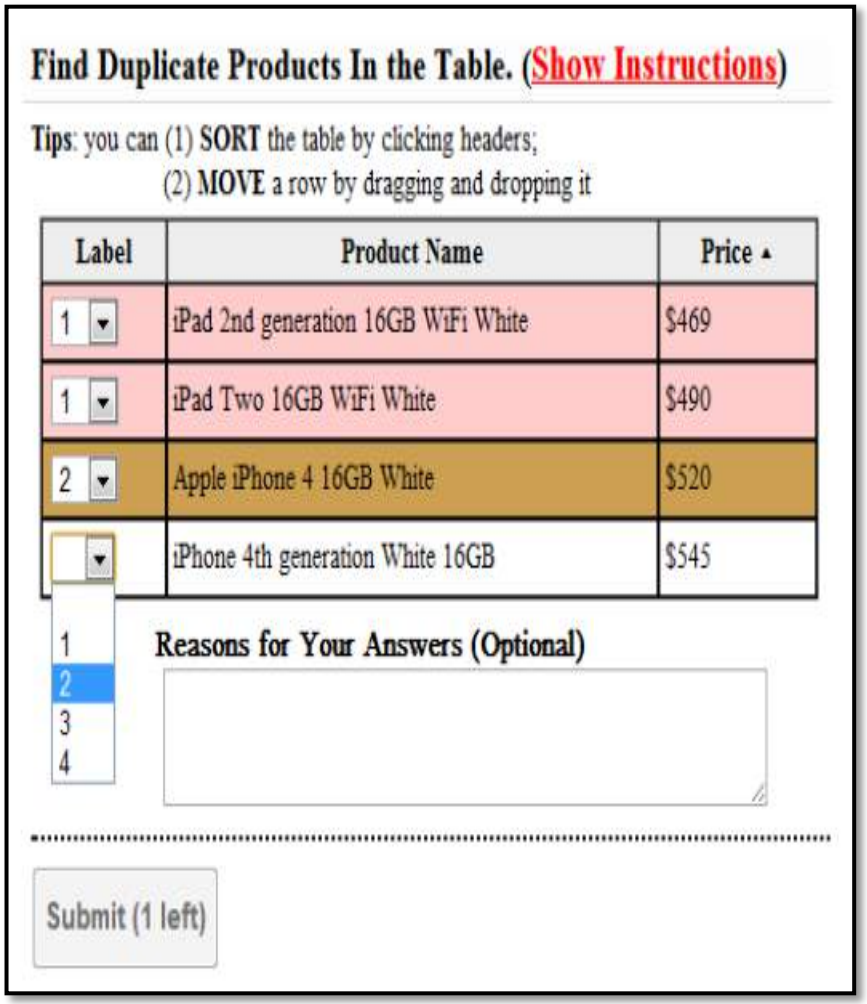
Your Choice (Required)

- ( ) They are the same product
- ( ) They are different products

Reasons for Your Choice (Optional)

Submit (1 left)

Cluster-based HIT Generation ≈ Smart Batching in [Marcus-VLDB11]



Find Duplicate Products In the Table. (Show Instructions)

Tips: you can (1) SORT the table by clicking headers;
(2) MOVE a row by dragging and dropping it

Label	Product Name	Price ▲
1	iPad 2nd generation 16GB WiFi White	$469
1	iPad Two 16GB WiFi White	$490
2	Apple iPhone 4 16GB White	$520
	iPhone 4th generation White 16GB	$545

1
2
3
4

Reasons for Your Answers (Optional)

Submit (1 left)

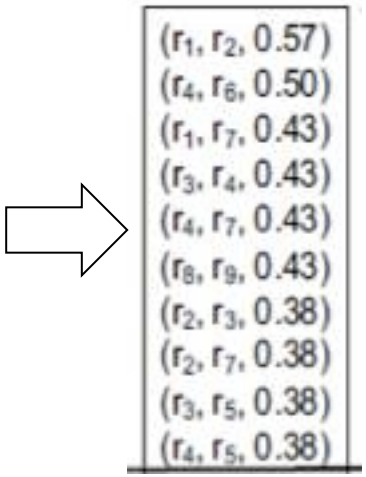
Join [Wang-VLDB12]

- HIT Generation Problem
  - Input: pairs of records *P*, # of records in HIT *k*
  - Output: minimum # of HITs s.t.
    1. All HITs have at most *k* records
    2. Each pair $(p_i, p_j) \in P$ must be in at least one HIT

1. Pair-based HIT Generation
   - Trivial: *P/k* # of HITs s.t. each HIT contains *k* pairs in *P*
2. Cluster-based HIT Generation
   - NP-hard problem → approximation solution

Join [Wang-VLDB12]

ID	Product Name	Price
r_1	iPad Two 16GB WiFi White	$490
r_2	iPad 2nd generation 16GB WiFi White	$469
r_3	iPhone 4th generation White 16GB	$545
r_4	Apple iPhone 4 16GB White	$520
r_5	Apple iPhone 3rd generation Black 16GB	$375
r_6	iPhone 4 32GB White	$599
r_7	Apple iPad2 16GB WiFi White	$499
r_8	Apple iPod shuffle 2GB Blue	$49
r_9	Apple iPod shuffle USB Cable	$19



***k* = 4**

Cluster-based HIT #1

r_1, r_2, r_3, r_7

Cluster-based HIT #2

r_3, r_4, r_5, r_6

Cluster-based HIT #3

r_4, r_7, r_8, r_9

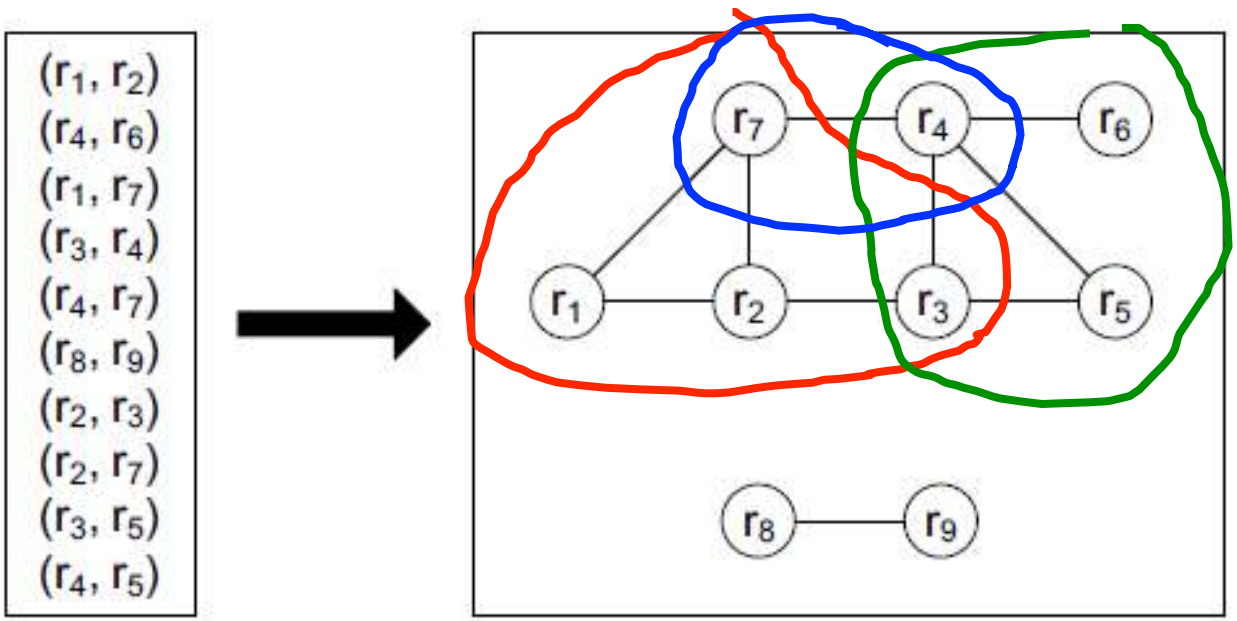
This is the minimal # of cluster-based HITs satisfying previous two conditions

Join [Wang-VLDB12]

- Two-tiered Greedy Algorithm
  - Build a graph *G* from pairs of records in *P*
  - CC ← connected components in *G*
    - LCC: large CC with more than *k* nodes
    - SCC: small CC with no more than *k* nodes
  - Step 1: Partition LCC into SCCs
  - Step 2: Pack SCCs into HITs with *k* nodes
    - Integer programming based

Join [Wang-VLDB12]

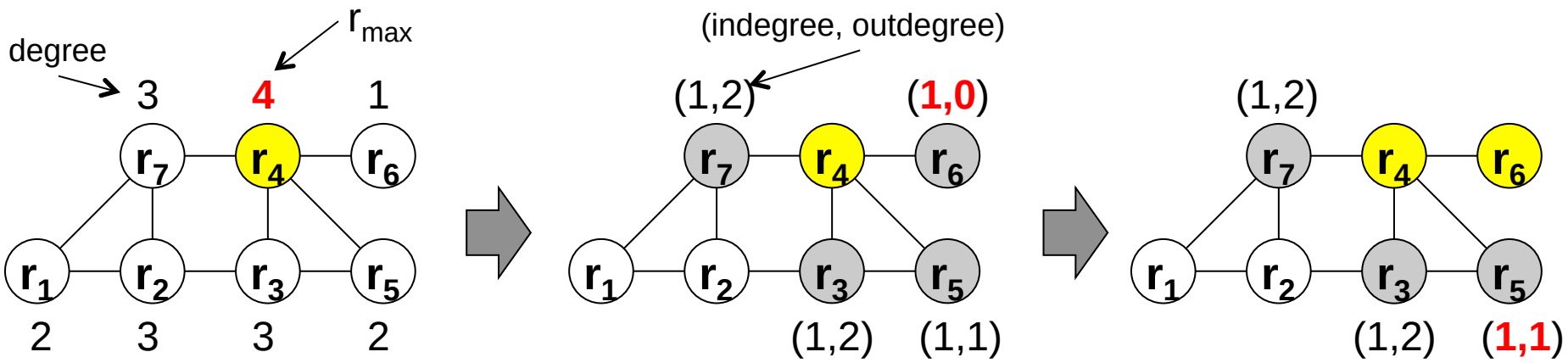
- Eg, Generate cluster-based HITs (k = 4)
  1. Partition the LCC into 3 SCCs
     - $\{r_1, r_2, r_3, r_7\}$, $\{r_3, r_4, r_5, r_6\}$, $\{r_4, r_7\}$
  2. Pack SCCs into HITs
     - A single HIT per $\{r_1, r_2, r_3, r_7\}$ and $\{r_3, r_4, r_5, r_6\}$
     - Pack $\{r_4, r_7\}$ and $\{r_8, r_9\}$ into a HIT



Join [Wang-VLDB12]

- Step 1: Partition
  - Input: LCC, *k* Output: SCCs
  - r_{max} ← node in LCC with the max degree
  - scc ← {r_{max}}
  - conn ← nodes in LCC directly connected to r_{max}
  - while |scc| < k and |conn| > 0
    - r_{new} ← node in conn with max indegree (# of edges to scc) and min outdegree (# of edges to non-scc) if tie
    - move r_{new} from conn to scc
    - update conn using new scc
  - add scc into SCC

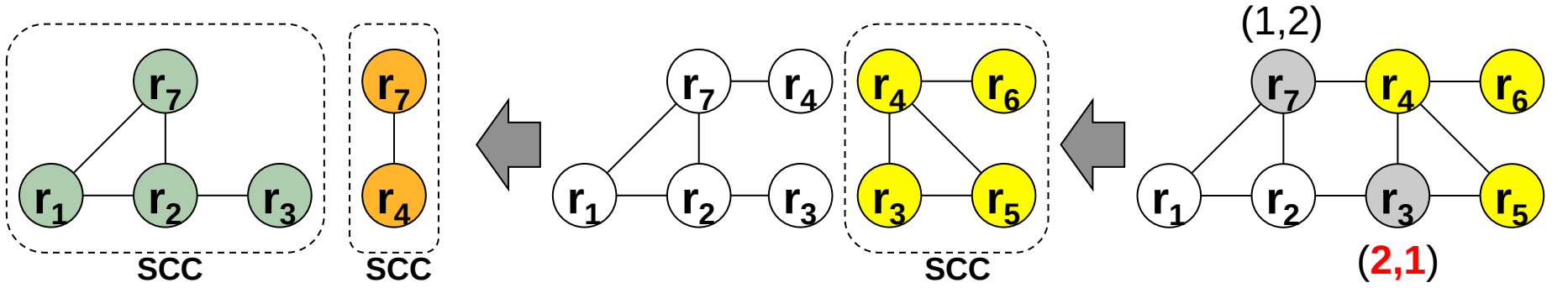
Join [Wang-VLDB12]



Initialize SCC={r_4}

conn={r_3, r_5, r_6, r_7}
Add r_6 into SCC

conn={r_3, r_5, r_7}
Add r_5 into SCC



conn={r_3, r_7}
Add r_3 into SCC

Output SCC

Output other SCC

Part 2 Summary

- New opportunities and challenges
  - Open-world assumption
  - Non-deterministic algorithmic behavior
  - Trade-off among cost, latency, and accuracy
- Human-Powered DB → "**Human-in-the-loop**" DB
  - Machines process majority of operations
  - Humans process a small fraction of challenging operations in big data

http://www.theoddblog.us/2014/02/21/damienwaltershumanloop/

Reference

- **[Chaudhuri-ICDE06]** A Primitive Operator for Similarity Join in Data Cleaning, Surajit Chaudhuri et al., ICDE 2006
- **[Davidson-ICDT13]** *Using the crowd for top-k and group-by queries***,** Susan Davidson et al., ICDT 2013
- **[Dwork-WWW01]** *Rank Aggregation Methods for the Web*, Cynthia Dwork et al., WWW 2001
- **[Franklin-SIGMOD11]** *CrowdDB: answering queries with crowdsourcing*, Michael J. Franklin et al, SIGMOD 2011
- **[Franklin-ICDE13]** *Crowdsourced enumeration queries*, Michael J. Franklin et al, ICDE 2013
- **[Gokhale-SIGMOD14]** *Corleone: Hands-Off Crowdsourcing for Entity Matching*, Chaitanya Gokhale et al., SIGMOD 2014
- **[Guo-SIGMOD12]** *So who won?: dynamic max discovery with the crowd*, Stephen Guo et al., SIGMOD 2012
- **[Marcus-VLDB11]** *Human-powered Sorts and Joins*, Adam Marcus et al., VLDB 2011
- **[Marcus-VLDB12]** *Counting with the Crowd*, Adam Marcus et al., VLDB 2012

Reference

- **[Parameswaran-SIGMOD12]** *CrowdScreen: Algorithms for Filtering Data with Humans*, Aditya Parameswaran et al., SIGMOD 2012
- **[Polychronopoulous-WebDB13]** *Human-Powered Top-k Lists*, Vassilis Polychronopoulous et al., WebDB 2013
- **[Sarma-ICDE14]** Crowd-Powered Find Algorithms, Anish Das Sarma et al., ICDE 2014
- **[Venetis-WWW12]** *Max Algorithms in Crowdsourcing Environments*, Petros Venetis et al., WWW 2012
- **[Wang-VLDB12]** *CrowdER: Crowdsourcing Entity Resolution*, Jiannan Wang et al., VLDB 2012
- **[Wang-SIGMOD13]** *Leveraging Transitive Relations for Crowdsourced Joins*, Jiannan Wang et al., SIGMOD 2013
- **[Whang-VLDB13]** *Question Selection for Crowd Entity Resolution*, Steven Whang et al., VLDB 2013
- **[Yan-MobiSys10]** *CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones*, T. Yan et al., MobiSys 2010

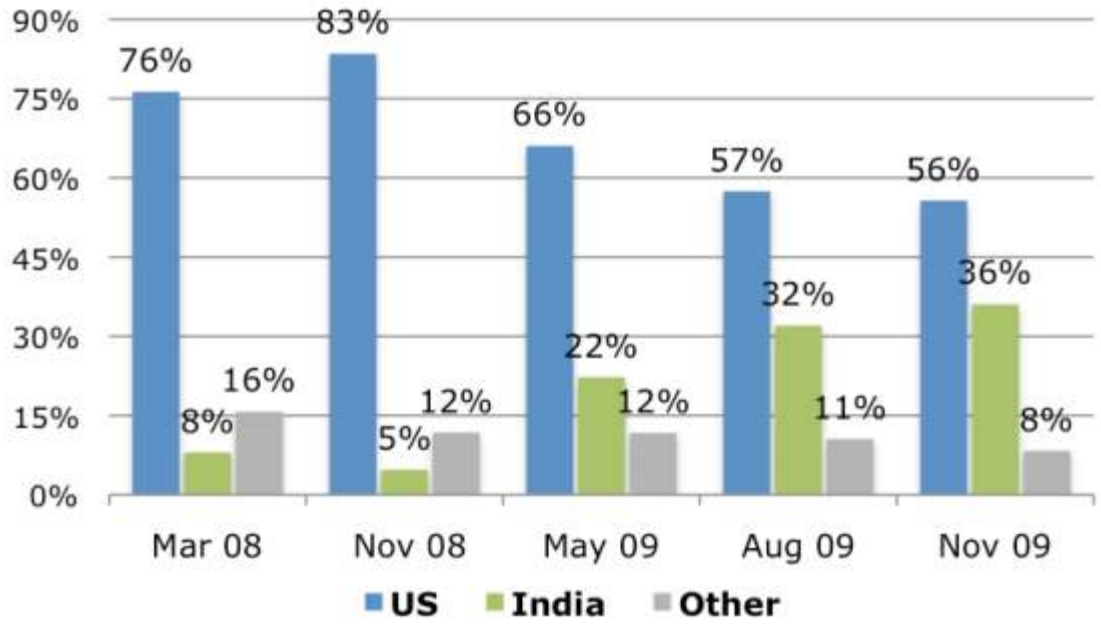
Part 3
QUALITY CONTROL

Challenges

- Data Quality is always the **first concern** in crowdsourcing applications

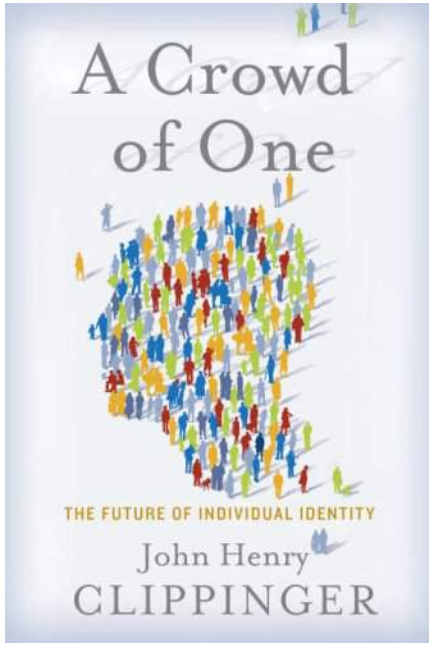
Challenges

- Bias
  - Narrowed Demographics
    - Mainly women in U.S. and India



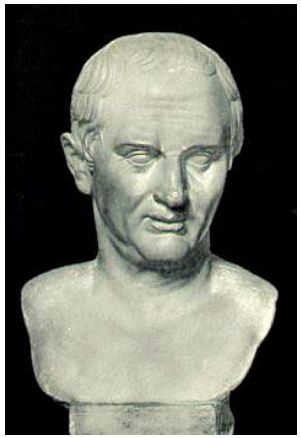
Workers on AMT

Challenges



- **Wisdom is within the human.**
- **We need to dig it out, and *manage* them.**
- **Well, it is difficult, because the humans are …**

Humans



- Erroneous

  "To err is human"

  —Marcus Tullius Cicero

- Greedy

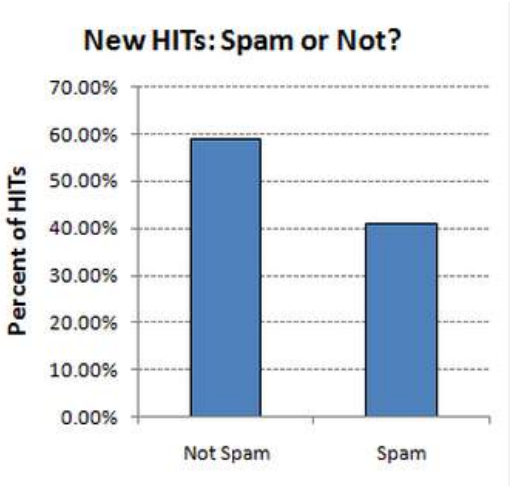
  "…my more-having would be a sauce to make me hunger more"

  —Macbeth act 4, sc. 3, Shakespeare

Challenges

- Spammers
  - Spam Worker
    - Finish tasks simply for rewards
    - Low quality of the answers

  - Spam HITs
    - Some tasks are to spamming "social media" metrics
    - Some tricks the worker by refuse to pay for their answers



Quality Control

- Qualification Tests
- Aggregation Methods (Majority, Rasch)
- Golden Tests and Estimation
- Task Design
- Feedback
- Reputation (trustworthy crowdsourcing)

Task Design Example

- ESP Game(Luis von Ahn)
  - Object: Images Labeling
  - Human task: online game, two players guessing one common item

PLAYER 1

GUESSING: CAR

GUESSING: HAT

GUESSING: KID

SUCCESS!
YOU AGREE ON CAR

PLAYER 2

GUESSING: BOY

GUESSING: CAR

SUCCESS!
YOU AGREE ON CAR

Quality Control

- Whom to Ask
- WiseMarket
- COPE

Social Network/Media services

the **virtualization** and **digitalization**

of people's social activities

calebcc

If take a cab, can I get to Gyeongbokgung(경복궁) from 63 Building in one hour? @zhiyangsu @童咏昕 @SeoulHeart @ocar_liv @刘工bong

10秒前 来自新浪微博 粉丝头条 | 转发 | 收藏 | 评论

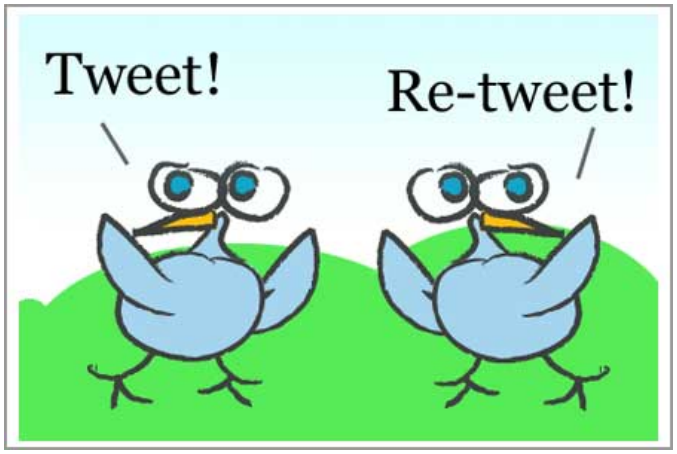
- Minor as dressing for a banquet
- Major as prediction of macro economy trends

“two-option decision making tasks”

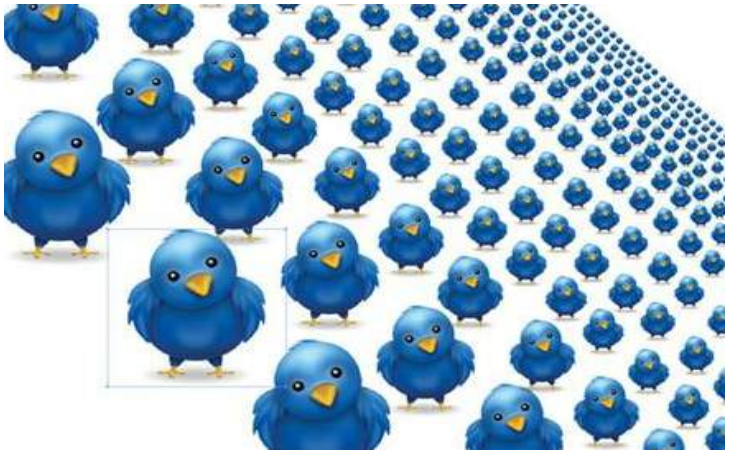
Can we extend the magic power of **Crowdsourcing** onto **social network**?

Microblog Users

- Simple
  - 140 characters
  - ‘RT’ + ‘@’

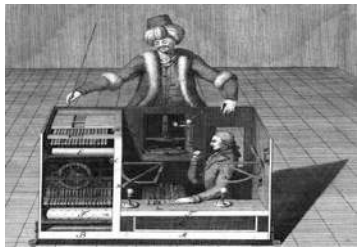


- **But comprehensive**
  - **Large network**
  - **Various**



Why Microblog Platform?

Twitter



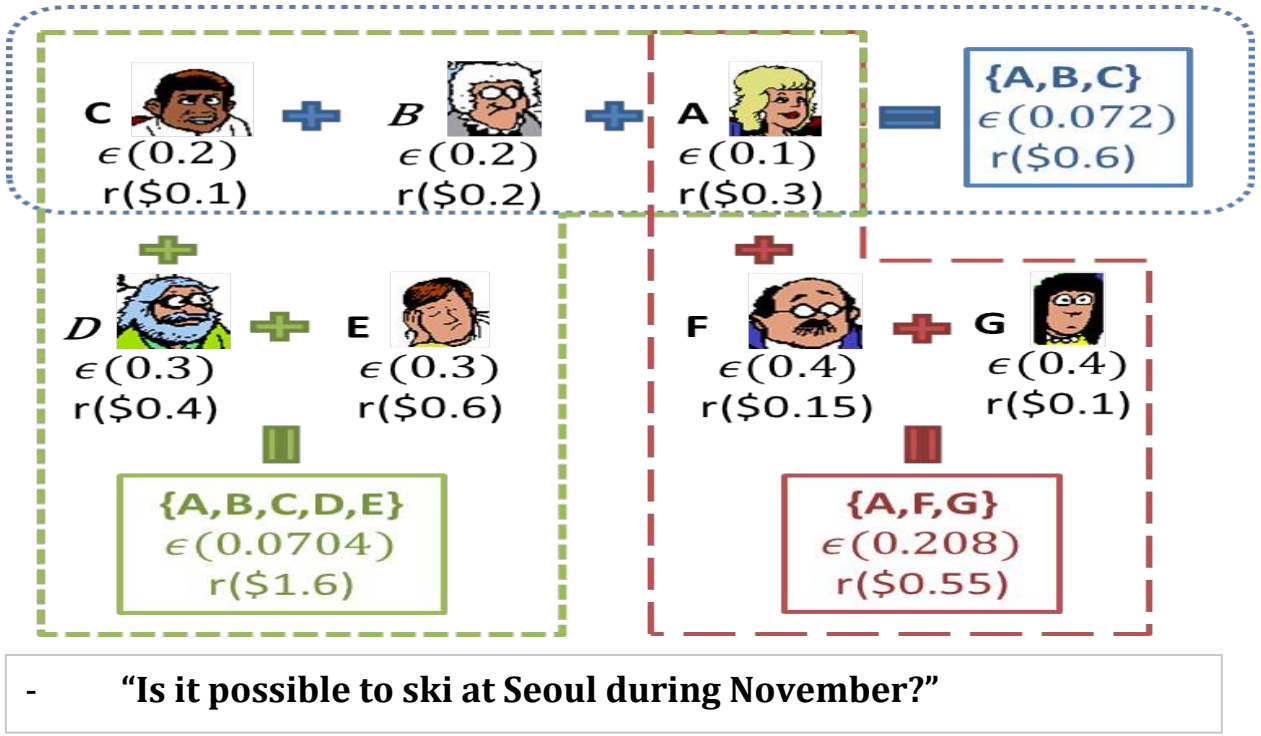
AMT

Accessibility	Highly convenient, on all kinds of mobile devices	Specific online platform
Incentive	Altruistic or payment	Mostly monetary incentive
Supported tasks	Simple task as decision making	Various types of tasks
Communication Infrastructure	'Tweet' and 'Reply' are enough	Complex workflow control mechanism
Worker Selection	**Active**, Enabled by '@'	Passively, No exact selection

Running Example

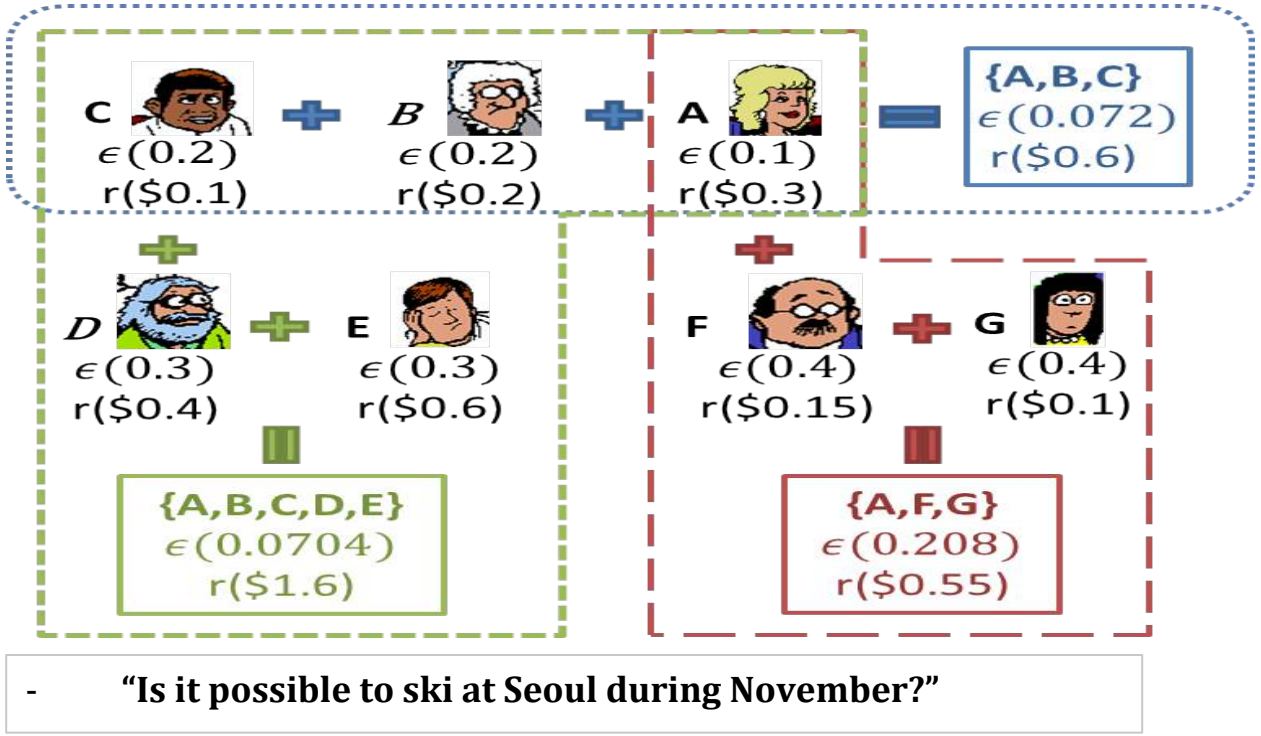
- **"Is it possible to ski at Seoul during November?"**

Motivation – Jury Selection Problem Running Case(1)



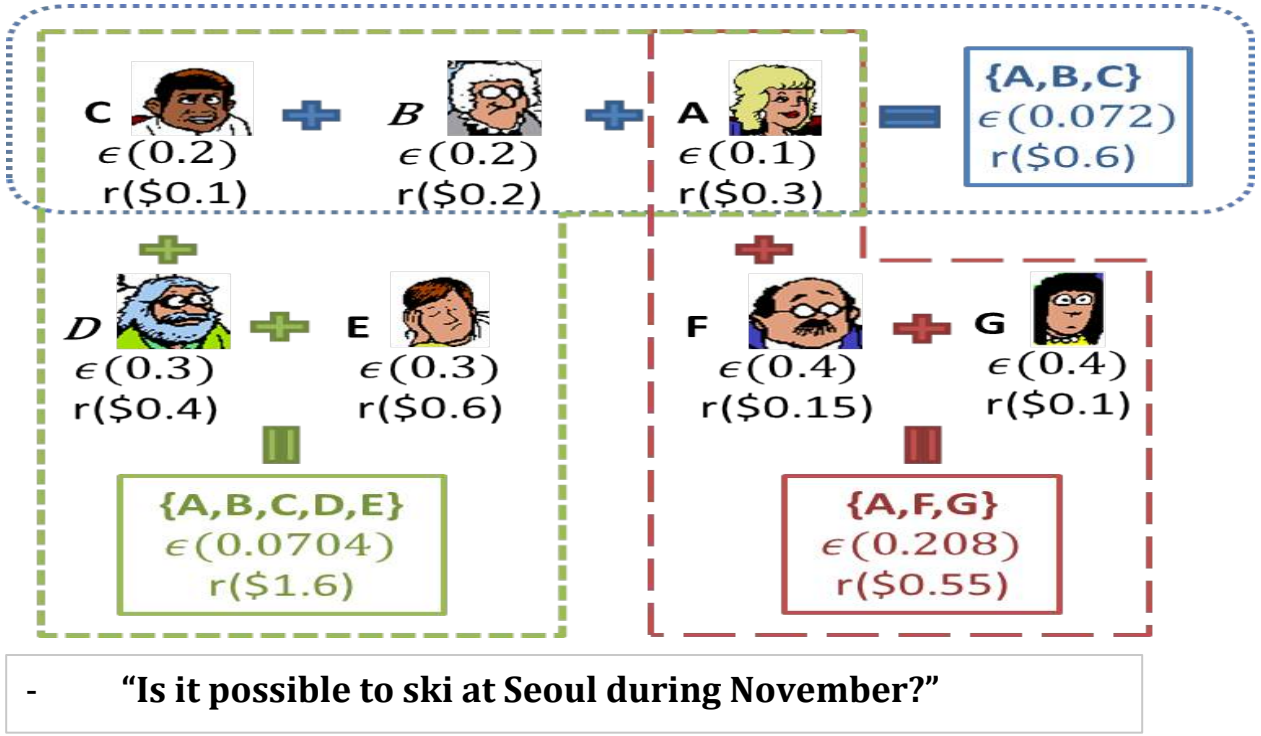
- Given a decision making problem, with budget $1, **whom should we ask**?

Motivation – Jury Selection Problem Running Case(2)



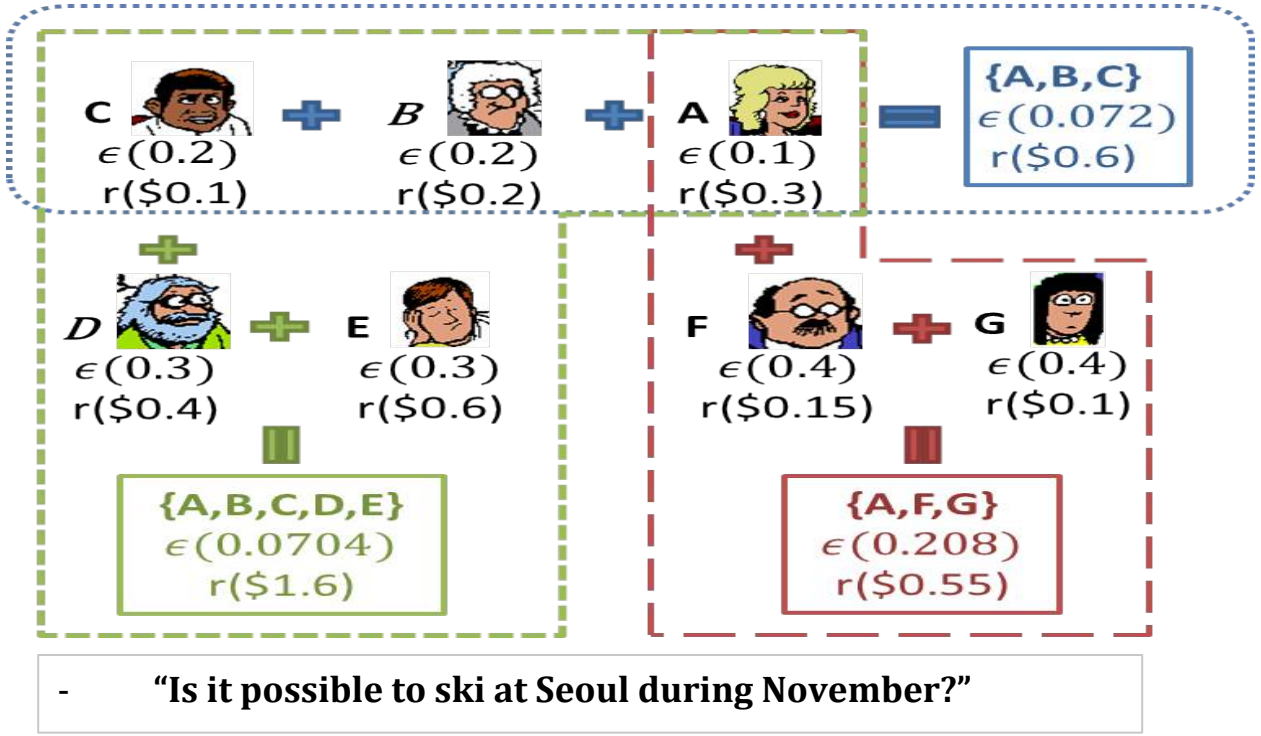
- Worker : Juror
- Crowds : Jury
- Data Quality : Jury Error Rate

Motivation – Jury Selection Problem Running Case(3)



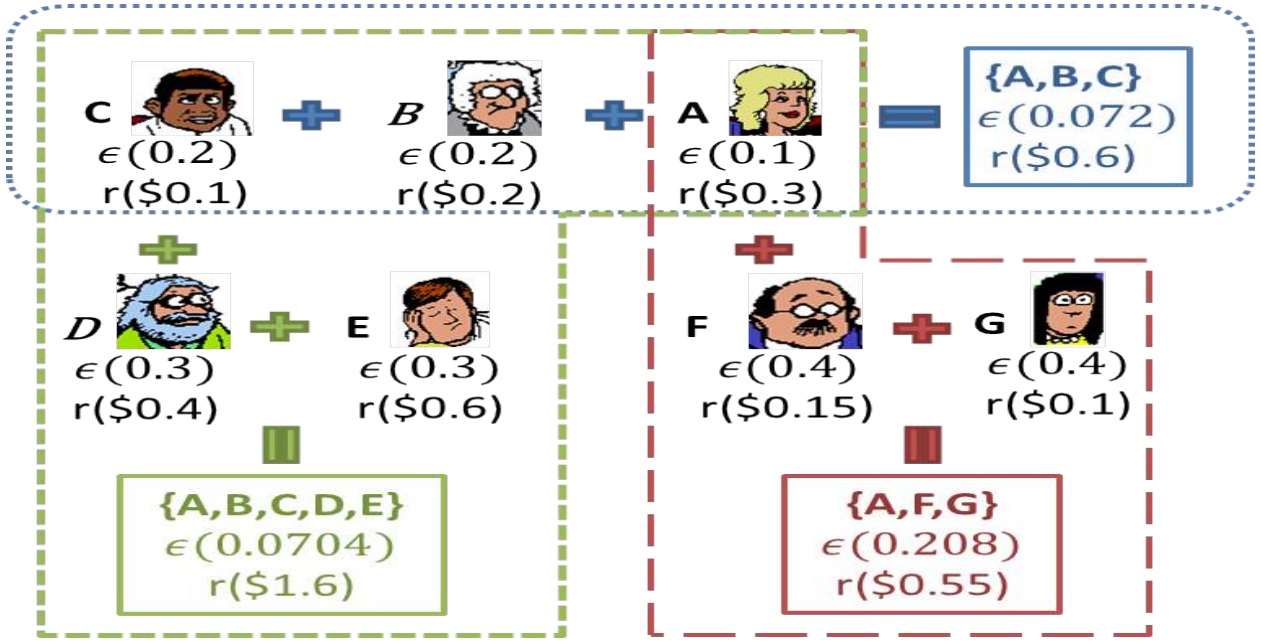
- If (A, B, C) are chosen(Majority Voting)
  - JER(A,B,C) = 0.1*0.2*0.2 + (1 - 0.1)*0.2*0.2 + 0.1* (1 - 0.2)*0.2 + 0.1*0.2*(1 - 0.2) = 0.072
  - Better than A(0.1), B(0.2) or C(0.2) individually

Motivation – Jury Selection Problem Running Case(4)



- What if we enroll more
  - JER(A,B,C,D,E) = 0.0704 < JER(A,B,C)
  - The more the better?

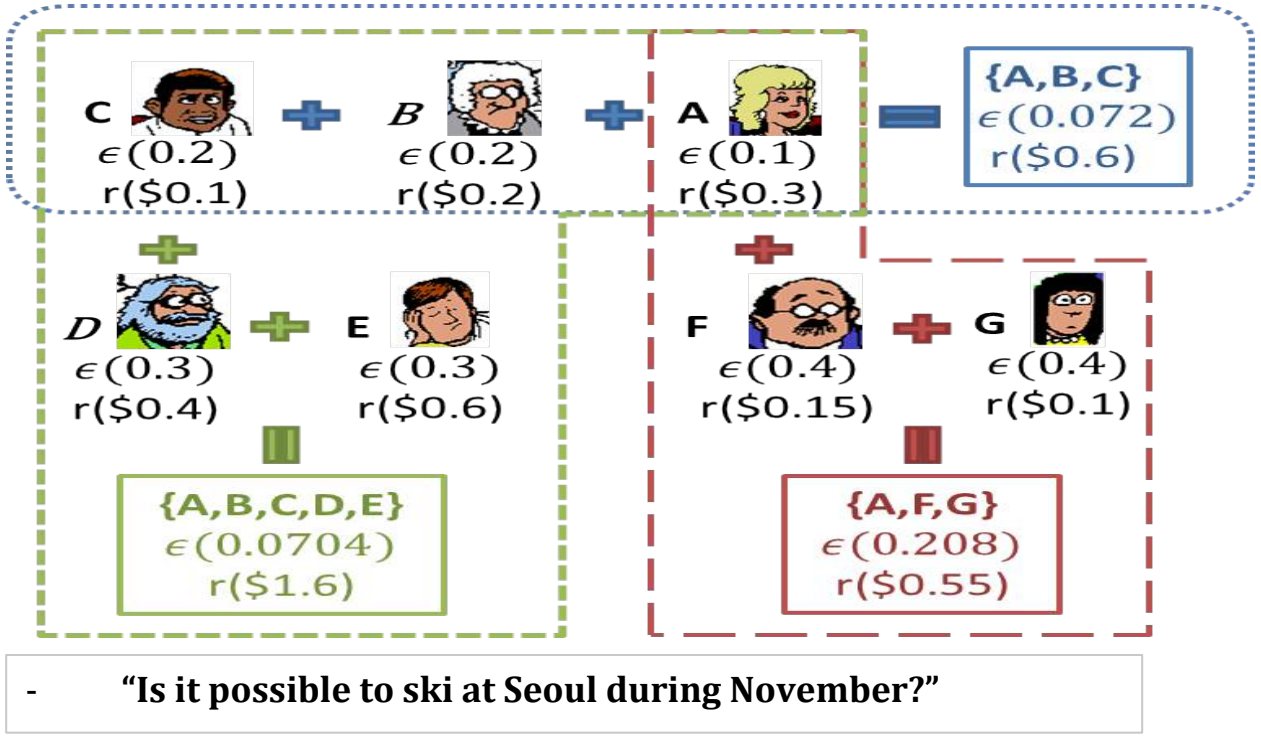
Motivation – Jury Selection Problem Running Case(5)



- "Is it possible to ski at Seoul during November?"

- What if we enroll even more?
  - JER(A,B,C,D,E,F,G) = 0.0805 > JER(A,B,C,D,E)
  - Hard to calculate JER

Motivation – Jury Selection Problem Running Case(6)



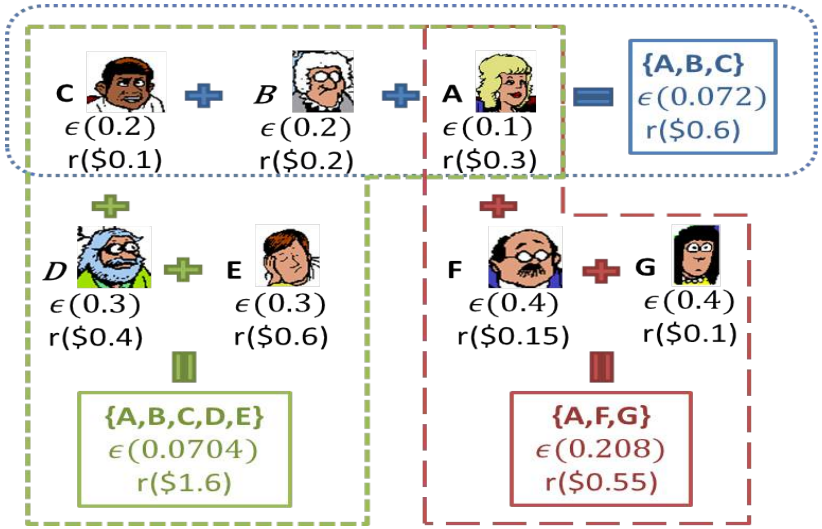
- So just pick up the best combination?
  - JER(A,B,C,D,E)=0.0704
  - R(A,B,C,D,E) = $1.6 > budget($1.0)

Motivation – Jury Selection Problem Running Case(7)

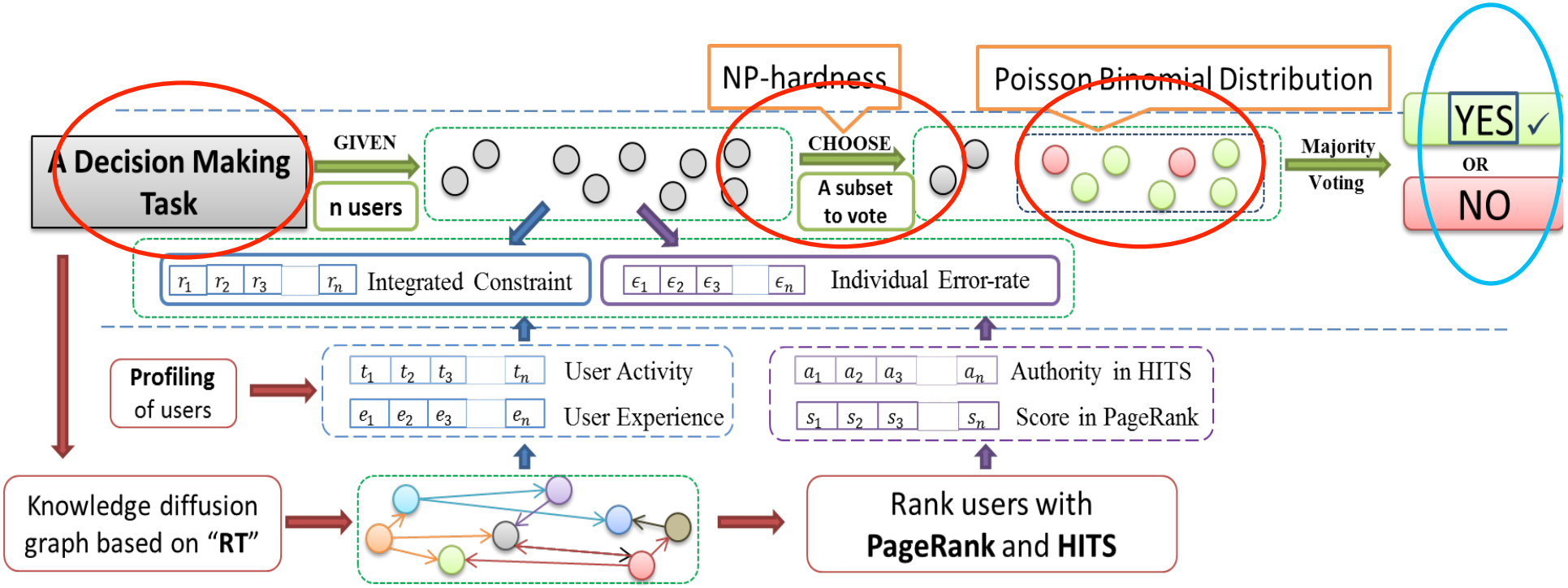
Crowd	Individual Error-rate	Jury Error-rate
C	0.2	0.2
A	0.1	0.1
C,D,E	0.2,0.2,0.3	0.174
A,B,C	0.1,0.2,0.2	0.072
A,B,C,D,E	0.1,0.2,0.2,0.3,0.3	0.0703
A,B,C,D,E,F,G	0.1,0.2,0.2,0.3,0.3,0.4,0.4	0.0805

Worker selection for maximize the quality of a particular type of product:

the reliability of voting.



Framework


A Decision Making Task
GIVEN
n users
NP-hardness
CHOOSE
A subset to vote
Poisson Binomial Distribution
Majority Voting
YES
OR
NO
Integrated Constraint
Individual Error-rate
Profiling of users
User Activity
User Experience
Authority in HITS
Score in PageRank
Knowledge diffusion graph based on "RT"
Rank users with PageRank and HITS

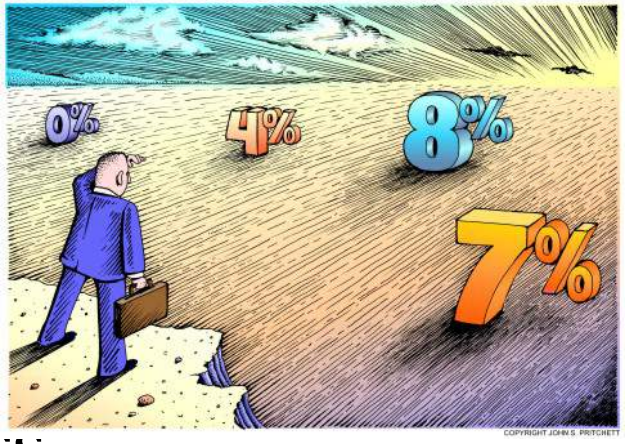
WiseMarket

- **Any structured method to manage the crowds?**

- **A Market**

Market

- Humans are investors
  - They have (partial)information
  - They invest to maximize incom



- A market consists of investors
  - Some of them win
  - Some of them lose

- A market can
  - Make Decisions/Show Preference
  - Based on Majority Voting

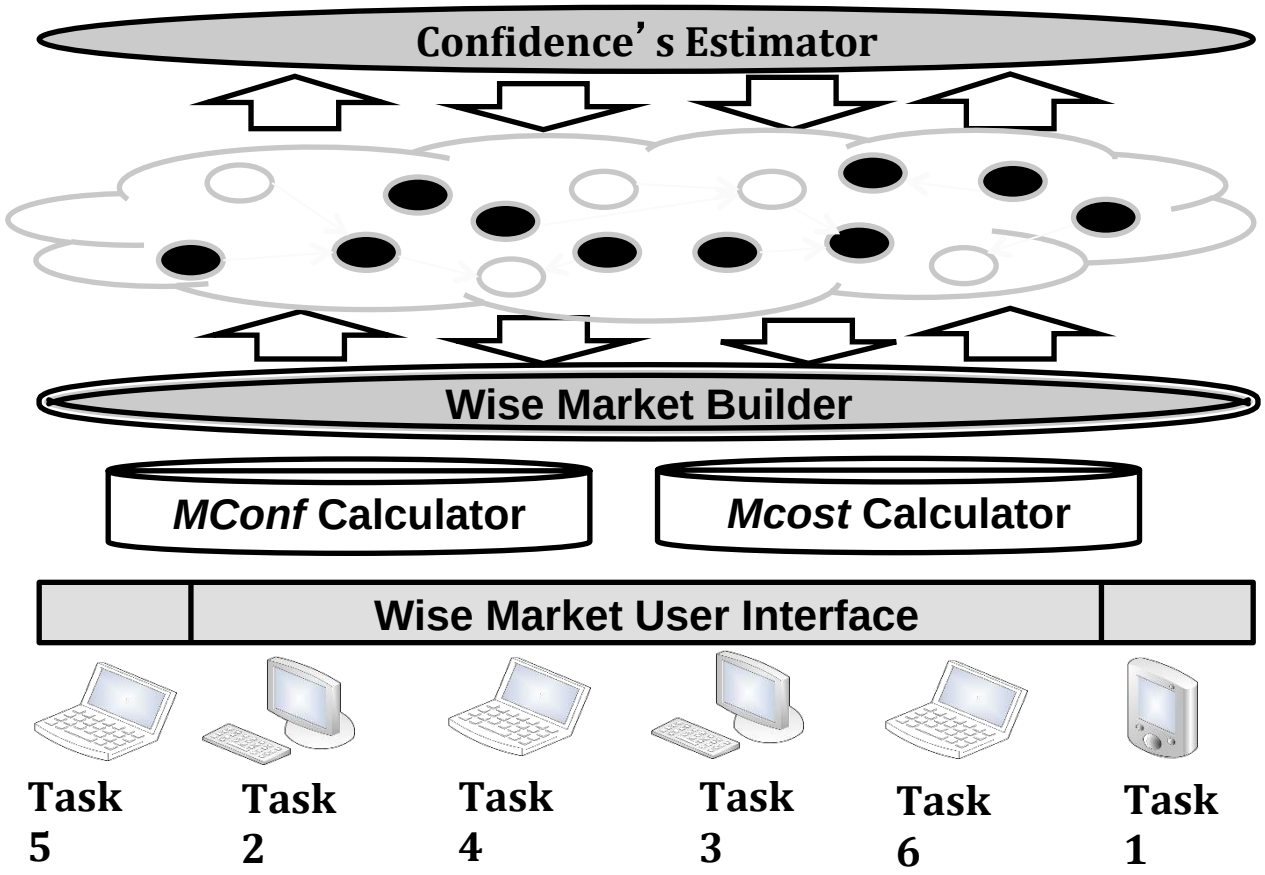
WiseMarket

Only winning investors get rewards

Why WiseMarket?

- Worriers in crowdsourcing, human computation services
  - Low Answers Quality
  - Spam Workers
  - Otiose Expenditure
- Drawbacks in survey samplings, online review aggregation
  - Vulnerable Quality Guarantee
  - Uncontrolled Demographic
- So How Does it Run?

How Does it Run?



Choose the best investors to build a market

Problem Definition

- Investors

DEFINITION 1 (INVESTOR CONFIDENCE). *For each investor ι_i, the* Investor Confidence c_i *is the probability that ι_i chooses the same option as the ground truth. Respectively, given a ground truth G, the confidence*

$$\begin{aligned} c_i &= \Pr\{\iota_i \textit{chooses correctly}\} \\ &= \Pr\{G=0\} \cdot \Pr\{v_i = 0 | G = 0\} \\ &\quad + \Pr\{G=1\} \cdot \Pr\{v_i = 1 | G = 1\} \\ &= \Pr\{v_i = G | G\} \end{aligned}$$

- v_i is the actual invest choice of the investor
- The two options are assumed to have equal prior preference

Problem Definition

- Wise Market

DEFINITION 2 (*Wise Market*). *A* Wise Market *is a set of investors* $WM_n = \{\iota_1, \iota_2, \ldots, \iota_n\} \subseteq I$ *with size* n*, where each* ι_i *is associated with an individual confidence* c_i *and actual voting* v_i*.*

- Market Opinion

DEFINITION 3 (MARKET OPINION). *Given a* Wise Market WM*, the Market Opinion* $OP(WM_n)$ *is the aggregated result according to the following equation:*

$$OP(WM_n) = \begin{cases} 1 & if \sum v_i \geq \lceil \frac{n}{2} \rceil \\ 0 & if \sum v_i \leq \lfloor \frac{n}{2} \rfloor \end{cases}$$

Problem Definition

- Market Confidence

DEFINITION 4 (MARKET CONFIDENCE). *The Market Confidence MC is defined as the probability that the Market Opinion is the same as ground truth G:*

$$MC(WM_n) = \Pr(OP(WM_n) = G|G)$$
$$= \Pr(|C| \geq \lceil \frac{n}{2} \rceil) = \Pr(|C| \geq \frac{n+1}{2})$$
$$= \sum_{k=\lceil \frac{n}{2} \rceil}^{n} \sum_{A \in F_k} \prod_{i \in A} c_i \prod_{j \in A^c} (1 - c_j)$$

- $F_k = \{A | |A| = k, A \subseteq WM_n\}$ is all the subsets of WM_n with size k
- A^c is the complementary set of A

Problem Definition

- Market Cost

DEFINITION 5 (MARKET COST). *Given a* Wise Market WM_n, *the Market Cost* $Cost(WM_n)$ *is defined as the size of the* Winning Set*:*

$$Cost(WM_n) = |W| = \left|\{\iota_i | \iota_i \in WM_n \text{ s.t. } v_i = OP(WM_n)\}\right|$$

- Expected Market Cost

$$\begin{aligned} & E[Cost(WM_n)] \\ &= \sum_{k=\lceil \frac{n}{2} \rceil}^{n} k \cdot \Pr(|W| = k) \\ &= \sum_{k=\lceil \frac{n}{2} \rceil}^{n} k \cdot [\sum_{A \in F_k} \prod_{i \in A} c_i \prod_{j \in A^c} (1 - c_j) \\ &\qquad + \sum_{A \in F_k} \prod_{i \in A} (1 - c_i) \prod_{j \in A^c} c_j] \end{aligned}$$

Problem Definition

- Effective Market Problem

DEFINITION 6 (EFFECTIVE MARKET PROBLEM). *Given a set of investors* $I = \{\iota_1, \ldots, \iota_N\}$ *with size* N, *a Market Confidence threshold* θ, *the Effective Market Problem(*EMP*) is to find a subset of all investors* $WM_n \subseteq I$, *so that:*

$$\begin{aligned} \text{minimize} \quad & E[Cost(WM_n)] \\ \text{subject to} \quad & MC(WM_n) \geq \theta \end{aligned}$$

- **A market *BUILDER* for tasks holders**

Market Building

- Overall Algorithm

Algorithm 6: Effective Market Algorithm (EMA)

Input: A set of candidate investors $I = \{\iota_1, \iota_2, \ldots, \iota_N\}$, an market confidence threshold θ

Output: *the effective market EM*

```
1  if (Max(c_i) ≥ θ) then
2      return EM ← c ∈ WM_n s.t. minimized{E[Cost(c)]} and
       MC(c) ≥ θ;
3  else
4      for i ← 1 to n − 1 do
5          if EM ≤ [AEC(min(s_{i+2}), θ)] then
6              return EM;
7          M_{i+2} ← RankMerge(M_i);
8          for s_j ∈ M_{i+2} do
9              if MC(s_j) ≥ θ && AEC(s_j, θ) ≤ EM then
10                 EM ← AEC(s_j, 0);
11             else
12                 continue;
```

COPE-Motivation

- Q: "What's your opinion about the game between Brazil and Germany tonight?"
- C1: "I vote for Germany, it will **definitely** win."
- C2: "I also vote for Germany. There's **no doubt**, since T. Silva and Neymar cannot play."
- C3: "There is still a **slight hope** that Brazil will win. I vote for Brazil."
- C4: "I know nothing about football. I'll **give it a shot** on Brazil."
- Judge: "2 v.s. 2. The crowds don't have an opinion."

Motivation

We need more than simple Binary Votes to capture the true opinion from the crowds.

From Labor to Trader: Motivation

- Opinion Elicitation
  - Opinion: *numerical statements expressing individual's degrees of belief about certain events*
  - Normally expressed as distribution

- Applications
  - Probabilistic Risk Analysis
    - Event Tree for industrial risk analysis
  - Causality Determination
    - PGM structure and probability

From Labor to Trader: Motivation

- Industrial Example

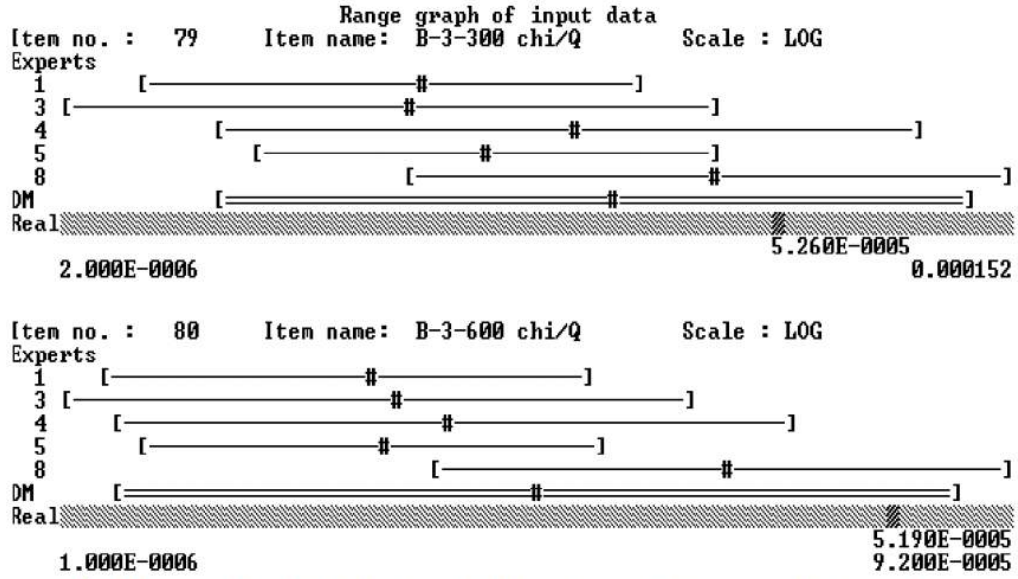


Figure 1: Example of Opinion Elicitation of five participants over two variables(NRC-EU accident uncertainty analysis [4])

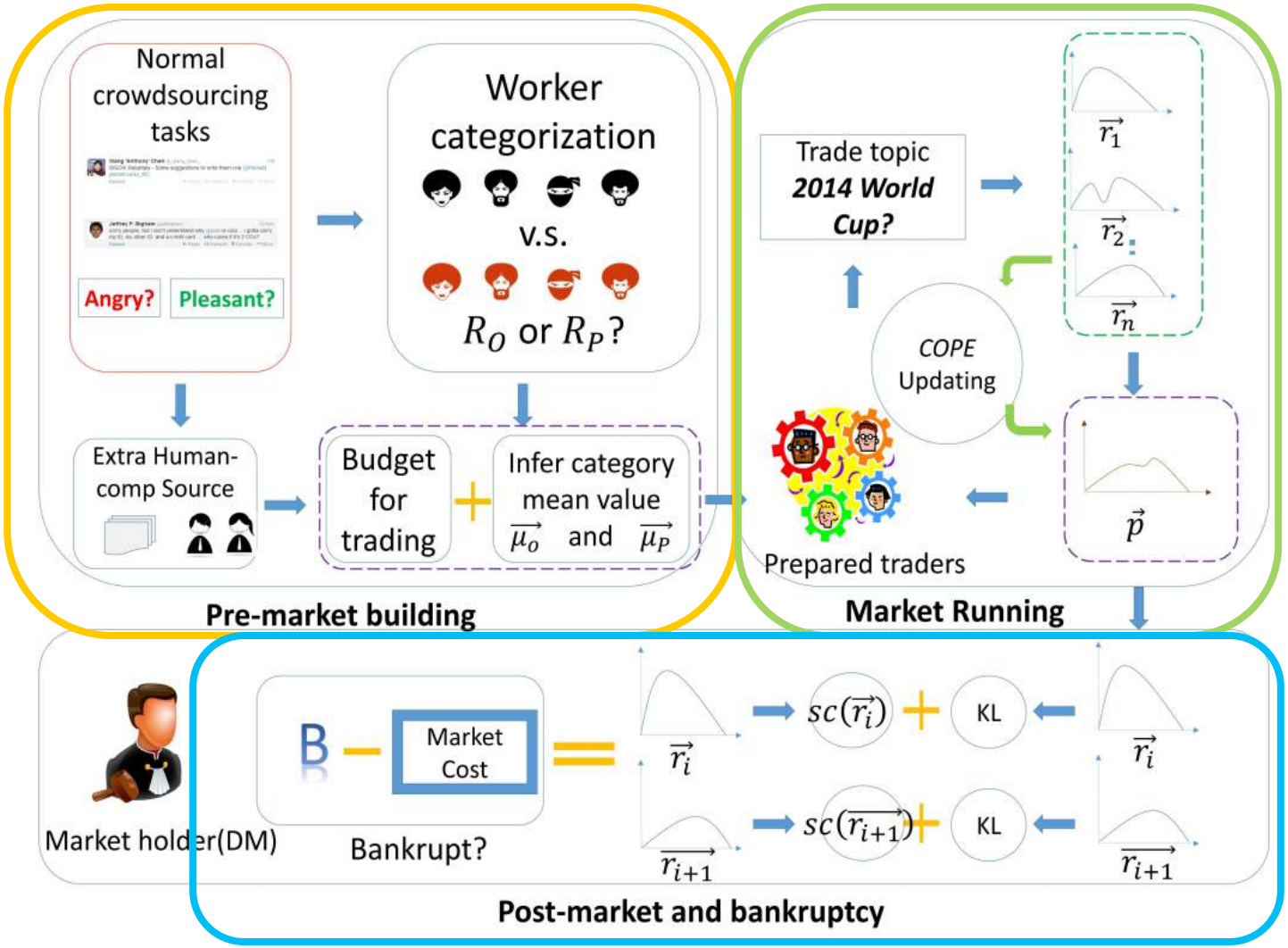
- Specifying (uniform) variable distribution over a range
- Multiple workers are involved to express their

Solution

- We propose COPE to tackle the challenges

- Crowd-powered OPinion Elicitation
  - General crowd workforce from any labor markets
  - Form an invest market situation
  - Payments are connected to their contribution

COPE Framework



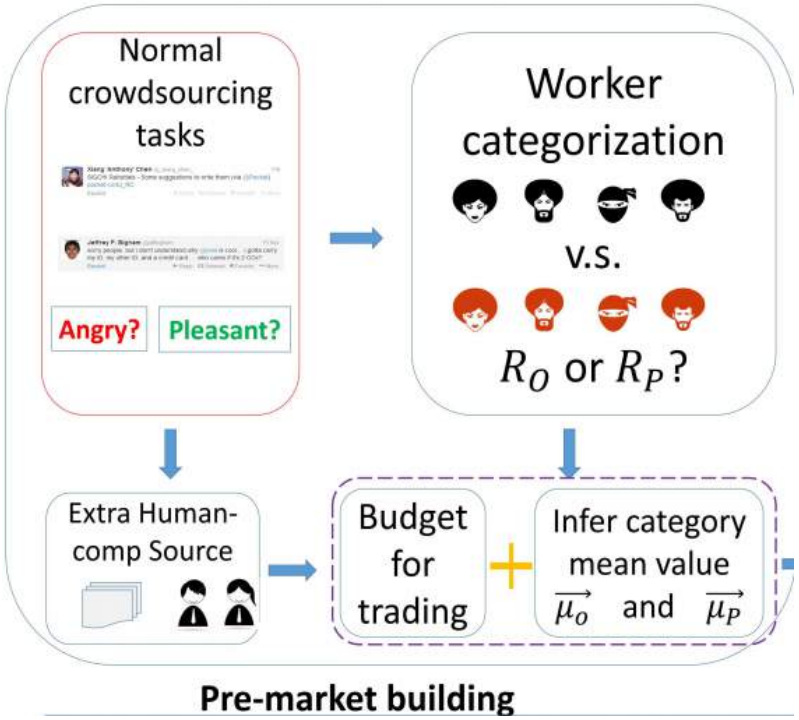
COPE – The Design

- Trader
  - A trader will present a report that maximize his/her payoff according to a payoff rule
  - Traders are assumed as Risk-neutral
    - i.e. expected payoff oriented
    - Risk aversion enhances sincerity but introduces bias

COPE – The Design

- Pre-market Building
  - Generate Seed Capital
    - Promised Salaries as initial funds
  - Tendency Evaluation
    - Optimistic
    - Pessimistic
    - Group Mean
      - For bias adjustment during Bayesian updating

$$\vec{\mu_o} = \frac{\sum_i \vec{r_i^o}}{|R_o|} \qquad \vec{\mu_p} = \frac{\sum_i \vec{r_i^p}}{|R_p|}$$

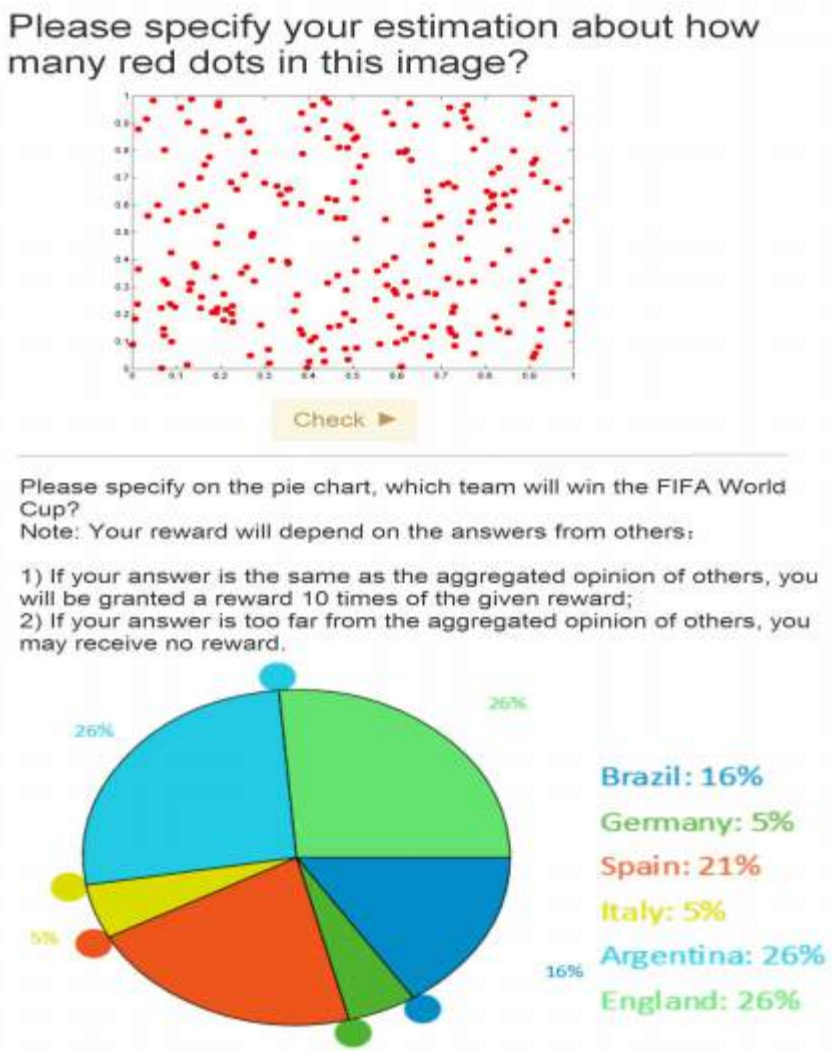


COPE – The Design

- Bayesian Updating Scheme
  - The design of COPE indicates the existence of a *latent decision maker*, as in the case of probabilistic risk analysis
  - Bayesian Updating is the best practice for such scenario*
  - Two principles for a normative Bayesian Updating
    - Unanimity: info-less report don't update global distribution
    - Compromise: global distribution is between the two extremes

$$p^* = \Pr(\vec{p}|\vec{r}) \propto \frac{\Pr(\vec{p})L(\vec{r}|\vec{p})}{\Pr(\vec{r})}$$

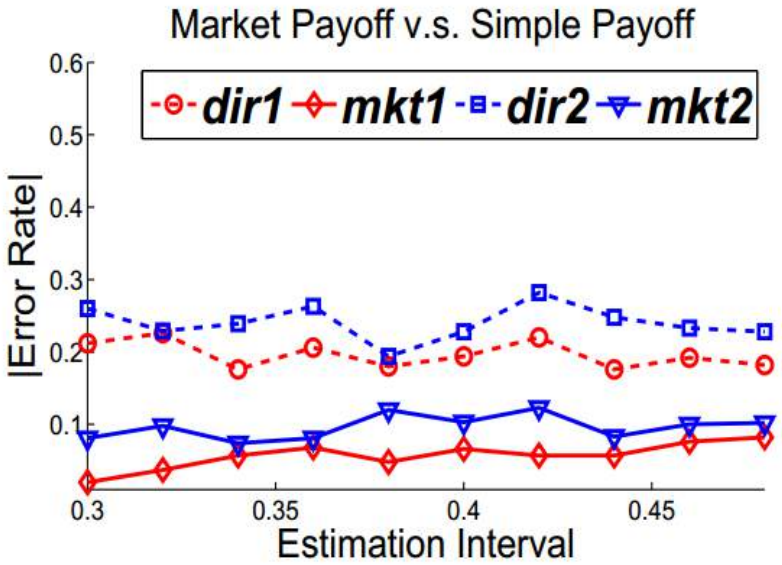
COPE – The Implementation

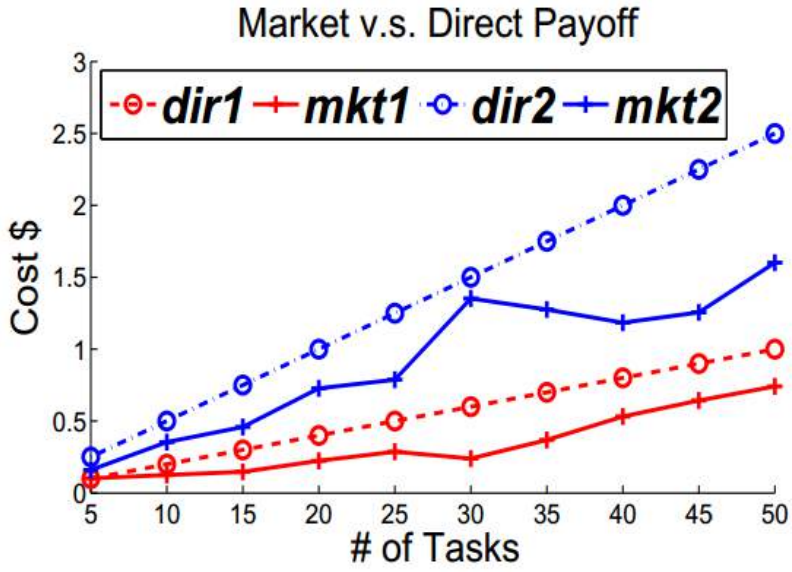


- Premarket Tasks
- Opinion Elicitation
  - Dynamic Chart
  - Kill *probability-phobia*
    - Unwilling or uncomfortable to give numerical probability
  - Workers are informed the payoff method
- Payoff Dispatch
  - Special "*payoff tasks*" are generated with workerID

COPE – The Evaluation

- Merits of Market Mechanism





- task: estimate man's age according to a photo
- *dir* means Direct Pay, *mkt* means market-based payoff

Summary

- Quality Control is a difficult problem

  Because we are dealing with human and we are…
- Possible Solutions:
  - Task Design
  - Reward Design
  - Aggregation Design
  - …..

Reference

- **[Cao-12]**, "Whom to Ask? Jury Selection for Decision Making Tasks on Microblog Services", Chen et al, VLDB 2012.
- **[Zhang-13]** "Reducing Uncertainty of Schema Matching via Crowdsourcing", Zhang et al, VLDB 2013
- **[Cao-13]**, "WiseMarket: A New Paradigm for Managing Wisdom of Online Social Users", Cao et al, KDD 2013
- **[Tong-14]**, "CrowdCleaner: Data Cleaning For Multi-Version Data on the Web via Crowdsourcing", Tong et al, ICDE 2014 (demo)
- **[Cao-14]** "From Labor to Trader: Opinion Elicitation via Online Crowds as a Market", Cao et al, KDD 2014
- **[Brabham-13]** *Crowdsourcing*, Daren Brabham, 2013
- **[Franklin-SIGMOD11]** *CrowdDB: answering queries with crowdsourcing*, Michael J. Franklin et al, SIGMOD 2011
- **[Howe-08]** *Crowdsourcing*, Jeff Howe, 2008
- **[LawAhn-11]** *Human Computation*, Edith Law and Luis von Ahn, 2011
- **[Li-HotDB12]** *Crowdsourcing: Challenges and Opportunities*, Guoliang Li, HotDB 2012
- **[Marcus-VLDB11]** *Human-powered Sorts and Joins*, Adam Marcus et al., VLDB 2011
- **[Miller-13]** *Crowd Computing and Human Computation Algorithms*, Rob Miller, 2013
- **[Shirky-08]** *Here Comes Everybody*, Clay Shirky, 2008