Crowdsourced Data Management: Overview and Challenges

Guoliang Li	Yudian Zheng	Ju Fan	Jiannan Wang	Reynold Cheng
Tsinghua University	Hong Kong University	Renmin University	SFU	Hong Kong University

Outline

- Crowdsourcing Overview (30min)
  - Motivation (5min)
  - Workflow (15min)
  - Platforms (5min)
  - Difference from Other Tutorials (5min)
- Fundamental Techniques (100min)
  - Quality Control (60min)
  - Cost Control (20min)
  - Latency Control (20min)
- Crowdsourced Database Management (40min)
  - Crowdsourced Databases (20min)
  - Crowdsourced Optimizations (10min)
  - Crowdsourced Operators (10min)
- Challenges (10min)

Part 1

Part 2

Crowdsourcing： Motivation

- A new computation model
  - Coordinating the crowd (Internet workers) to do micro-tasks in order to solve computer-hard problems.
- Examples
  - Categorize the products and create product taxonomies from the user's standpoint.
  - An example question
    - Select the product category of Samsung S7
      - Phone
      - TV
      - Movie

Crowdsourcing：Applications

- Wikipedia
  - Collaborative knowledge
- reCAPTCHA
  - Digitalizing newspapers
- Foldit
  - fold the structures of selected proteins
- App Testing
  - Test apps

WIKIPEDIA
The Free Encyclopedia

Crowdsourcing: Popular Tasks

- **Sentiment Analysis**
  - Understand conversation: positive/negative

- **Search Relevance**
  - Return relevant results on the first search

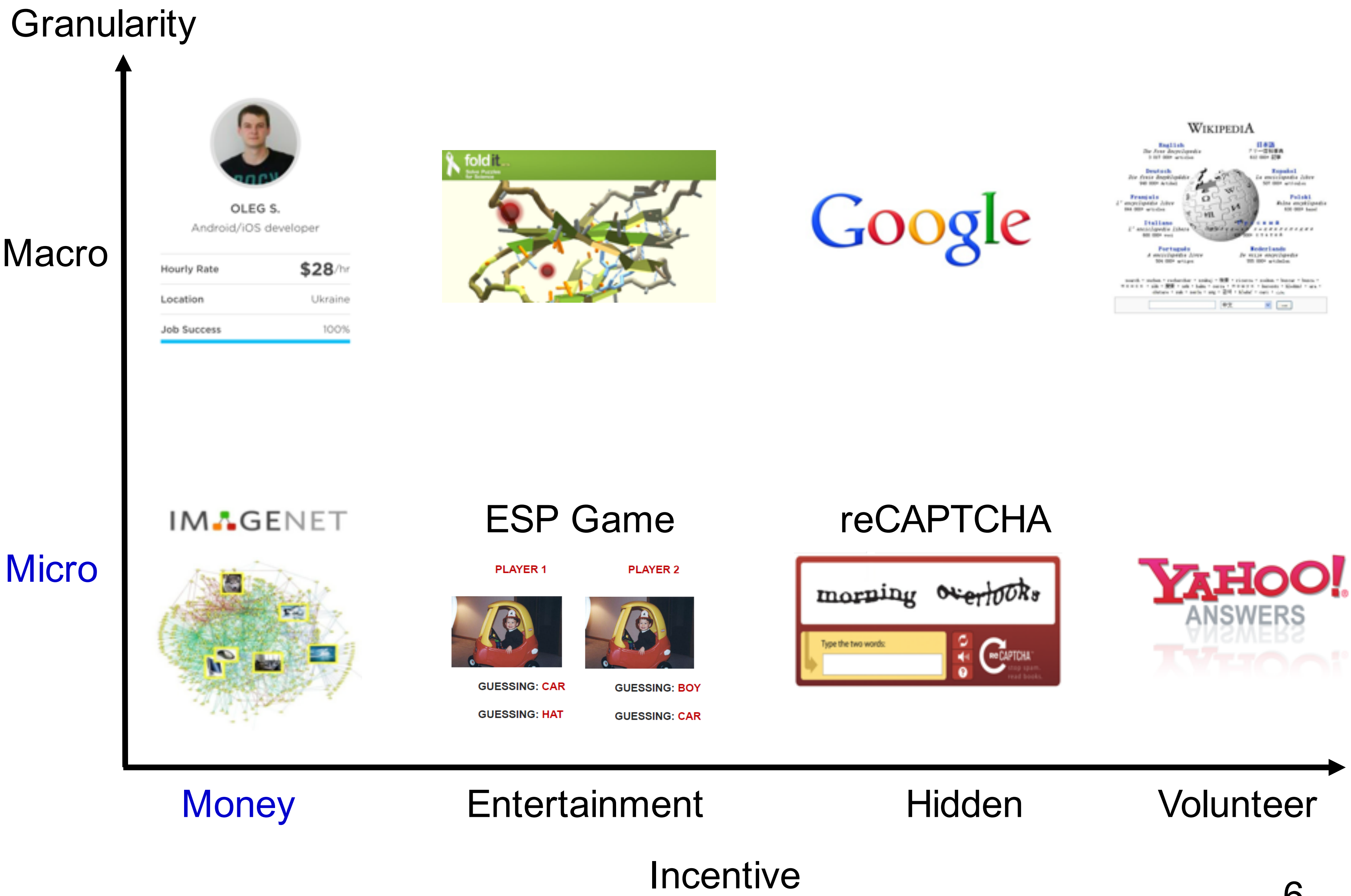
- **Content Moderation**
  - Keep the best, lose the worst

- **Data Collection**
  - Verify and enrich your business data

- **Data Categorization**
  - Organize your data

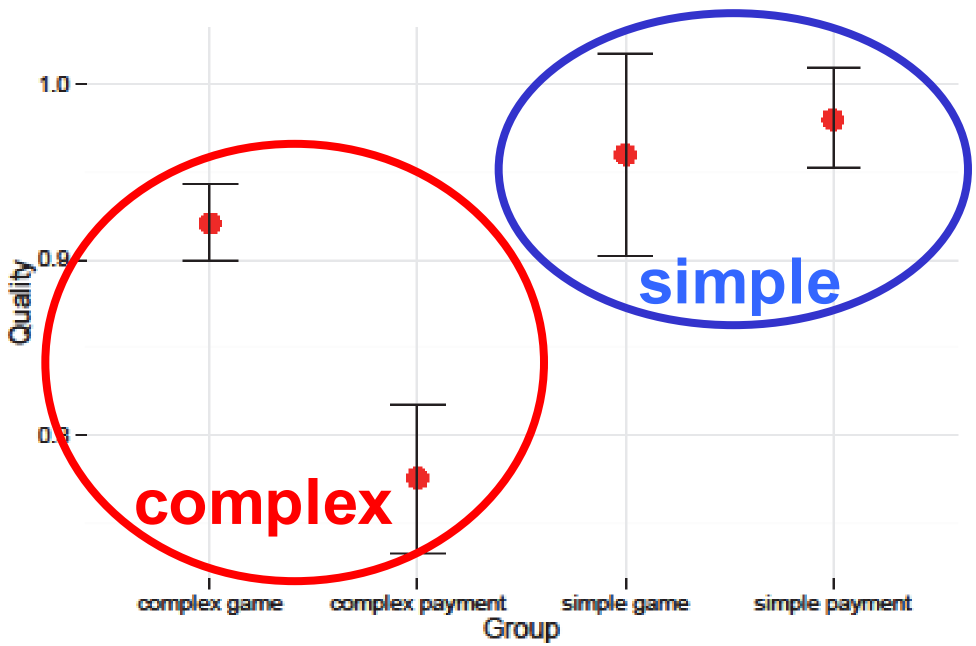
- **Transcription**
  - Turn images and audio into useful data

Crowdsourcing Space



Crowdsourcing Category

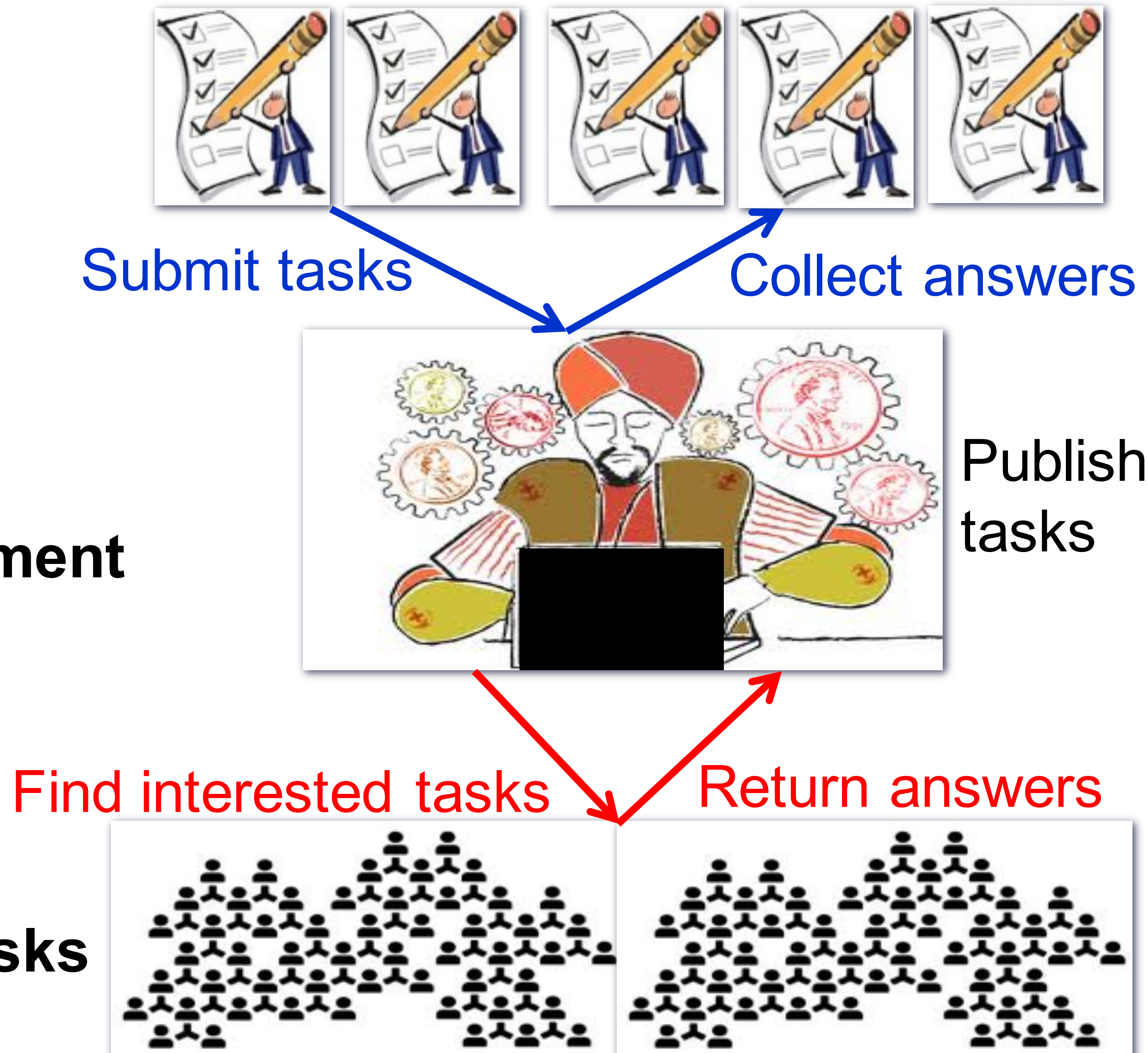
- Game vs Payment
  - Simple tasks
    - Both payment and game can achieve high quality
  - Complex tasks
    - Game has better quality



Quality is rather important!

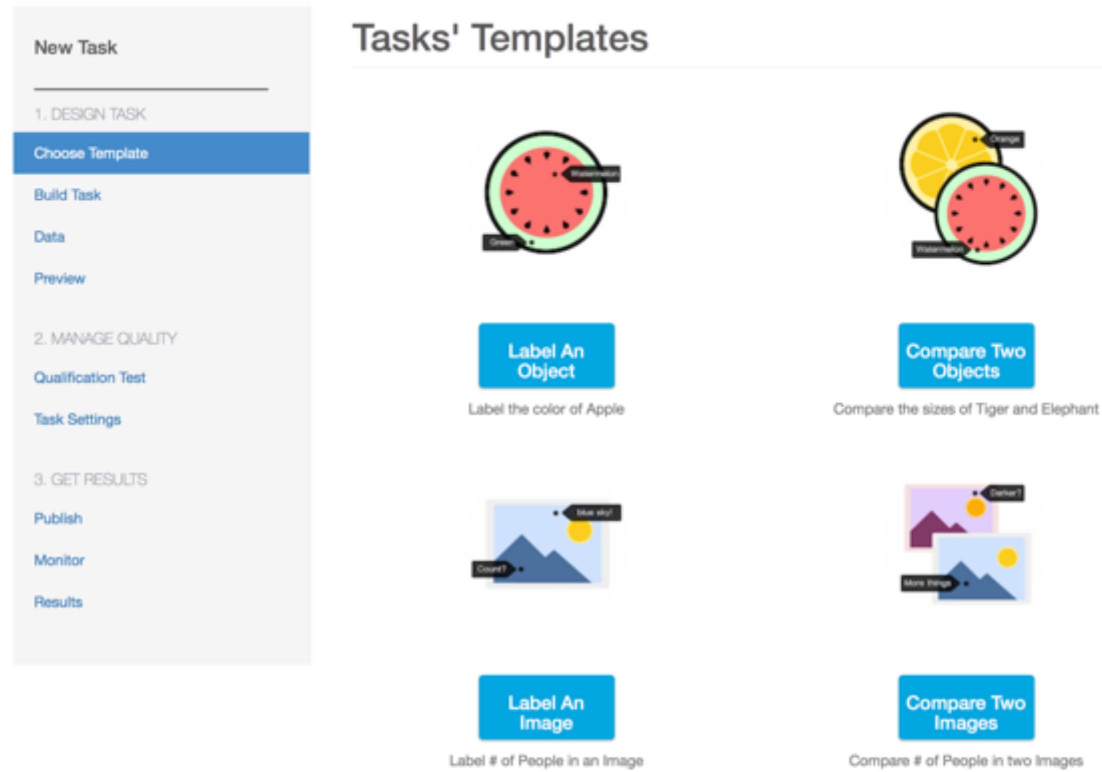
Crowdsourcing：Workflow

- **Requester**
  - **Submit Tasks**
- **Platforms**
  - **Task Management**
- **Workers**
  - **Worker on Tasks**



Crowdsourcing Requester：Workflow

- ○ **Design Tasks**
  - **Task Type**
  - **Design Strategies**
    - – UI, API, Coding
- ○ **Upload Data**
- ○ **Set Tasks**
  - **Price**
  - **Time**
  - **Quality**
- ○ **Publish Task**
  - **Pay**
  - **Monitor**



Crowdsourcing Requester：Task Type

○ Task Type

Please choose the brand of the phone

- Apple
- Samsung
- Blackberry
- Other

What are comment features?

- [ ] Same band
- [ ] Same color
- [x] Similar price
- [x] Same size

Please fill the attributes of the product

Brand	
Price	
Size	
Camera	

Please submit a picture of a phone with the same size as the left one.

Submit

Crowdsourcing Requester: Task Design

- **UI**

Choose the best category for the image

- Kitchen
- Bath
- Living
- Bed

- **API**

The Amazon Mechanical Turk API consists of web service operations for every task the service can perform. This section describes each operation in detail.

- AcceptQualificationRequest
- ApproveAssignment
- AssociateQualificationWithWorker
- CreateAdditionalAssignmentsForHIT
- CreateHIT

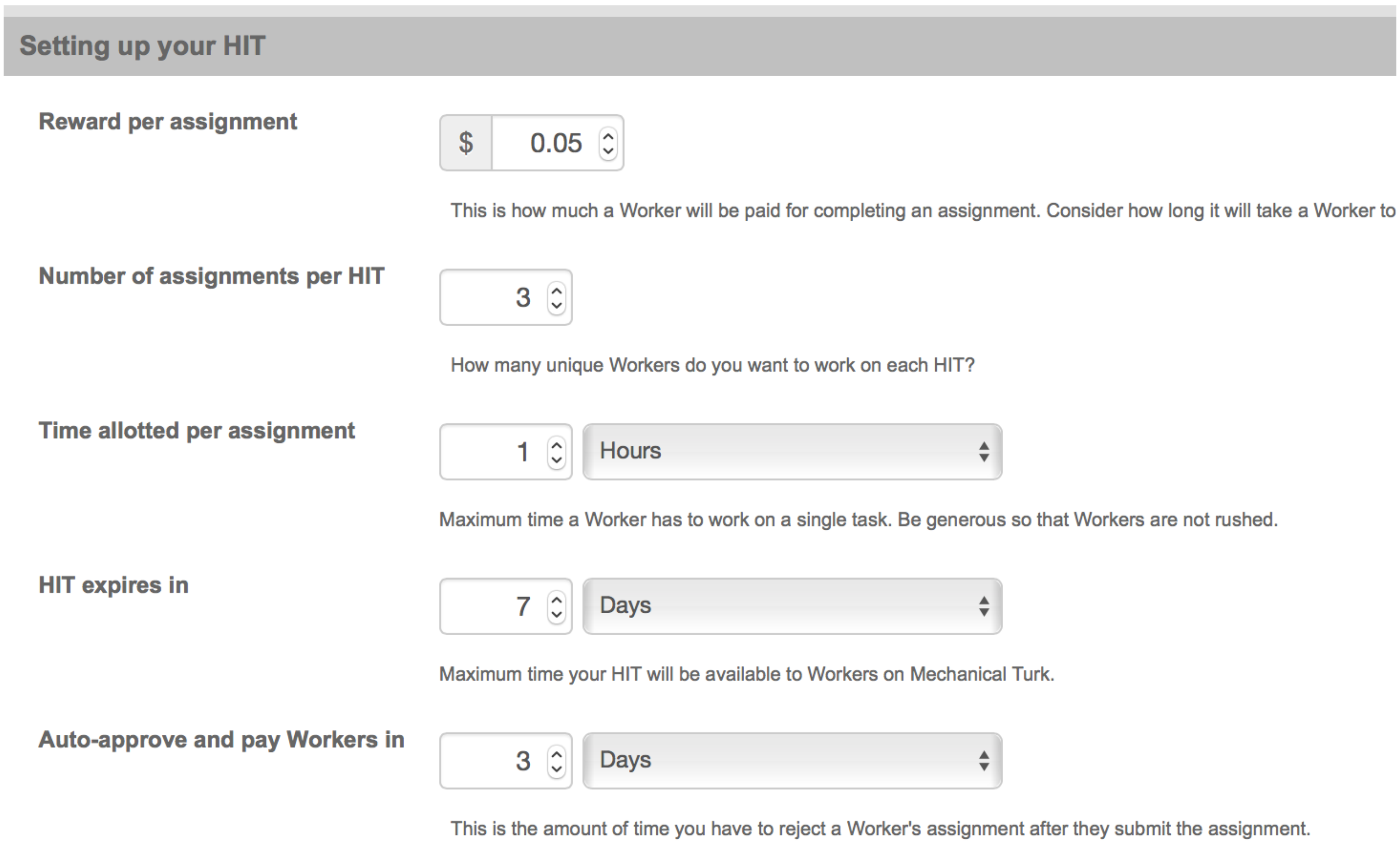
- **Coding**
  (Your own Server)
  innerhtml

```
# Create the HIT
response = client.create_hit(
    MaxAssignments = 10,
    LifetimeInSeconds = 600,
    AssignmentDurationInSeconds = 600,
    Reward ='0.20',
    Title = 'Answer a simple question',
    Keywords = 'question, answer, research',
    Description = 'Answer a simple question',
    Question = questionSample,
    QualificationRequirements = localRequirements
)

# The response included several fields that will be helpful later
hit_type_id = response['HIT']['HITTypeId']
hit_id = response['HIT']['HITId']
print "Your HIT has been created. You can see it at this link:"
print "https://workersandbox.mturk.com/mturk/preview?groupId={}".format(hit_type_id)
print "Your HIT ID is: {}".format(hit_id)
```

Crowdsourcing Requester: Task Setting

- HIT – A group of micro-tasks (e.g., 5)
- Price, Assignment, Time



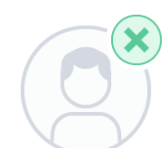
Crowdsourcing Requester: Task Setting

- **Quality Control**
  - **Qualification test - Quiz**

    Create some test questions to enable a quiz that workers must pass to work on your task.

  - **Hidden test - Training**

    Add some questions with ground truths in your task so workers who get them wrong will be eliminated.



  - **Worker selection**

    Ensure high-quality results by eliminating workers who repeatedly fail test questions in your task

Crowdsourcing Requester: Publish

- **Prepay**

  cost for workers + cost for platform +cost for test

Expected Cost:

Contributor judgments	$0.00
Cost buffer	$10.00
Transaction fee (20%)	$0.00
Due Now	**$10.00**
Available Funds	$16.01
Add Funds	

Reward per Assignment:		$0.05
	x	3
Estimated Total Reward:		$0.15
Estimated Fees to Mechanical Turk:	+	$0.03
Estimated Cost:		$0.18

- **Monitor**

0%	3	¥0
Finished Units	Workers per unit	Cost
5	10	5
All Units	Qualification Units	No of Hidden Units

Real-time Statistics

0	0
Finished Units	Workers

Crowdsourcing: Workers

- Task Selection
- Task Completion
- Workers are not free **Cost**
  - Make Money

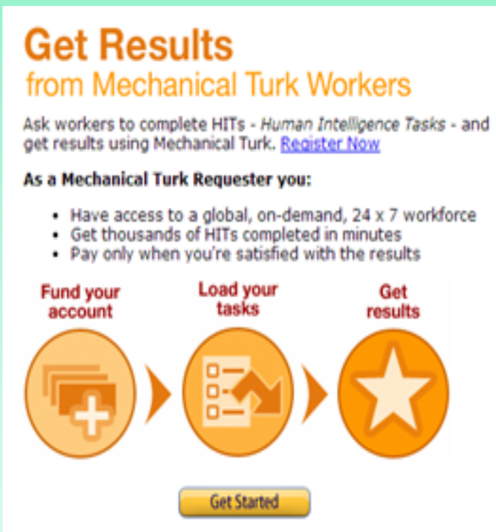
- Workers are not oracle **Quality**
  - Make errors
  - Malicious workers

- Workers are dynamic **Latency**
  - Hard to predict

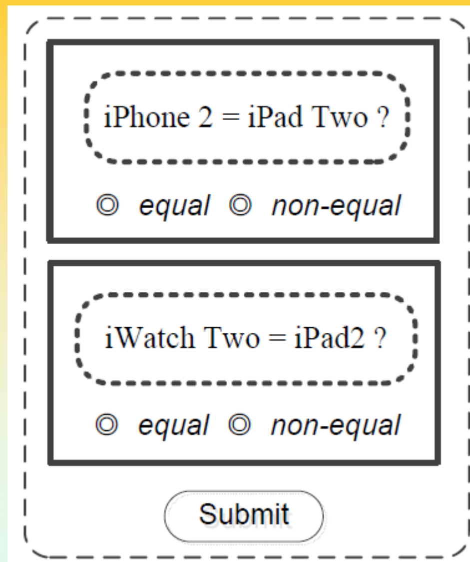
Crowdsourcing: Platforms

- Amazon Mechanical Turk (AMT)

□ Requesters



□ HIT (k tasks)



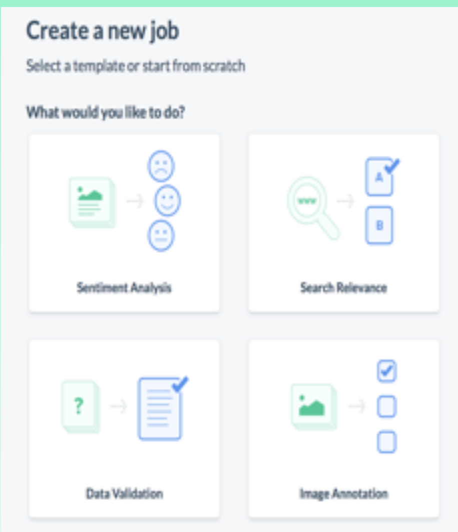
□ Workers

more than ***500,000 workers*** *from* ***190 countries***

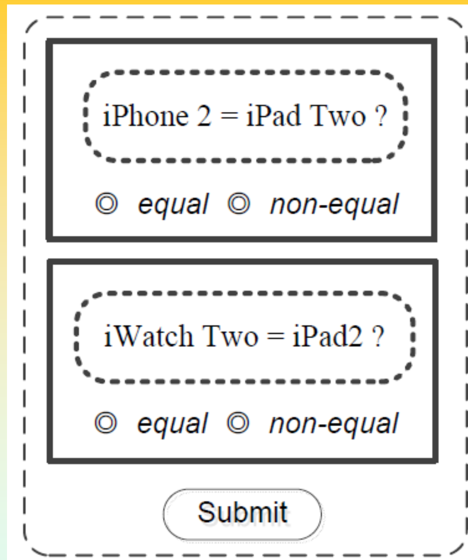
Crowdsourcing： Platforms

- CrowdFlower

□ Requesters



□ HIT (k tasks)

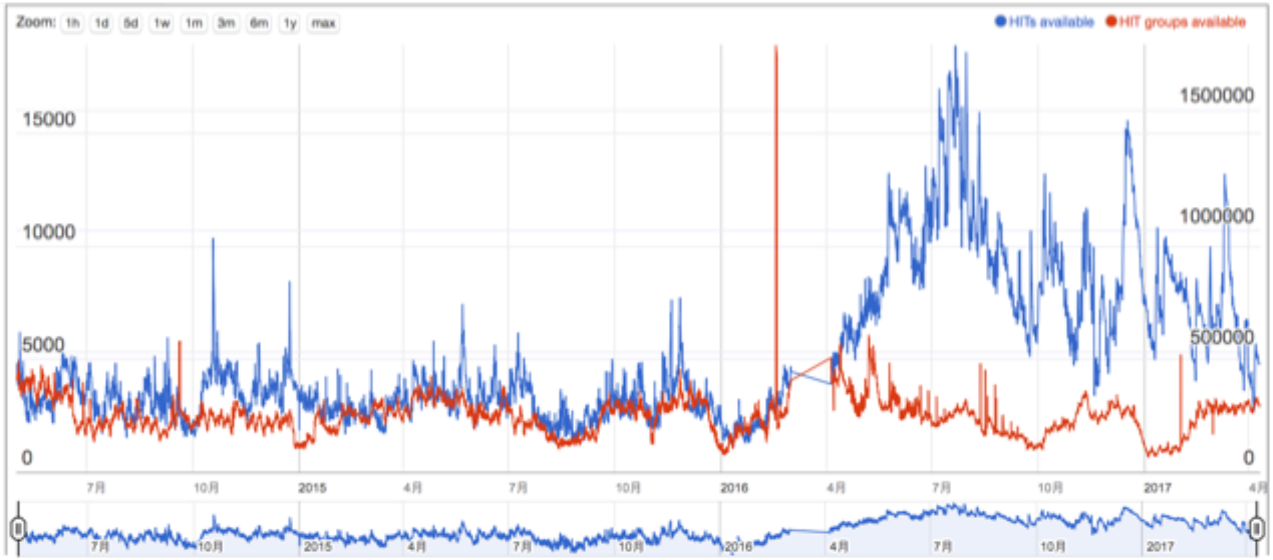


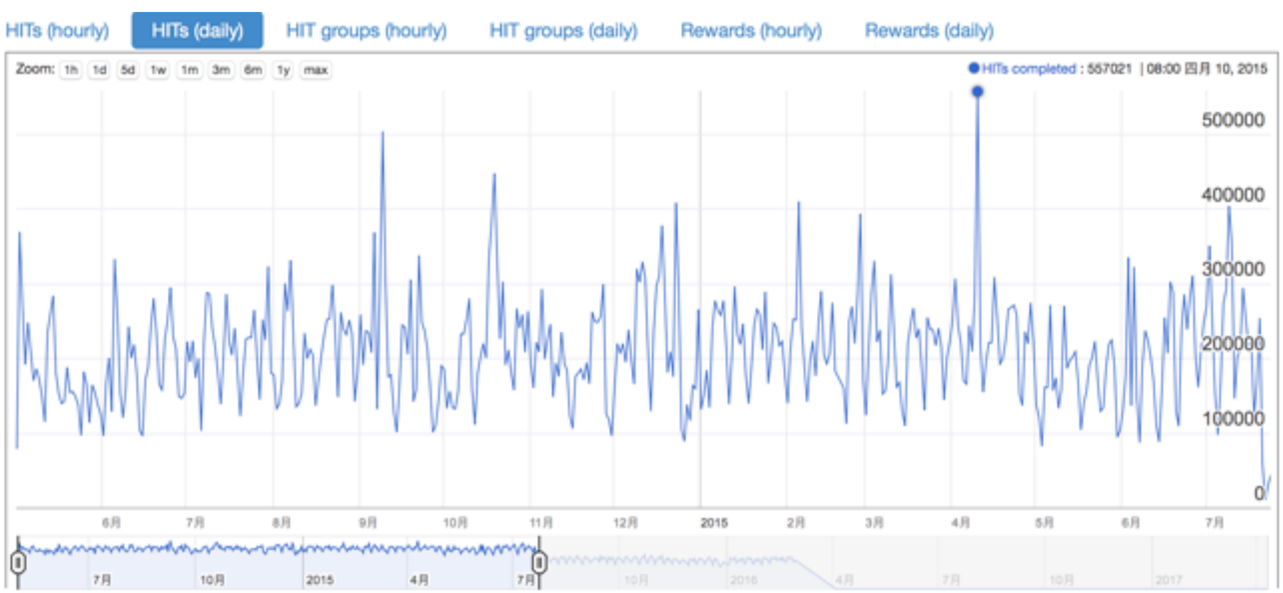
□ Workers

AMT vs CrowdFlower

	AMT	CrowdFlower
Task Design: UI	√	√
Task Design: API	√	√
Task Design: Coding	√	×
Quality: Qualification Test	√	√
Quality: Hidden Test	×	√
Quality: Worker Selection	√	√
Task Types	All Types	All Types

AMT Task Statistics

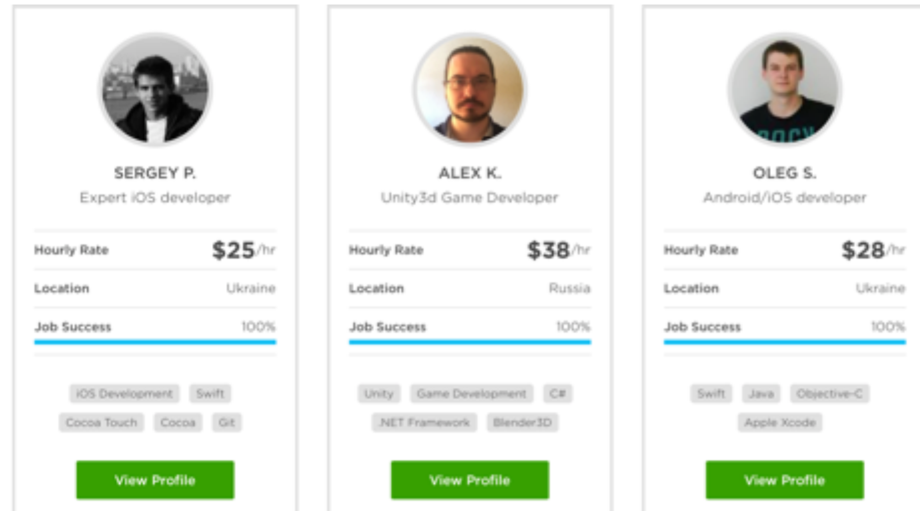




http://www.mturk-tracker.com

Other Crowdsourcing Platforms

- **Macrotask**
  - **Upwork**
    - https://www.upwork.com
  - **Zhubajie**
    - http://www.zbj.com



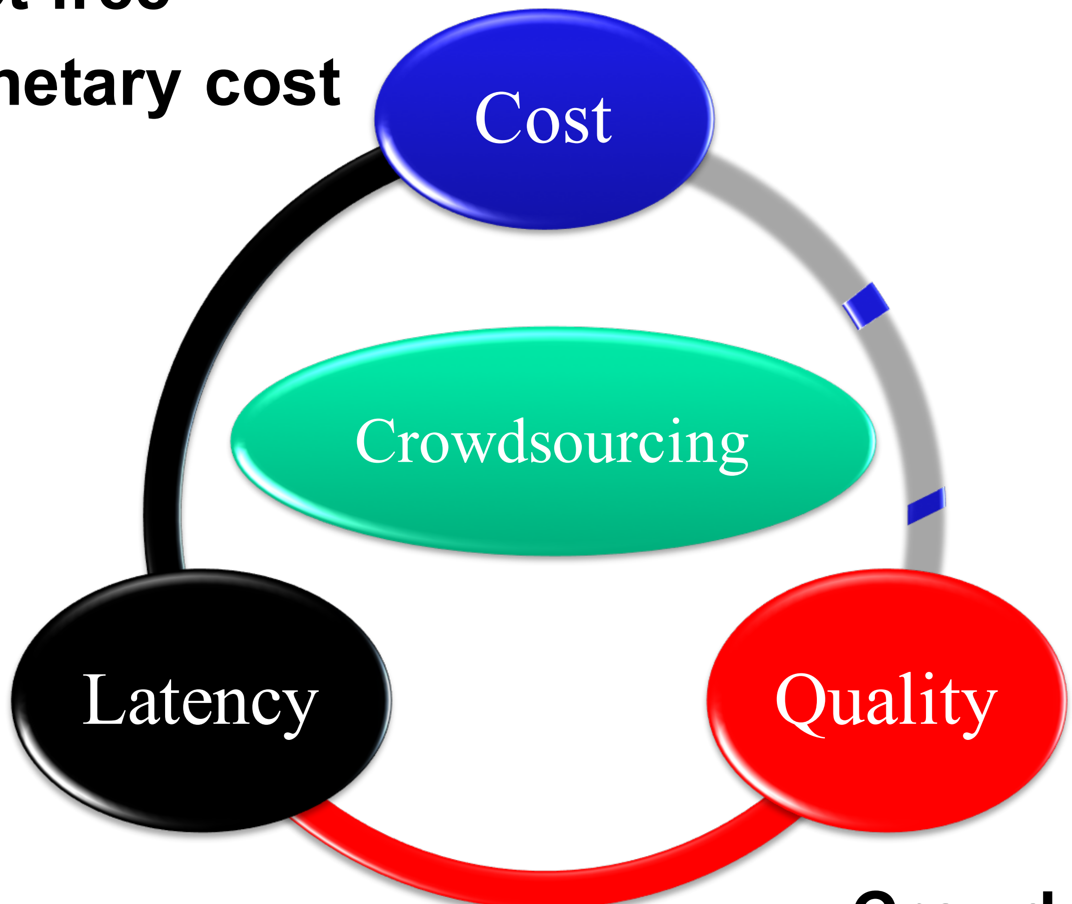
- **Microtask**
  - **ChinaCrowds (cover all features of AMT and CrowdFlower)**
    - http://www.chinacrowds.com

iOS

Android

Crowdsourcing：Challenges

- Crowd is not free
- Reduce monetary cost

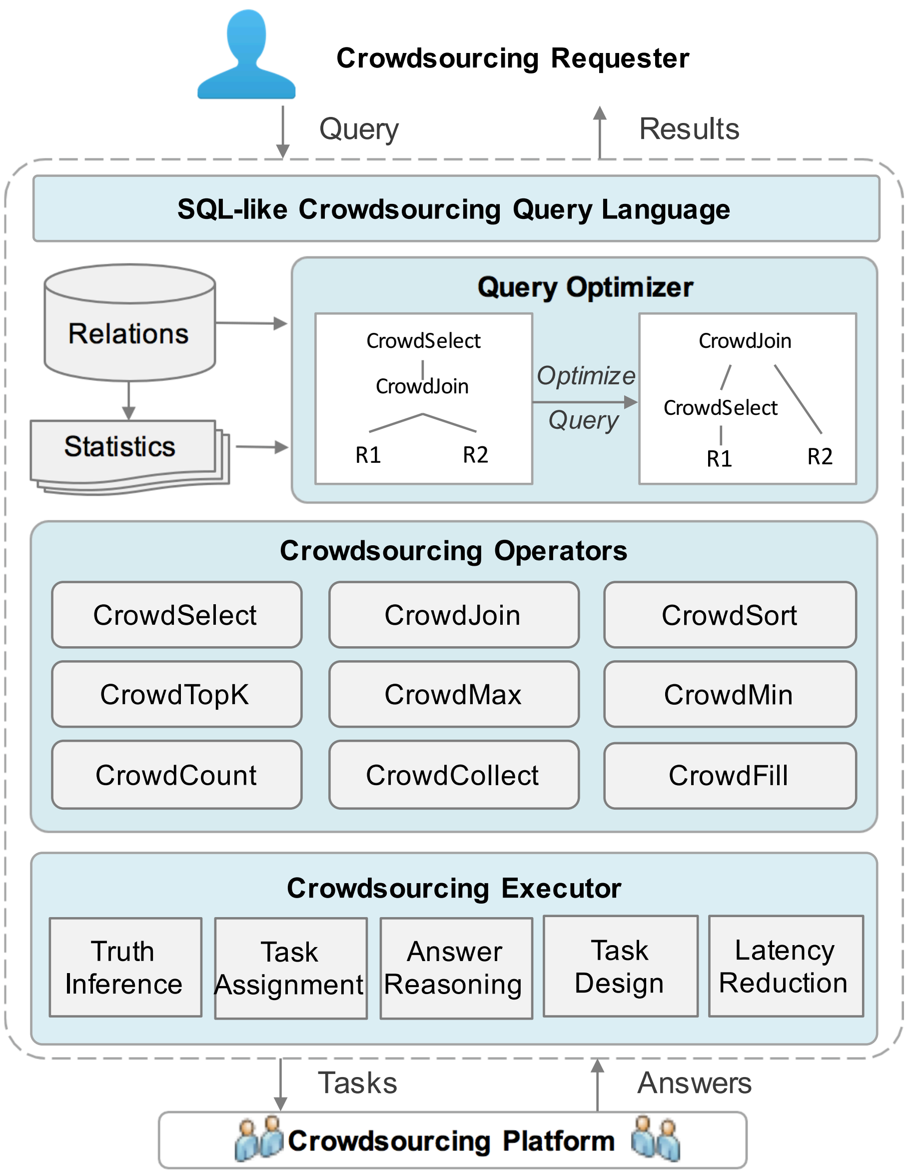


- Crowd is not real-time
- Reduce time

- Crowd may return incorrect answers
- Improve quality

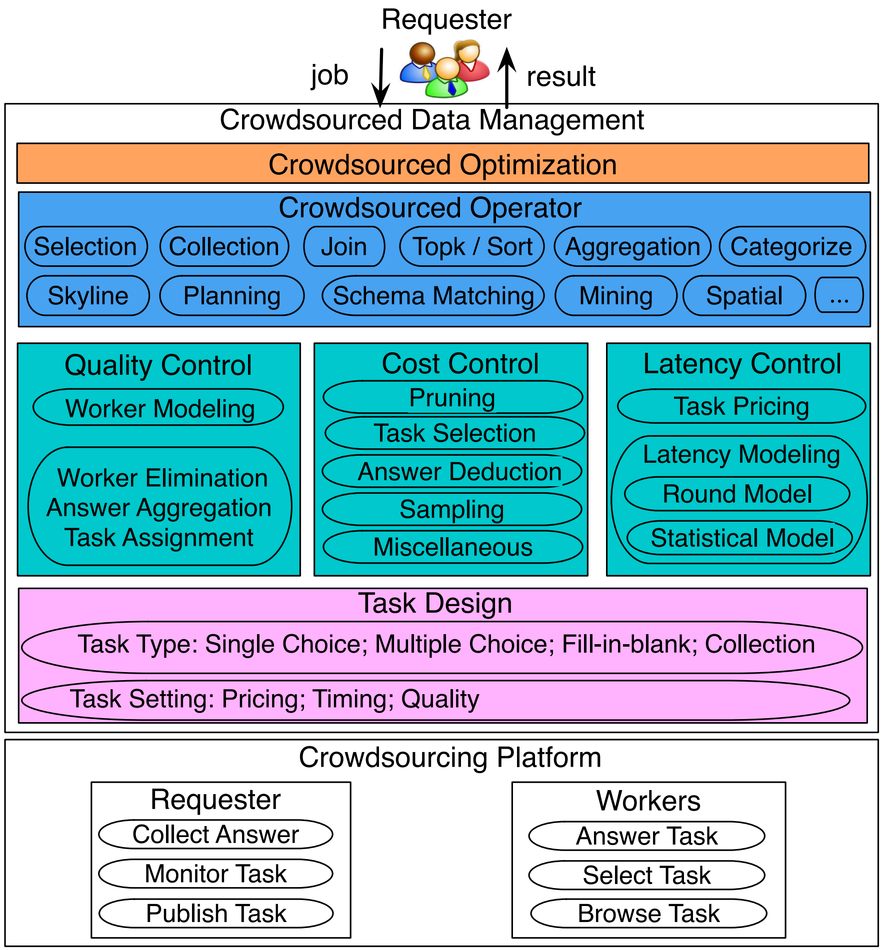
Crowdsourced Data Management

- **A crowd-powered database system**
  - Users require to write code to utilize crowdsourcing platforms
  - Encapsulates the complexities of interacting with the crowd
  - Make DB more powerful
- **Crowd-powered interface**
- **Crowd-powered Operators**
- **Crowdsourcing Optimization**

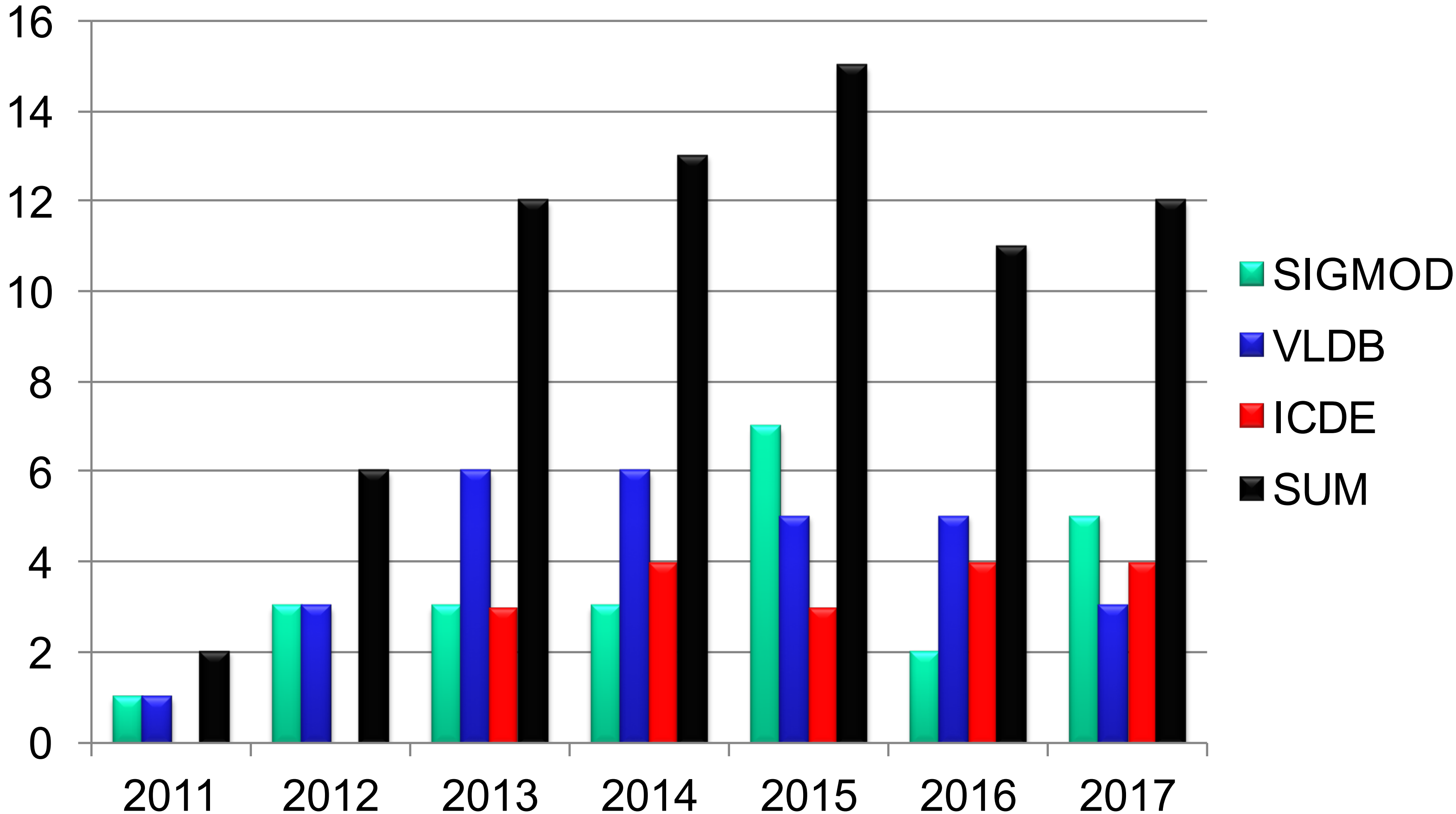


Tutorial Outline

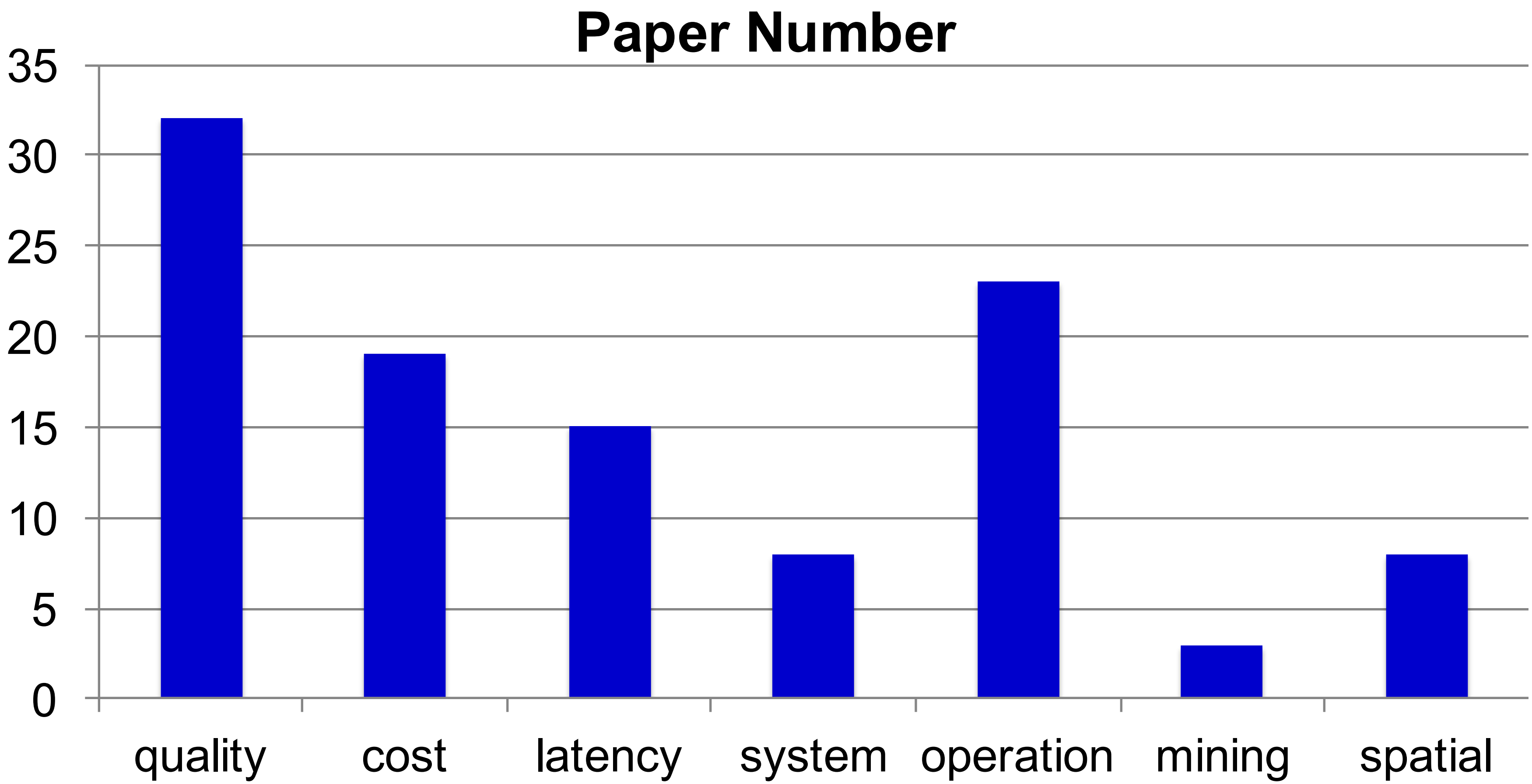
- Fundamental Optimization
  - Quality Control
  - Cost Control
  - Latency Control
- Crowd-powered Database
- Crowd-powered Operators
  - Selection/Join/Group
  - Topk/Sort
  - Collection/Fill
- Challenges



Existing Works



Existing Works



Differences with Existing Tutorials

- **VLDB'16**
  - Human factors involved in task assignment and completion.
- **VLDB'15**
  - Truth inference in quality control
- **ICDE'15**
  - Individual crowdsourcing operators, crowdsourced data mining and social applications
- **VLDB'12**
  - Crowdsourcing platforms and Design principles
- **Our Tutorial**
  - Control quality, cost and latency
  - Design Crowdsourced Database

Outline

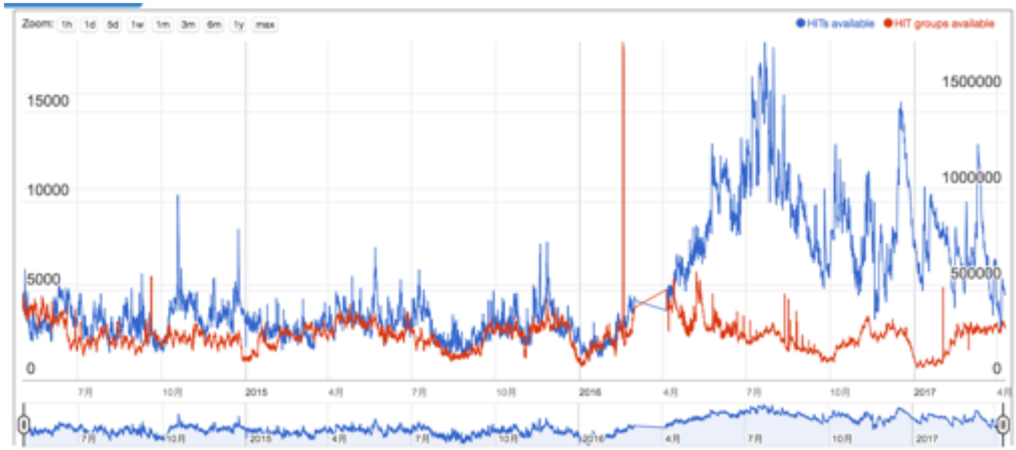
- Crowdsourcing Overview (30min)
  - Motivation (5min)
  - Workflow (15min)
  - Platforms (5min)
  - Difference from Other Tutorials (5min)
- Fundamental Techniques (100min)
  - Quality Control (60min)
  - Cost Control (20min)
  - Latency Control (20min)
- Crowdsourced Database Management (40min)
  - Crowdsourced Databases (20min)
  - Crowdsourced Optimizations (10min)
  - Crowdsourced Operators (10min)
- Challenges (10min)

Part 1

Part 2

Why Quality Control?

- **Huge Amount** of Crowdsourced Data



amazonmechanical turk
Artificial Artificial Intelligence

Statistics in AMT:
Over 500K workers
Over 1M tasks

- **Inevitable noise & error**

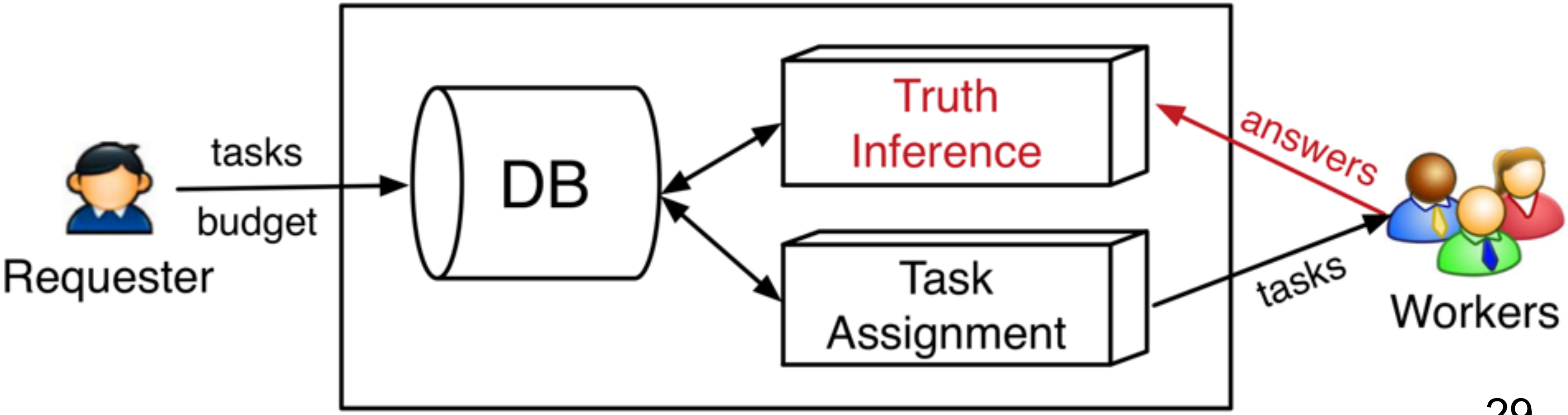
- **Goal: Obtain reliable information in Crowdsourced Data**

Crowdsourcing Workflow

- **Requester** deploys tasks and budget on crowdsourcing platform (e.g., AMT)
- **Workers** interact with platform (**2 phases**)

  (1) when a worker comes to the platform, the worker will be assigned to a set of tasks (**task assignment**);

  (2) when a worker accomplishes tasks, the platform will collect answers from the worker (**truth inference**).



Outline of Quality Control

- Part I. Truth Inference
  - Problem Definition
  - Condition 1: with ground truth
    - Qualification Test & Hidden Test
  - Condition 2: without ground truth
    - Unified Framework
    - Differences in Existing Works
    - Experimental Results

- Part II. Task Assignment
  - Problem Definition
  - Differences in Existing Works

Part I. Truth Inference

- **An Example Task**

What is the current affiliation for Michael Franklin ?

A. University of California, Berkeley
B. University of Chicago

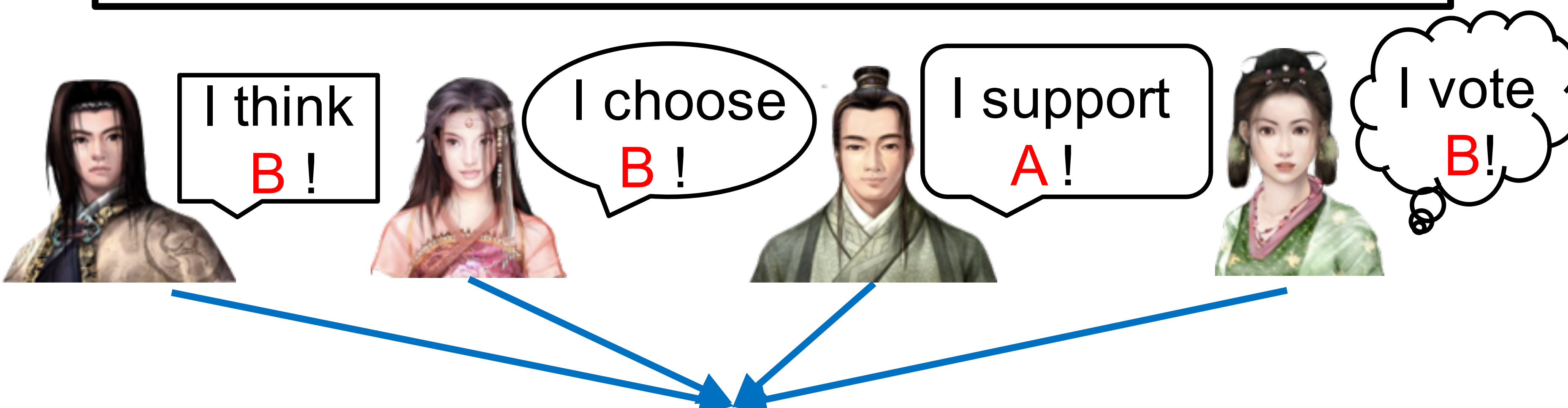
I support
A. UCB !

Principle: Redundancy

- Collect Answers from Multiple Workers

What is the current affiliation for Michael Franklin ?

A. University of California, Berkeley
B. University of Chicago



How to infer the truth of the task ?

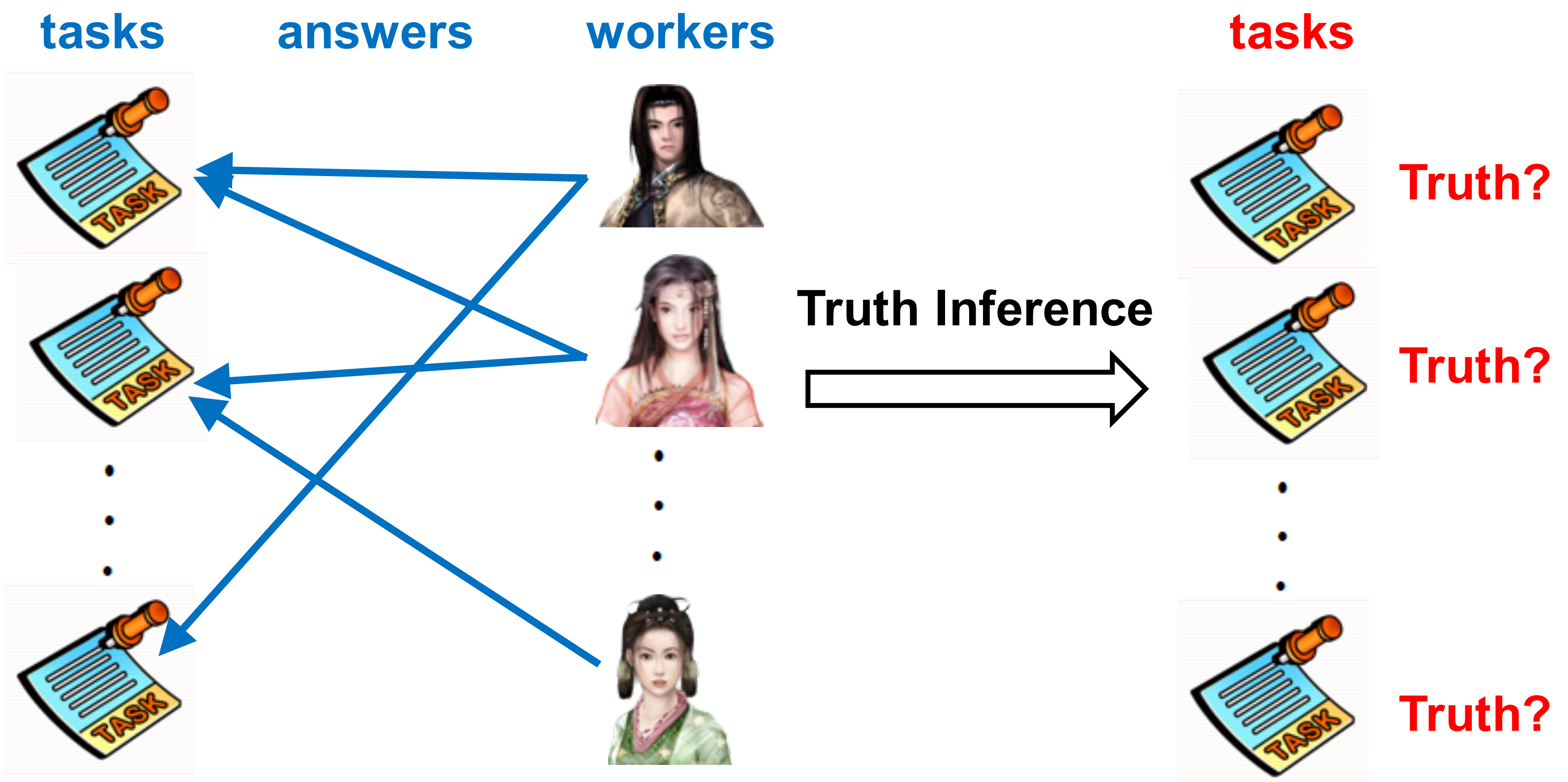
Outline of Quality Control

- Part I. Truth Inference
  - Problem Definition
  - Condition 1: with ground truth
    - Qualification Test & Hidden Test
  - Condition 2: without ground truth
    - Unified Framework
    - Differences in Existing Works
    - Experimental Results

- Part II. Task Assignment
  - Problem Definition
  - Differences in Existing Works

Truth Inference Definition

Given different tasks' answers collected from workers, the target is to infer the truth of each task.



A Simple Solution

- **Majority Voting**

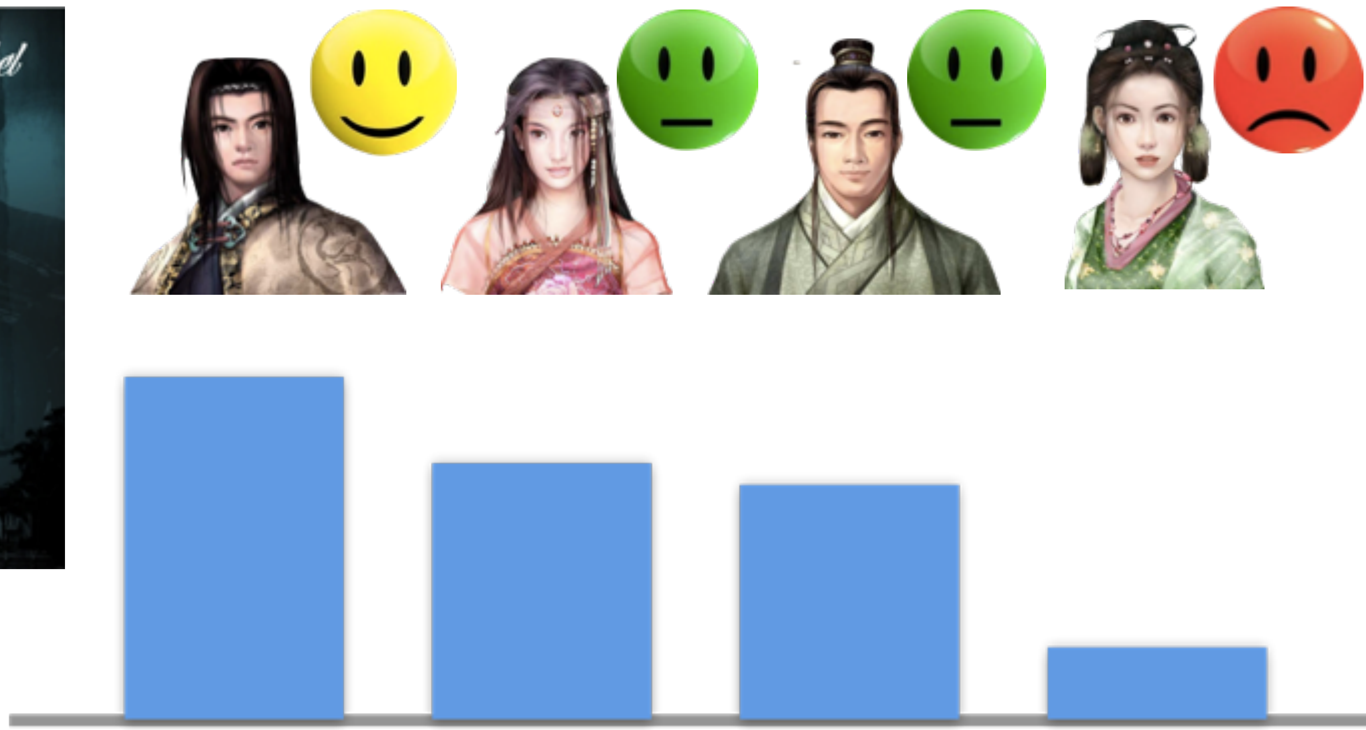
  Take the answer that is voted by the majority (or most) of workers.

- **Limitation**

  Treat each worker equally, neglecting the diverse quality for each worker.

The Key to Truth Inference

- The key is to know **each worker's quality**



Suppose quality of 4 workers are known

How to know worker's quality ?

- 1. If **a small set of tasks with ground truth are known** in advance (e.g., refer to experts)

We can estimate each worker's quality based on the *answering performance for the tasks with known truth*

- 2. If **no ground truth is known** in advance

The only way is to estimate each worker's quality based on *the collected answers from all workers for all tasks*

Outline

- Part I. Truth Inference
  - Problem Definition
  - Condition 1: with ground truth
    - Qualification Test & Hidden Test
  - Condition 2: without ground truth
    - Unified Framework
    - Existing Works
    - Experimental Results
- Part II. Task Assignment
  - Problem Definition
  - Differences in Existing Works

1. A Small Set of Ground Truth is Known

- **Qualification Test (*like an "exam"*)**

amazonmechanical turk
Artificial Artificial Intelligence

Assign the tasks (with known truth) to the worker when the worker comes at first time
e.g., if the worker answers 8 over 10 tasks correctly, then the quality is 0.8

- **Hidden Test (*like a "landmine"*)**

Embed the tasks (with known truth) in all the tasks assigned to the worker
e.g., each time 10 tasks are assigned to a worker, then 10 tasks compose of 9 real tasks (with unknown truth), and 1 task with known truth

1. A Small Set of Ground Truth is Known

- **Limitations of two approaches**

  (1) need to know ground truth (may refer to experts);

  (2) waste of money because workers need to answer these "extra" tasks;

  (3) as reported (Zheng et al. VLDB'17), these techniques may not improve much quality.

Thus the assumption of "no ground truth is known" is widely adopted by existing works

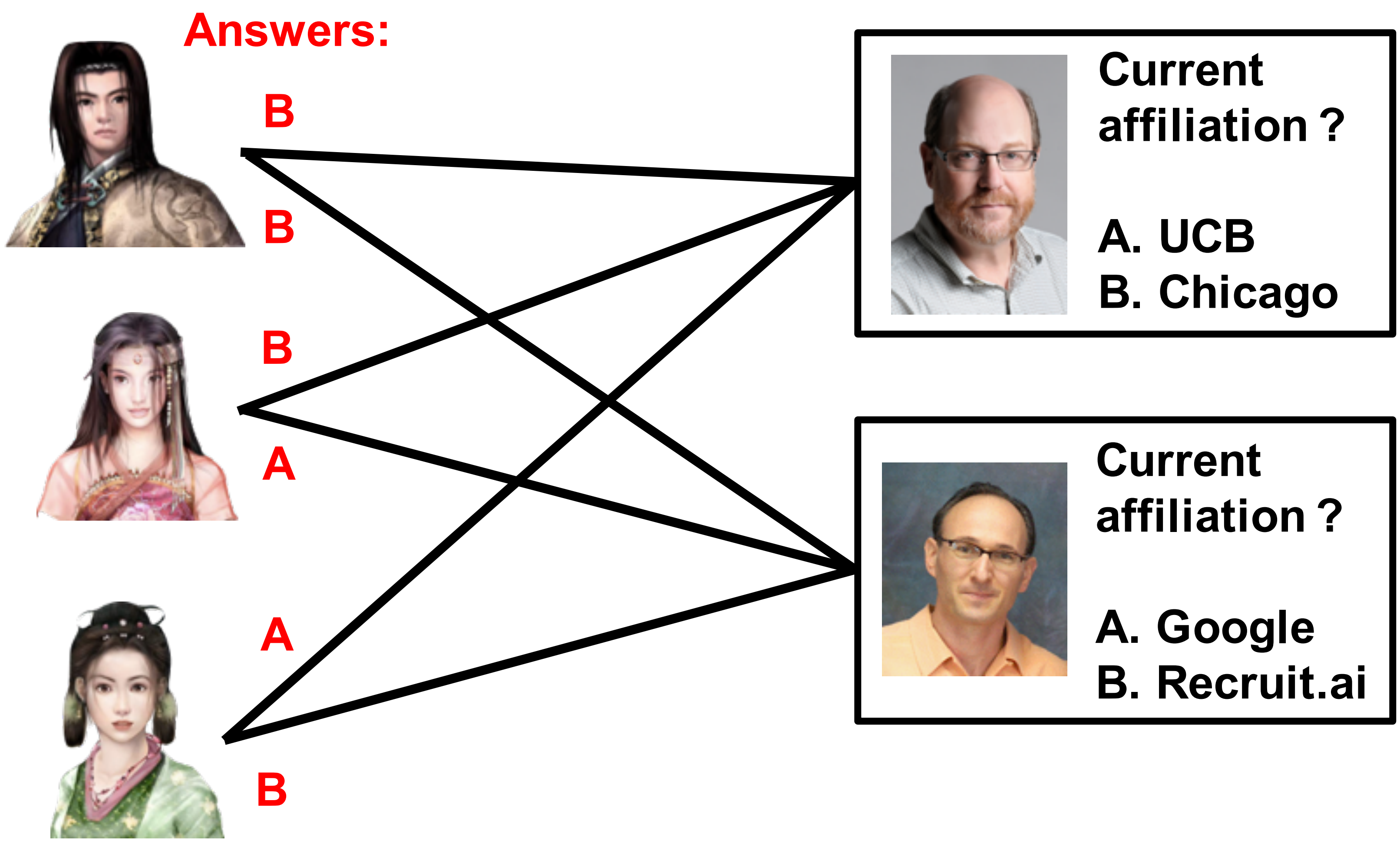
Outline

- Part I. Truth Inference
  - Problem Definition
  - Condition 1: with ground truth
    - Qualification Test & Hidden Test
  - Condition 2: without ground truth
    - Unified Framework
    - Existing Works
    - Experimental Results

- Part II. Task Assignment
  - Problem Definition
  - Differences in Existing Works

2. If No Ground Truth is Known

- How to know each worker's quality given the collected answers for all tasks ?



Unified Framework in Existing Works

- **Input: Workers' answers for all tasks**
- **Algorithm Framework:**

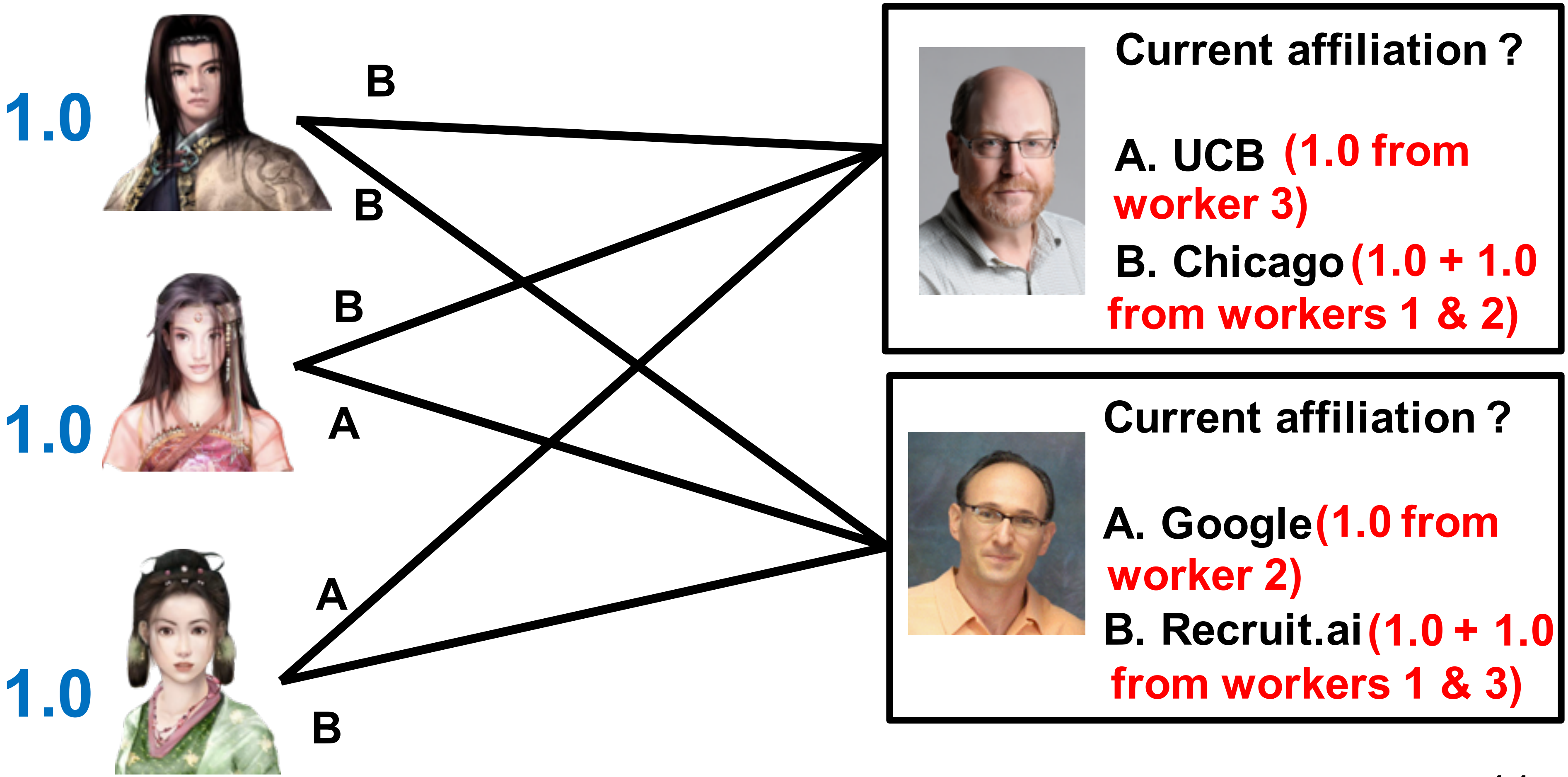
```
Initialize Quality for each worker
while (not converged) {
    Quality for each worker ⟹ Truth for each task ;
    Truth for each task ⟹ Quality for each worker ;
}
```

- **Output: Quality for each worker and Truth for each task**

Inherent Relationship 1

- 1. Quality for each worker ⟹ Truth for each task

Quality: Truth:



Inherent Relationship 2

- 2. Truth for each task → Quality for each worker

Truth:

Current affiliation ?

A. UCB
B. Chicago

Current affiliation ?

A. Google
B. Recruit.ai

Quality:

B
B

1.0

correct: 2/2

B
A

0.5

correct: 1/2

A
B

0.5

correct: 1/2

Outline

- **Part I. Truth Inference**
  - Problem Definition
  - Condition 1: with ground truth
    - Qualification Test & Hidden Test
  - Condition 2: without ground truth
    - Unified Framework
    - Existing Works
    - Experimental Results
- **Part II. Task Assignment**
  - Problem Definition
  - Differences in Existing Works

Existing works

- **Classic Method**

D&S [Dawid and Skene. JRSS 1979]

- **Recent Methods**

(1) Database Community:

CATD [Li et al. VLDB14], PM [Li et al. SIGMOD14], iCrowd [Fan et al. SIGMOD15], DOCS [Zheng et al. VLDB17]

(2) Data Mining Community:

ZC [Demartini et al. WWW12], Multi [Welinder et al. NIPS 2010], CBCC [Venanzi et al. WWW14]

(3) Machine Learning Community:

GLAD [Whitehill et al. NIPS09], Minimax [Zhou et al. NIPS12], BCC [Kim et al. AISTATS12], LFC [Raykar et al. JLMR10], KOS [Karger et al. NIPS11], VI-BP [Liu et al. NIPS12], VI-MF [Liu et al. NIPS12], LFC_N [Raykar et al. JLMR10]

Differences in Existing works

Tasks

- **Different Task Types**
  What type of tasks they focus on ?
  E.g., single-label tasks …

- **Different Task Models**
  How they model each task ?
  E.g., task difficulty …

Workers

- **Different Worker Models**
  How they model each worker ?
  E.g., worker probability (a value) …

Tasks: Different Tasks Types

- **Decision-Making Tasks** (yes/no task)

Is Bill Gates currently the CEO of Microsoft ?
O Yes O No

e.g., Demartini et al. WWW12, Whitehill et al. NIPS09, Kim et al. AISTATS12, Venanzi et al. WWW14, Raykar et al. JLMR10

- **Single-Label Tasks** (multiple choices)

Identify the sentiment of the tweet: ……
O Pos O Neu O Neg

e.g., Li et al. VLDB14, Li et al. SIGMOD14, Demartini et al. WWW12, Whitehill et al. NIPS09, Kim et al. AISTATS12

- **Numeric Tasks** (answer with numeric values)

What is the height for Mount Everest ?
[ ] m

e.g., Li et al. VLDB14, Li et al. SIGMOD14

Tasks: Different Tasks Models

- **Task Difficulty: a value**

If a task receives many contradicting (or ambiguous) answers, then it is regarded as a difficult task.

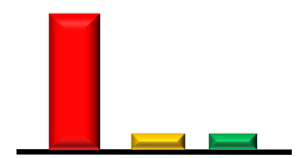
e.g., Welinder et al. NIPS 2010, Ma et al. KDD16

- **Diverse Domains: a vector**

Sports Politics Entertainment

Did Michael Jordan win more NBA championships than Kobe Bryant?

Sports



Is there a name for the song that FC Barcelona is known for?

Sports & Entertainment

Tasks: Different Task Models (cont’d)

- Diverse Domains (cont’d)

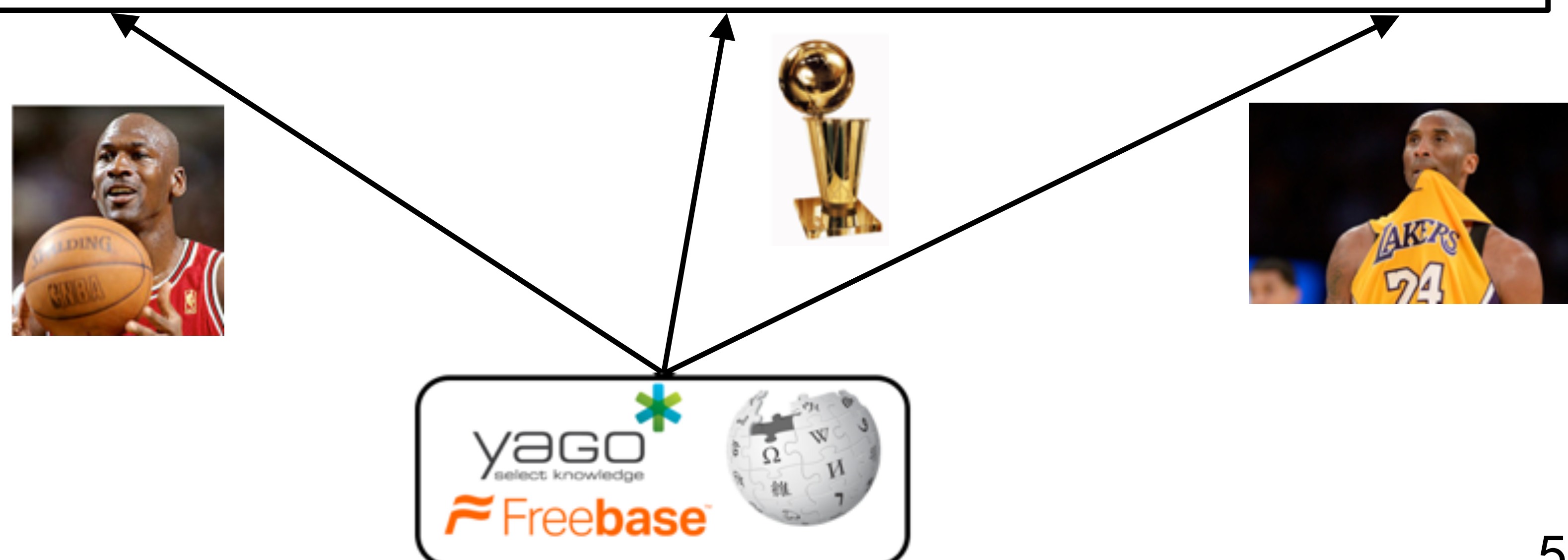
To obtain the each task’s model:

(1) Use machine learning approaches

e.g., LDA [Blei e al. JMLR03],
TwitterLDA [Zhao et al. ECIR11].

(2) Use entity linking (map entity to knowledge bases).

Did Michael Jordan win more NBA championships than Kobe Bryant?



Workers: Different Worker Models

- **Worker Probability**: a value $p \in [0,1]$

  The probability that the worker answers tasks correctly
  e.g., a worker answers 8 over 10 tasks correctly, then the worker probability is 0.8.

  e.g., Demartini et al. WWW12, Whitehill et al. NIPS09

- **Confidence Interval**: a range $[p-\varepsilon, p+\varepsilon]$

  ε is related to the number of tasks answered
  => the more answers collected, the smaller ε is.
  e.g., two workers answer 8 over 10 tasks and 40 over 50 tasks correctly, then the latter worker has a smaller ε.

  e.g., Li et al. VLDB14

Workers: Different Worker Models (cont'd)

○ **Confusion Matrix**: a matrix

Capture a worker's answer for different choices given a specific truth

$$\begin{array}{c c} & \begin{array}{ccc} \textit{Pos} & \textit{Neu} & \textit{Neg} \end{array} \\ \begin{array}{c} \textit{Pos} \\ \textit{Neu} \\ \textit{Neg} \end{array} & \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.3 & 0.6 & 0.1 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} \end{array}$$

Given that the truth of a task is "Neu", the probability that the worker answers "Pos" is 0.3.

e.g., Kim et al. AISTATS12, Venanzi et al. WWW14

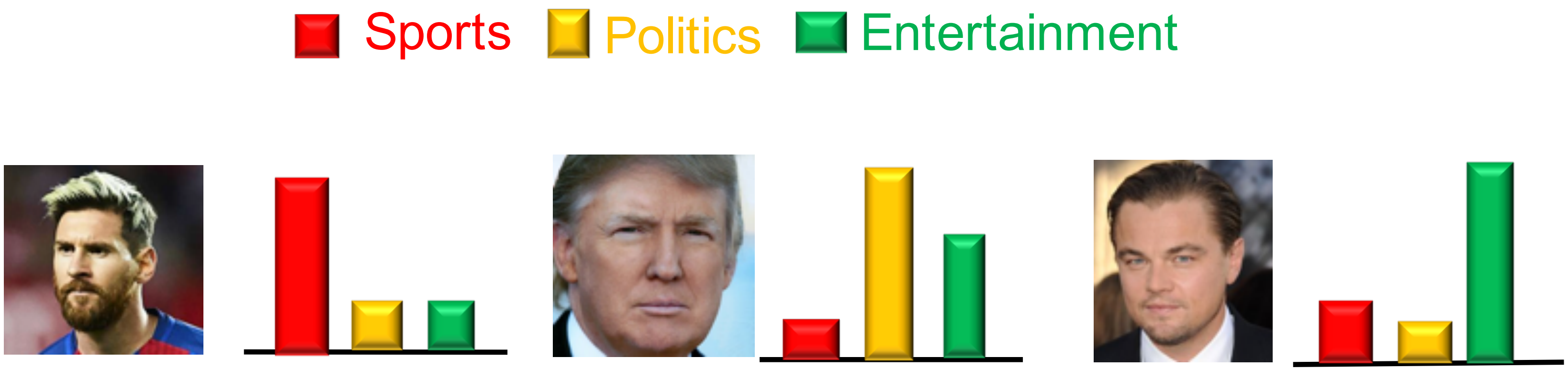
○ **Bias τ & Variance σ**: numerical task

Answer follows Gaussian distribution: $ans \sim N(t+\tau, \sigma)$

e.g., Raykar et al. JLMR10

Workers: Different Worker Models (cont’d)

- **Quality Across Diverse Domains**: **a vector**



How to decide the scope of domains ?

Idea: Use domains from Knowledge Bases

e.g., Ma et al. KDD16, Zheng et al. VLDB17

Summary of Truth Inference Methods

Method	Task Type	Task Model	Worker Model
Majority Voting	Decision-Making Task, Single-Choice Task	No	No
Mean / Median	Numeric Task	No	No
ZC [Demartini et al. WWW12]	Decision-Making Task, Single-Choice Task	No	Worker Probability
GLAD [Whitehill et al. NIPS09]	Decision-Making Task, Single-Choice Task	Task Difficulty	Worker Probability
D&S [Dawid and Skene. JRSS 1979]	Decision-Making Task, Single-Choice Task	No	Confusion Matrix
Minimax [Zhou et al. NIPS12]	Decision-Making Task, Single-Choice Task	No	Diverse Domains
BCC [Kim et al. AISTATS12]	Decision-Making Task, Single-Choice Task	No	Confusion Matrix
CBCC [Venanzi et al. WWW14]	Decision-Making Task, Single-Choice Task	No	Confusion Matrix
LFC [Raykar et al. JLMR10]	Decision-Making Task, Single-Choice Task	No	Confusion Matrix
CATD [Li et al. VLDB14]	Decision-Making Task, Single-Choice Task, Numeric Task	No	Worker Probability, Confidence

Summary of Truth Inference Methods (cont'd)

Method	Task Type	Task Model	Worker Model
PM [Li et al. SIGMOD14]	Decision-Making Task, Single-Choice Task, Numeric Task	No	Worker Probability
Multi [Welinder et al. NIPS 2010]	Decision-Making Task	Diverse Domains	Diverse Domains, Worker Bias, Worker Variance
KOS [Karger et al. NIPS11]	Decision-Making Task	No	Worker Probability
VI-BP [Liu et al. NIPS12]	Decision-Making Task	No	Confusion Matrix
VI-MF [Liu et al. NIPS12]	Decision-Making Task	No	Confusion Matrix
LFC_N [Raykar et al. JLMR10]	Numeric Task	No	Worker Variance
iCrowd [Fan et al. SIGMOD15]	Decision-Making Task, Single-Choice Task	Diverse Domains	Diverse Domains
FaitCrowd [Ma et al. KDD16]	Decision-Making Task, Single-Choice Task	Diverse Domains	Diverse Domains
DOCS [Zheng et al. VLDB17]	Decision-Making Task, Single-Choice Task	Diverse Domains	Diverse Domains

Outline

- Part I. Truth Inference
  - Problem Definition
  - Condition 1: with ground truth
    - Qualification Test & Hidden Test
  - Condition 2: without ground truth
    - Unified Framework
    - Existing Works
    - Experimental Results
- Part II. Task Assignment
  - Problem Definition
  - Differences in Existing Works

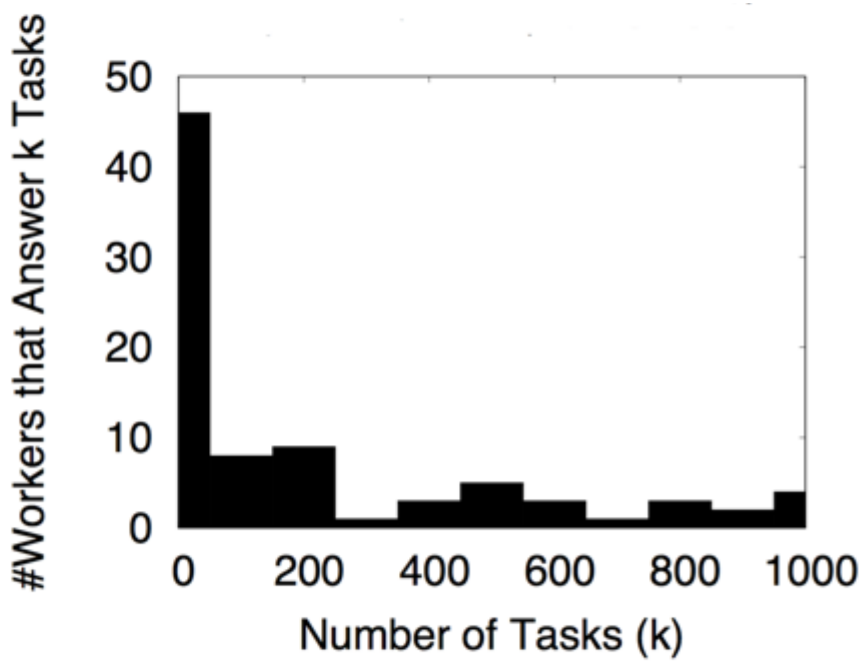
Experimental Results (Zheng et al. VLDB17)

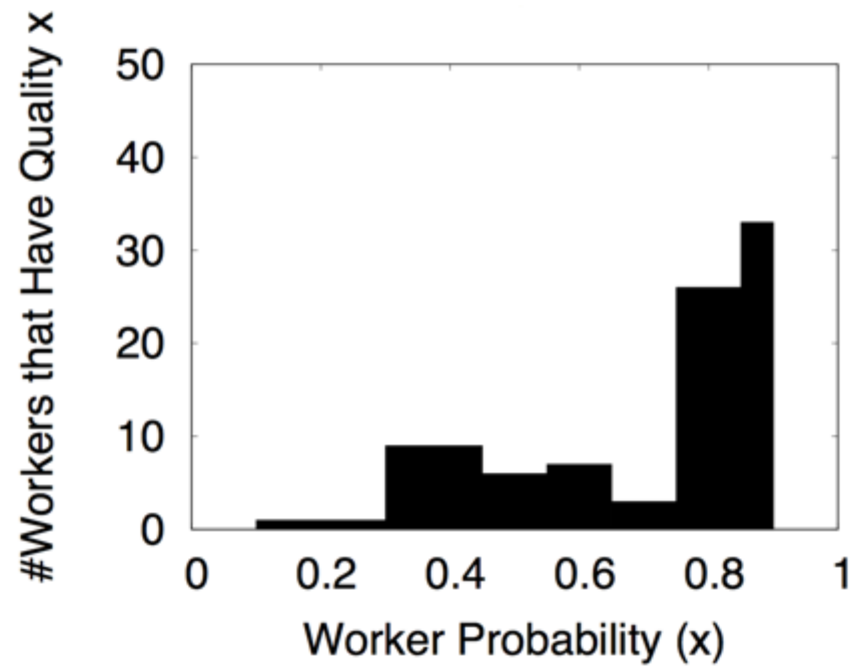
- **Statistics of Datasets**

Dataset	# Tasks	# Answers Per Task	# Workers	Description
Sentiment Analysis [Zheng et al. VLDB17]	1000	20	185	Given a tweet, the worker will identify the sentiment of the tweet
Duck [Welinder et al. NIPS10]	108	39	39	Given an image, the worker will identify whether the image contains a duck or not
Product [Wang et al. VLDB12]	8315	3	85	Given a pair of products, the worker will identify whether or not they refer to the same product

Experimental Results

- **Observations (Sentiment Analysis)**



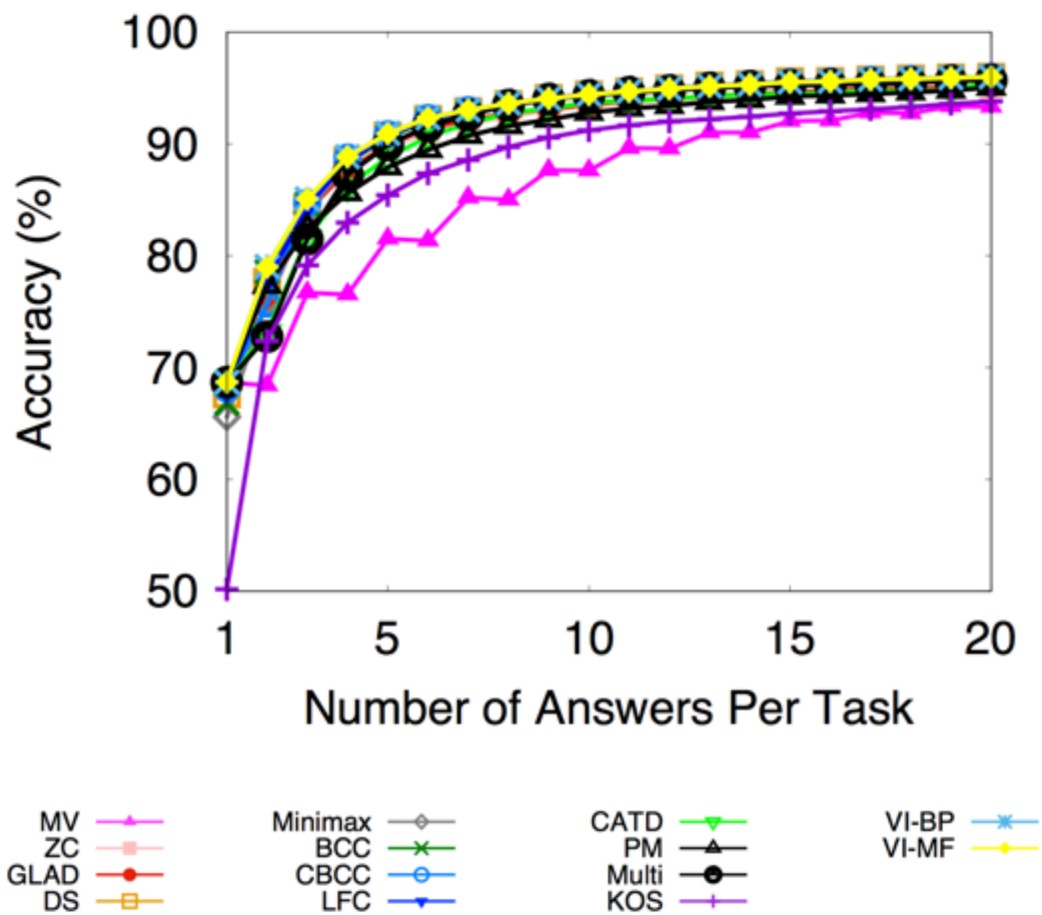


#workers' answers conform to long-tail phenomenon (Li et al. VLDB14)

Not all workers are of very high quality

Experimental Results (cont'd)

- **Change of Quality vs. #Answers (Sentiment Analysis)**



Observations:

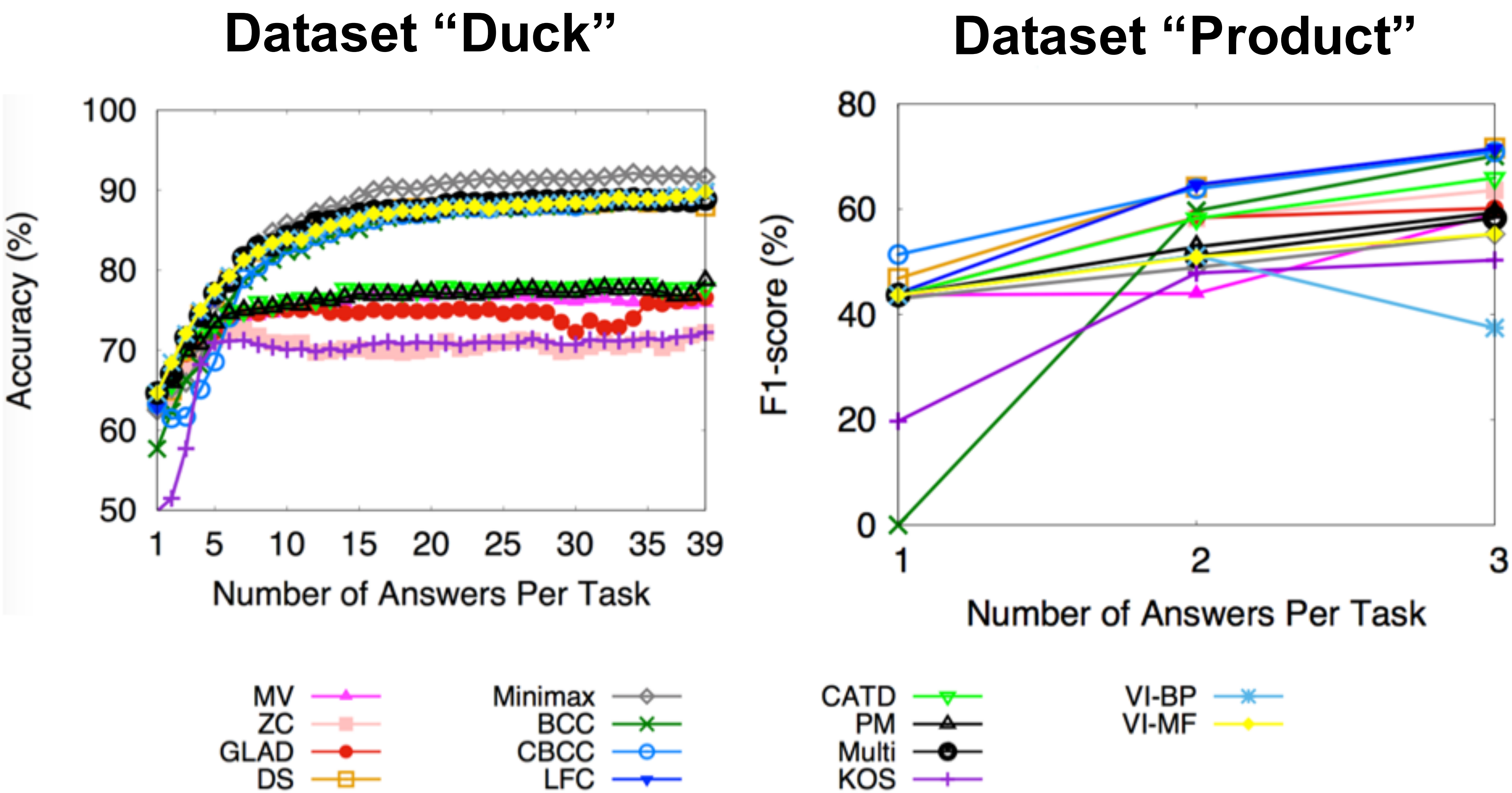
1. The quality increases with #answers;

2. The quality improvement is significant with few answers, and is marginal with more answers;

3. Most methods are similar, except for Majority Voting (in pink color).

Experimental Results (cont'd)

- Performance on more datasets



Which method is the best ?

- **Decision-Making & Single-Label Tasks**
  - **"Majority Voting" if sufficient data is given (each task collects more than 20 answers);**
  - **"D&S [Dawid and Skene JRSS 1979]" if limited data is given (a robust method);**
  - **"Minimax [Zhou et al. NIPS12]" and "Multi [Welinder et al. NIPS 2010]" as advanced techniques.**

- **Numeric Tasks**
  - **"Mean" since it is robust in practice;**
  - **"PM [Li et al. SIGMOD14]" as advanced techniques.**

Take-Away for Truth Inference

- The key to truth is to **compute each worker's quality**
- if some truth is known: 

  qualification test and **hidden test**;
- if no truth is known: 

  (1) relationships between **"quality for each worker"** and **"truth for each task"**

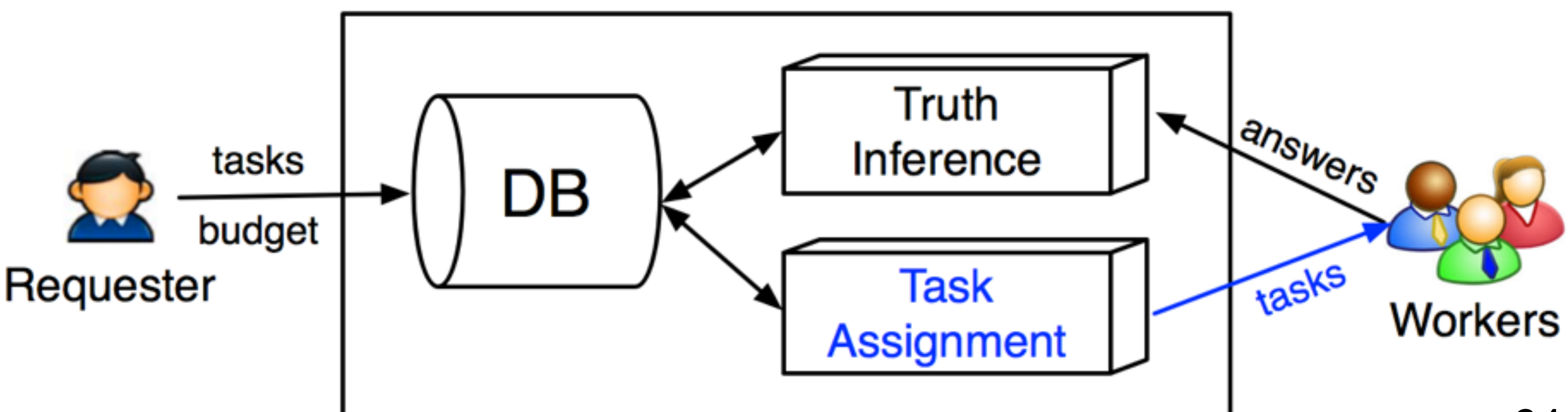
  (2) different **task types & models** and **worker models**

Crowdsourcing Workflow

- Requester deploys tasks and budget on crowdsourcing platform (e.g., Amazon Mechanical Turk)
- Workers interact with platform (2 phases)

  (1) when a worker comes to the platform, the worker will be assigned to a set of tasks (task assignment);

  (2) when a worker accomplishes tasks, the platform will collect answers from the worker (truth inference).



Part II. Task Assignment

- Existing platforms support online task assignment

amazonmechanical turk Artificial Artificial Intelligence ⟹ **"External HIT"**

- Intuition: requesters want to wisely use the budgets

I am requester, and I want to use my budgets very well !

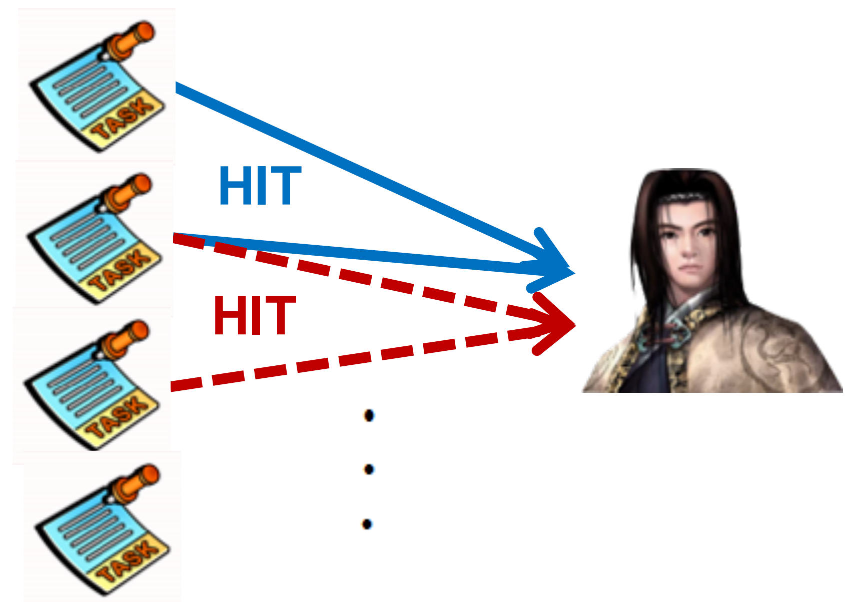
We are workers !

How to allocate suitable tasks to workers?

Task Assignment Problem

Given a pool of n tasks, which set of the k tasks should be batched in a HIT and assigned to the worker?

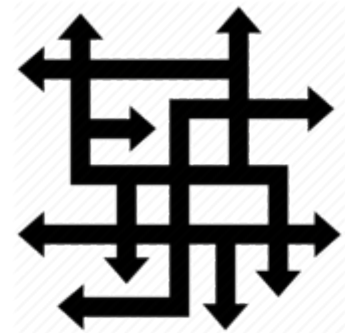
Example:
Suppose we have n=4 tasks, and each time k=2 tasks are assigned as a HIT.



This problem is complex!

- Simple enumeration:
  "n choose k" combinations

  (n = 100, k = 5) ➔ 100M assignments



- Need efficient (online) assignment

  Fast response to worker's request

- Develop efficient heuristics

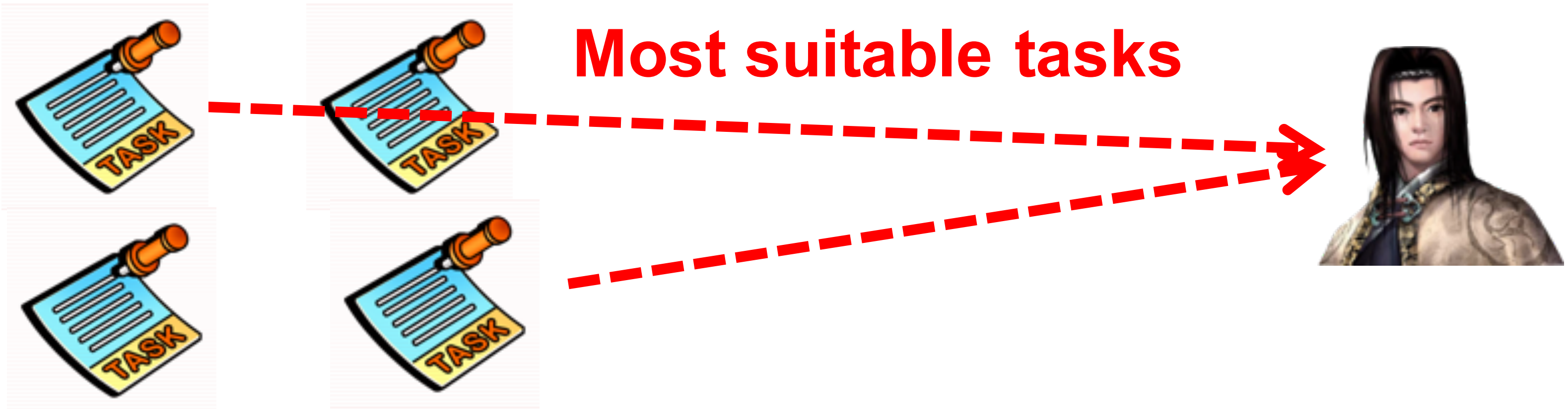
  Assignment time linear in #tasks: O(n)

Outline

- Part I. Truth Inference
  - Problem Definition
  - Condition 1: with ground truth
    - Qualification Test & Hidden Test
  - Condition 2: without ground truth
    - Unified Framework
    - Existing Works
    - Experimental Results

- Part II. Task Assignment
  - Problem Definition
  - **Existing Works**

Main Idea



3 factors for characterizing a suitable task:

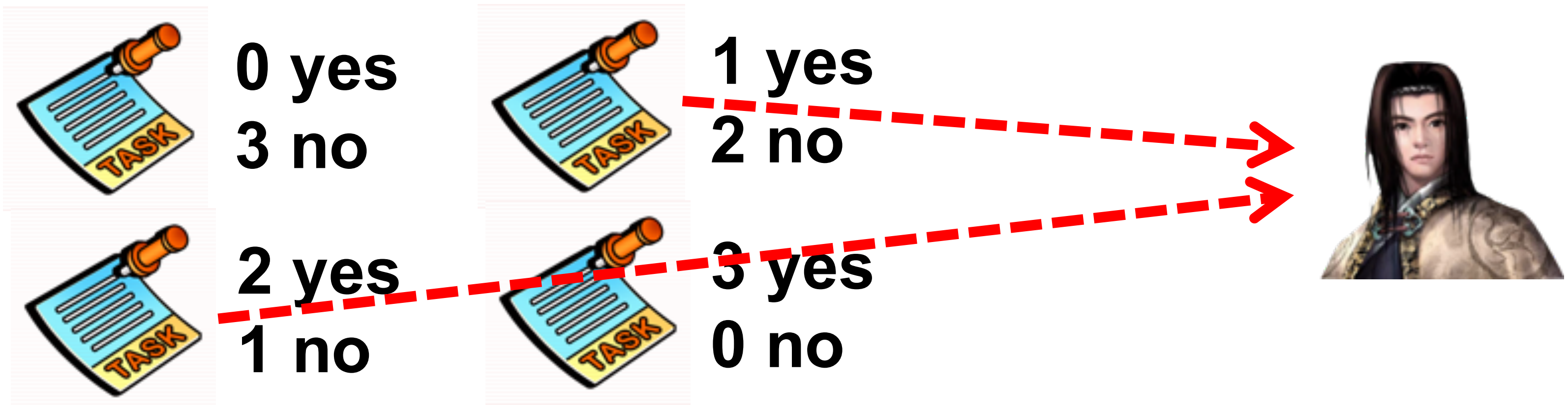
Answer uncertainty

Worker quality

Requesters’ objectives

Factor 1: Answer Uncertainty

○ Consider a decision-making task (yes/no)



○ Select a task whose answers are the most uncertain or inconsistent

e.g., Liu et al. VLDB12, Roim et al. ICDE12

Factor 1: Answer Uncertainty

- **Entropy (Zheng et al. SIGMOD15)**

  Given *c* choices for a task and the distribution of answers for a task $\vec{p} = (p_1, p_2, \ldots, p_c)$

  The task's entropy is:

$$H(\vec{p}) = -\sum_{i=1}^{c} p_i \log p_i$$

  e.g., a task receives 1 "yes" and 2 "no", then the distribution is (1/3, 2/3), and entropy is 0.637.

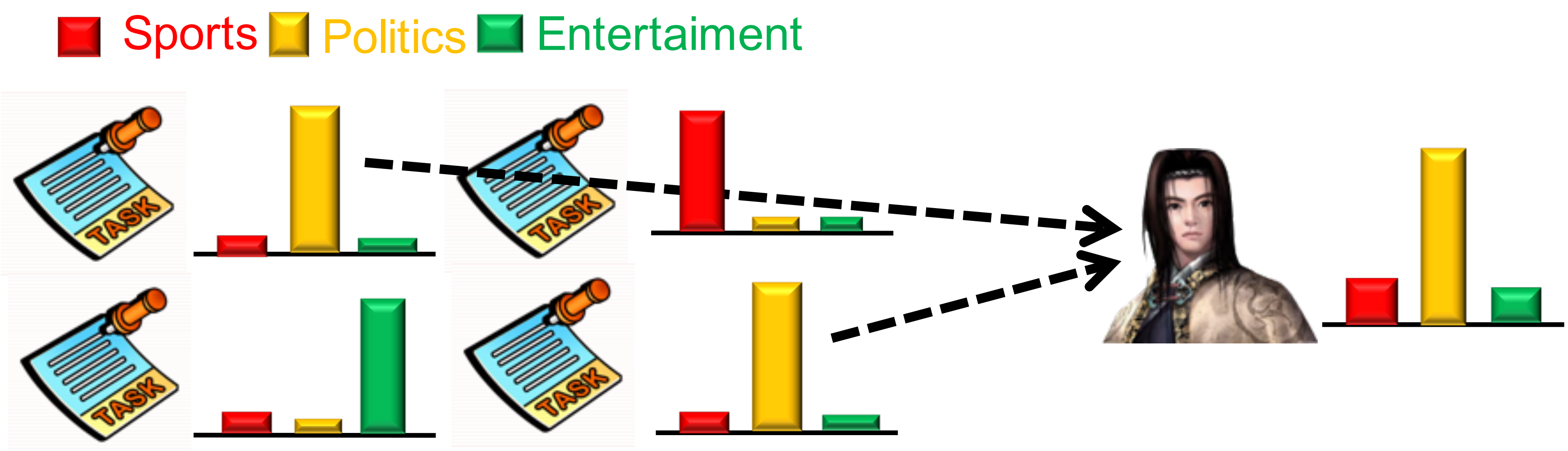
- **Expected change of entropy (Roim et al. ICDE12)**

  (1/3, 2/3) should be more uncertain than (10/30, 20/30):

$$E[H(\vec{p}')] - H(\vec{p})$$

Factor 2: Worker Quality

- Assign tasks to the worker with the suitable expertise



- Uncertainty: consider **the matching domains** in tasks and the worker

e.g., Ho et al. AAAI12, Zheng et al. VLDB17

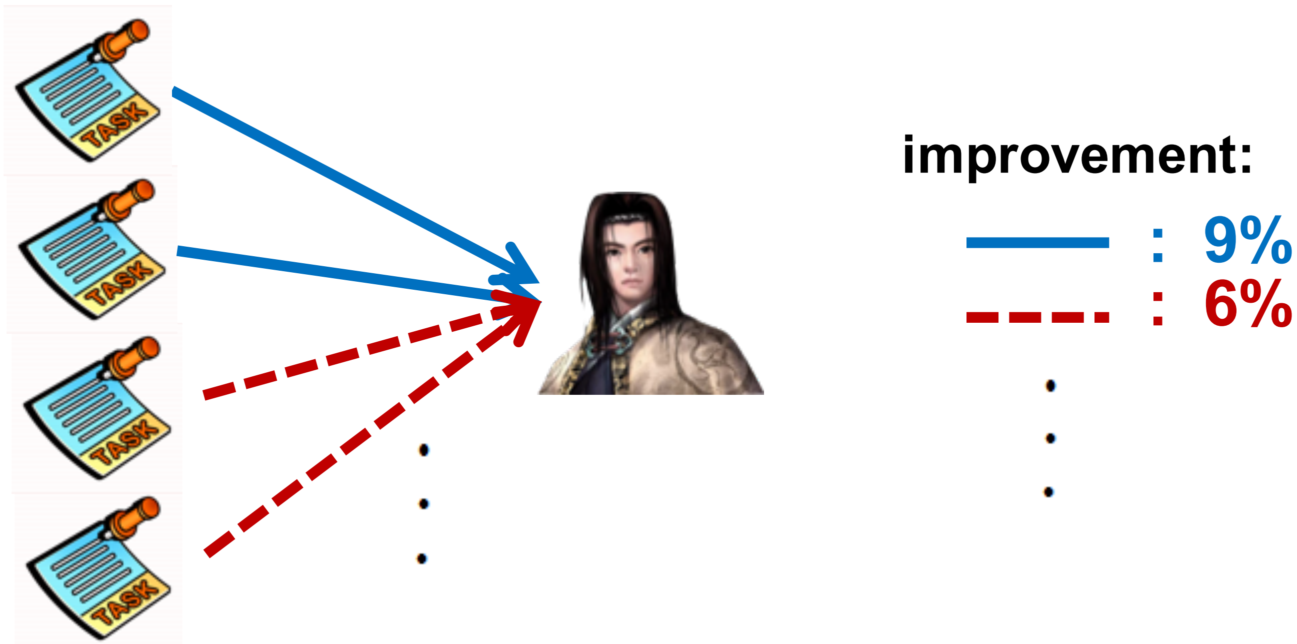
Factor 3: Objectives of Requesters

- Requesters may have different objectives (aka **"evaluation metric"**) for different applications

Application	Sentiment Analysis	Entity Resolution
Task	I had to wait for six friggin' hours in line at the @apple store. ◎positive ◎neutral ◎negative	iPad 2 = iPad 3rd Gen ? ◎ *equal* ◎ *non-equal*
Evaluation Metric	Accuracy	F-score ("equal" label)

Factor 3: Objectives of Requesters

- Solution in QASCA (Zheng et al. SIGMOD15)
  (1) Leverage the answers collected from workers to create a "distribution matrix";
  (2) leverage the "distribution matrix" to estimate the quality improvement for a specific set of selected tasks.
- Idea: Select the best set of tasks with highest quality improvement in the specified evaluation metric.



Factor 3: Objectives of Requesters

- **Other Objectives**

(1) Threshold on entropy (e.g., Li et al. WSDM17)
e.g., in the final state, each task should have constraint that its entropy ≥ 0.6.

(2) Threshold on worker quality (e.g., Fan et al. SIGMOD15)
e.g., in the final state, each task should have overall aggregated worker quality ≥ 2.0.

(3) Maximize total utility (e.g., Ho et al. AAAI12)
e.g., after the answer is given, the requester receives some utility related to worker quality, and the goal is to assign tasks that maximize the total utility.

Task Assignment

Method	Factor 1: Answer Uncertainty	Factor 2: Worker Quality	Factor 3: Requesters' Objectives
OTA [Ho et al. AAAI12]	Majority	Worker probability	Maximize total utility
CDAS [Liu et al. VLDB12]	Majority	Worker probability	A threshold on confidence + early termination of confident tasks
iCrowd [Fan et al. SIGMOD15]	Majority	Diverse domains	Maximize overall worker quality
AskIt! [Roim et al. ICDE12]	Entropy-based	No	No
QASCA [Zheng et al. SIGMOD15]	Maximize specified quality	Confusion matrix	Maximize specified quality
DOCS [Zheng et al. VLDB17]	Expected change of entropy	Diverse domains	No
CrowdPOI [Hu et al. ICDE16]	Expected change of accuracy	Worker probability	No
Opt-KG [Li et al. WSDM17]	Majority	No	≥ threshold on entropy

Take-Away for Task Assignment

- **Require online and efficient heuristics**

- **Key idea: assign the most suitable task to worker, based on:**

  (1) uncertainty of collected answers;
  (2) worker quality; and
  (3) requester' objectives.

Public Datasets & Codes

- **Public crowdsourcing datasets** (http://i.cs.hku.hk/~ydzheng2/crowd_survey/datasets.html).

- **Implementations of truth inference algorithms** (https://github.com/TsinghuaDatabaseGroup/crowdsourcing/tree/master/truth/src/methods).

- **Implementations of task assignment algorithms** (https://github.com/TsinghuaDatabaseGroup/CrowdOTA).

Reference – Truth Inference

[1] ZenCrowd: G. Demartini, D. E. Difallah, and P. Cudré-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In WWW, pages 469–478, 2012.

[2] EM: A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. J.R.Statist.Soc.B, 30(1):1–38, 1977.

[3] Most Traditional Work (D&S): A.P.Dawid and A.M.Skene. Maximum likelihood estimation of observererror-rates using em algorithm. Appl.Statist., 28(1):20–28, 1979.

[4] iCrowd: J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng. icrowd: An adaptivecrowdsourcing framework. In SIGMOD, pages 1015–1030, 2015.

[5] J. Gao, Q. Li, B. Zhao, W. Fan, and J. Han. Truth discovery andcrowdsourcing aggregation: A unified perspective. VLDB, 8(12):2048–2049, 2015

[6] CrowdPOI: H. Hu, Y. Zheng, Z. Bao, G. Li, and J. Feng. Crowdsourced poi labelling:Location-aware result inference and task assignment. In ICDE, 2016.

[7] P. Ipeirotis, F. Provost, and J. Wang. Quality management on amazonmechanical turk. In SIGKDD Workshop, pages 64–67, 2010.

[8] M. Joglekar, H. Garcia-Molina, and A. G. Parameswaran. Evaluating thecrowd with confidence. In SIGKDD, pages 686–694, 2013.

[9] G. Li, J. Wang, Y. Zheng, and M. J. Franklin. Crowdsourced datamanagement: A survey. TKDE, 28(9):2296–2319, 2016.

[10] CATD: Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. PVLDB,8(4):425–436, 2014.

[11] PM: Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts inheterogeneous data by truth discovery and source reliability estimation. InSIGMOD, pages 1187–1198, 2014.

[12] KOS / VI-BP / VI-MF: Q. Liu, J. Peng, and A. T. Ihler. Variational inference for crowdsourcing. In NIPS, pages 701–709, 2012.

[13] CDAS: X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. CDAS: Acrowdsourcing data analytics system. PVLDB, 5(10):1040–1051, 2012

Reference – Truth Inference (cont'd)

[14] FaitCrowd: F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han.Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In KDD, pages 745–754. ACM, 2015.
[15] V. C. Raykar and S. Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. Journal of Machine Learning Research,13:491–518, 2012.
[16] V. C. Raykar, S. Yu, L. H. Zhao, A. K. Jerebko, C. Florin, G. H. Valadez,L. Bogoni, and L. Moy. Supervised learning from multiple experts: whom totrust when everyone lies a bit. In ICML, pages 889–896, 2009.
[17] LFC: V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, andL. Moy. Learning from crowds. JMLR, 11(Apr):1297–1322, 2010.
[18] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, Reynold Cheng. Truth Inference in Crowdsourcing: Is the Problem Solved? VLDB 2017.
[19] DOCS: Yudian Zheng, Guoliang Li, Reynold Cheng. DOCS: A Domain-Aware Crowdsourcing System Using Knowledge Bases. VLDB 2017.
[20] CBCC: M. Venanzi, J. Guiver, G. Kazai, P. Kohli, and M. Shokouhi.Community-based bayesian aggregation models for crowdsourcing. In WWW,pages 155–164, 2014.
[21] Minimax: D. Zhou, S. Basu, Y. Mao, and J. C. Platt. Learning from the wisdom ofcrowds by minimax entropy. In NIPS, pages 2195–2203, 2012.
[22] P. Smyth, U. M. Fayyad, M. C. Burl, P. Perona, and P. Baldi. Inferring groundtruth from subjective labelling of venus images. In NIPS, pages 1085–1092,1994.
[23] Multi: P. Welinder, S. Branson, P. Perona, and S. J. Belongie. The multidimensional wisdom of crowds. In NIPS, pages 2424–2432, 2010.
[24] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In NIPS, pages 2035–2043, 2009.
[25] BCC: H.-C. Kim and Z. Ghahramani. Bayesian classifier combination. In AISTATS, pages 619–627, 2012.
[26] Aditya Parameswaran ,Human-Powered Data Management ,
http://msrvideo.vo.msecnd.net/rmcvideos/185336/dl/185336.pdf

Reference – Truth Inference (cont'd)

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3(Jan):993–1022, 2003.
[28] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In ECIR, pages 338–349, 2011.
[29] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. PVLDB, 6(2):37–48, 2012.
[30] X. Liu, X. L. Dong, B. C. Ooi, and D. Srivastava. Online data fusion. PVLDB, 4(11):932–943, 2011.
[31] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. Journal of Machine Learning Research, 3(Jan):993–1022, 2003.
[32] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In ECIR, pages 338–349, 2011.

Reference – Task Assignment

[1] CDAS: X. Liu, M. Lu, B. C. Ooi, Y. Shen, S. Wu, and M. Zhang. CDAS: Acrowdsourcing data analytics system. PVLDB, 5(10):1040–1051, 2012
[2] OTA: C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcingmarkets. In AAAI, 2012.
[3] QASCA: Yudian Zheng, Jiannan Wang, Guoliang Li, Reynold Cheng, Jianhua Feng. QASCA: A Quality-Aware Task Assignment System for Crowdsourcing Applications. SIGMOD 2015.
[4] C.-J. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment forcrowdsourced classification. In ICML, pages 534–542, 2013.
[5] CrowdPOI: H. Hu, Y. Zheng, Z. Bao, G. Li, and J. Feng. Crowdsourced poi labelling:Location-aware result inference and task assignment. In ICDE, 2016.
[6] DOCS: Yudian Zheng, Guoliang Li, Reynold Cheng. DOCS: A Domain-Aware Crowdsourcing System Using Knowledge Bases. VLDB 2017.
[7] AskIt: R. Boim, O. Greenshpan, T. Milo, S. Novgorodov, N. Polyzotis, and W. C. Tan. Asking the right questions in crowd data sourcing. In ICDE, 2012.
[8] iCrowd: J. Fan, G. Li, B. C. Ooi, K. Tan, and J. Feng. icrowd: An adaptivecrowdsourcing framework. In SIGMOD, pages 1015–1030, 2015.
[9] Opt-KG: Qi Li, Fenglong Ma, Jing Gao, Lu Su, and Christopher J Quinn, Crowdsourcing High Quality Labels with a Tight Budget, WSDM 2016.
[10] Jing Gao, Qi Li, Bo Zhao, Wei Fan, and Jiawei Han, Enabling the Discovery of Reliable Information from Passively and Actively Crowdsourced Data, KDD'16 tutorial.

Outline

- Crowdsourcing Overview (30min)
  - Motivation (5min)
  - Workflow (15min)
  - Platforms (5min)
  - Difference from Other Tutorials (5min)
- Fundamental Techniques (100min)
  - Quality Control (60min)
  - Cost Control (20min)
  - Latency Control (20min)
- Crowdsourced Database Management (40min)
  - Crowdsourced Databases (20min)
  - Crowdsourced Optimizations (10min)
  - Crowdsourced Operators (10min)
- Challenges (10min)

Part 1

Part 2

Cost Control

- **Goal**
  - How to reduce monetary cost?

- **Cost = $\boldsymbol{n}\times\boldsymbol{c}$**
  - n: number of tasks
  - c: cost of each task

- **Challenges**
  - How to reduce n?
  - How to reduce c?

Classification of Existing Techniques

- **How to reduce n?**
  - Task Pruning
  - Answer Deduction
  - Task Selection
  - Sampling

The Database Community

- **How to reduce c?**
  - Task Design

The HCI Community

Task Pruning

- **Key Idea**
  - Prune the tasks that machines can do well
- **Easy Task vs. Hard Task**

Are they the same?
IPHONE 6 = iphone 6

Are they the same?
IBM = Big Blue

- **How to quantify "difficulty"**
  - Similarity value
  - Match probability

- Jiannan Wang, Tim Kraska, Michael J. Franklin, Jianhua Feng: CrowdER: Crowdsourcing Entity Resolution. VLDB 2012
- Steven Euijong Whang, Peter Lofgren, Hector Garcia-Molina: Question Selection for Crowd Entity Resolution. VLDB 2013

Task Pruning (cont'd)

- **Workflow (non-iterative)**
  1. Rank tasks based on "difficulty"
  2. Prune the tasks whose difficulty $\leq$ threshold

- **Pros**
  - Support a **large variety** of applications
- **Cons**
  - Only work for **easy** tasks (i.e., the ones that machines can do well)

Classification of Existing Techniques

- **How to reduce n?**
  - Task Pruning
  - Answer Deduction
  - Task Selection
  - Sampling

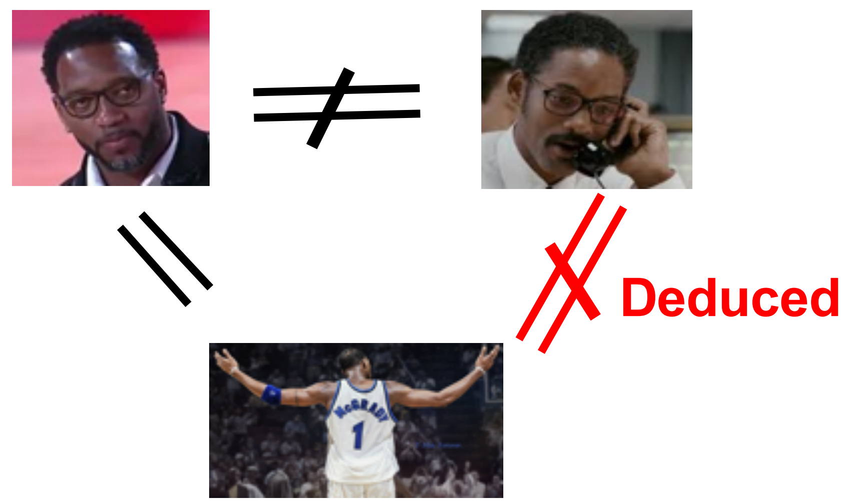
The Database Community

- **How to reduce c?**
  - Task Design

The HCI Community

Answer Deduction

- **Key Idea**
  - Prune the tasks whose answers can be **deduced** from existing crowdsourced tasks
- **Example: Transitivity**

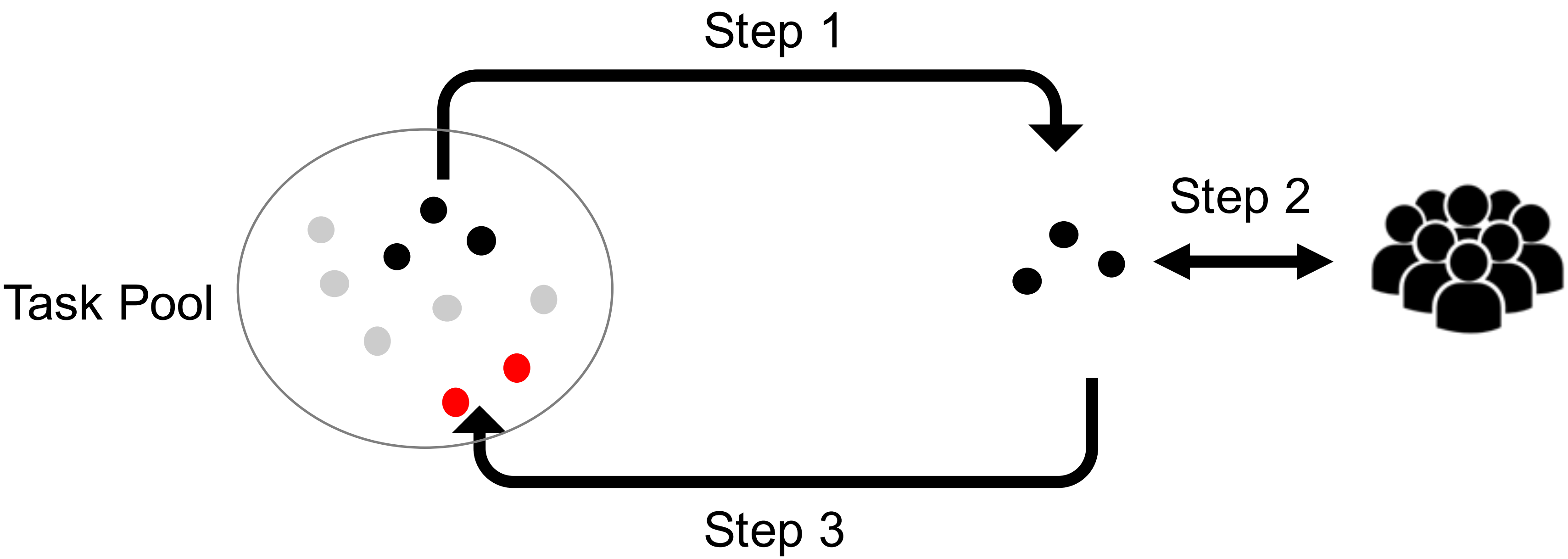


Jiannan Wang, Guoliang Li, Tim Kraska, Michael J. Franklin, Jianhua Feng: Leveraging transitive relations for crowdsourced joins. SIGMOD 2013
Donatella Firmani, Barna Saha, Divesh Srivastava: Online Entity Resolution Using an Oracle. PVLDB 2016

Answer Deduction (cont'd)

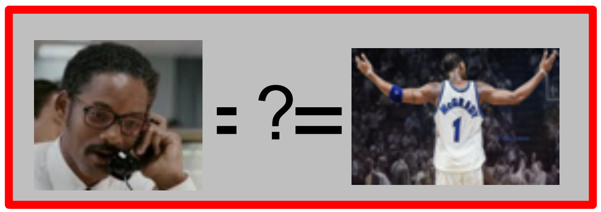
- **Workflow (iterative)**

1. Pick up some tasks from a task pool
2. Collect answers of the tasks from the Crowd
3. Remove the tasks whose answers can be deduced

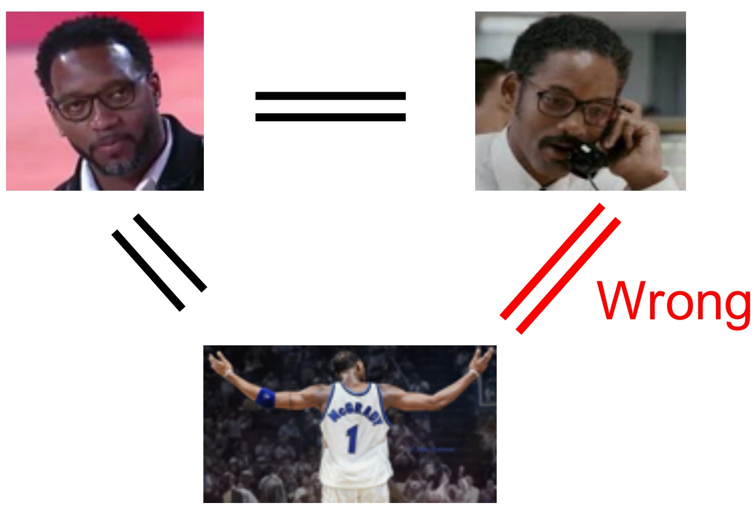


Answer Deduction (cont'd)

- **Pros**
  - Work for both easy and **hard** tasks



- **Cons**
  - Human errors can be amplified



Classification of Existing Techniques

- **How to reduce n?**
  - Task Pruning
  - Answer Deduction
  - Task Selection
  - Sampling

The Database Community

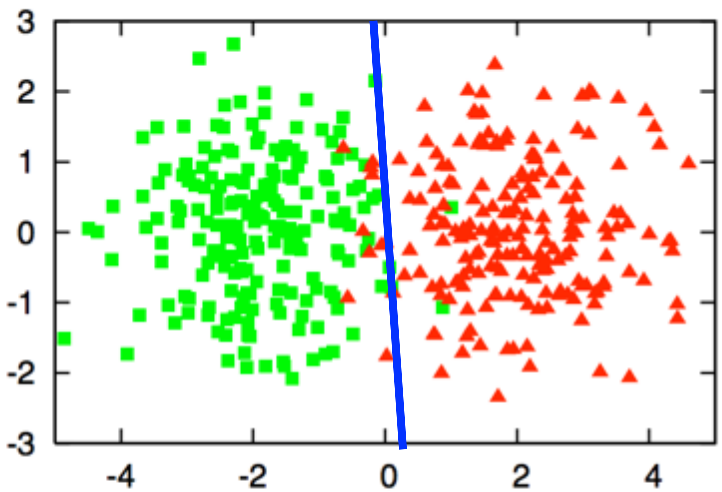
- **How to reduce c?**
  - Task Design

The HCI Community

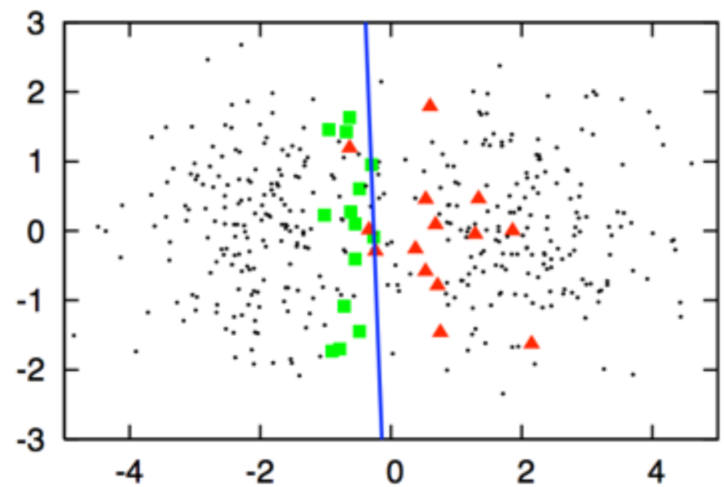
Task Selection

- **Key Idea**
  - Select the most **beneficial** tasks to crowdsource
- **Example 1: Active Learning**
  - Most beneficial for training a model

Supervised Learning



Active Learning



- Mozafari et al. Scaling Up Crowd-Sourcing to Very Large Datasets: A Case for Active Learning. PVLDB 2014
- Gokhale et al. Corleone: hands-off crowdsourcing for entity matching. SIGMOD 2014

Task Selection

- **Key Idea**
  - Select the most **beneficial** tasks to crowdsource
- **Example 2: Top-k**
  - Most beneficial for getting the top-k results

Which picture visualizes the best SFU Campus?

Rank by computers





The most beneficial task:

VS.

Xiaohang Zhang, Guoliang Li, Jianhua Feng: Crowdsourced Top-k Algorithms: An Experimental Evaluation. PVLDB 2016

Task Selection (cont'd)

- **Workflow (iterative)**
  1. Select a set of most beneficial tasks
  2. Collect their answers from the Crowd
  3. Update models and results

- **Pros**
  – Allow for a flexible quality/cost trade-off
- **Cons**
  – Hurt latency (since only a small number of tasks can be crowdsourced at each iteration)

Classification of Existing Techniques

- **How to reduce n?**
  - Task Pruning
  - Answer Deduction
  - Task Selection
  - Sampling

The Database Community

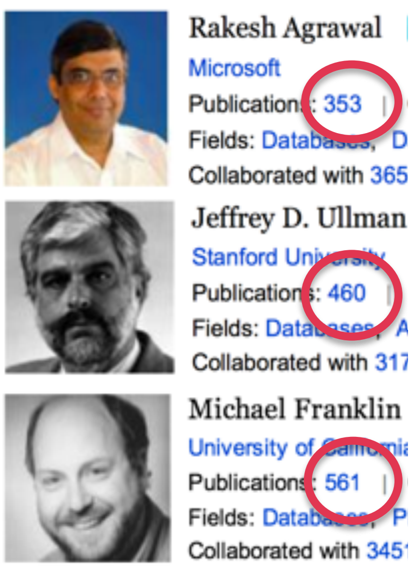
- **How to reduce c?**
  - Task Design

The HCI Community

Sampling

- **Key Idea**
  – Ask the crowd to work on **sample** data
- **Example: SampleClean**

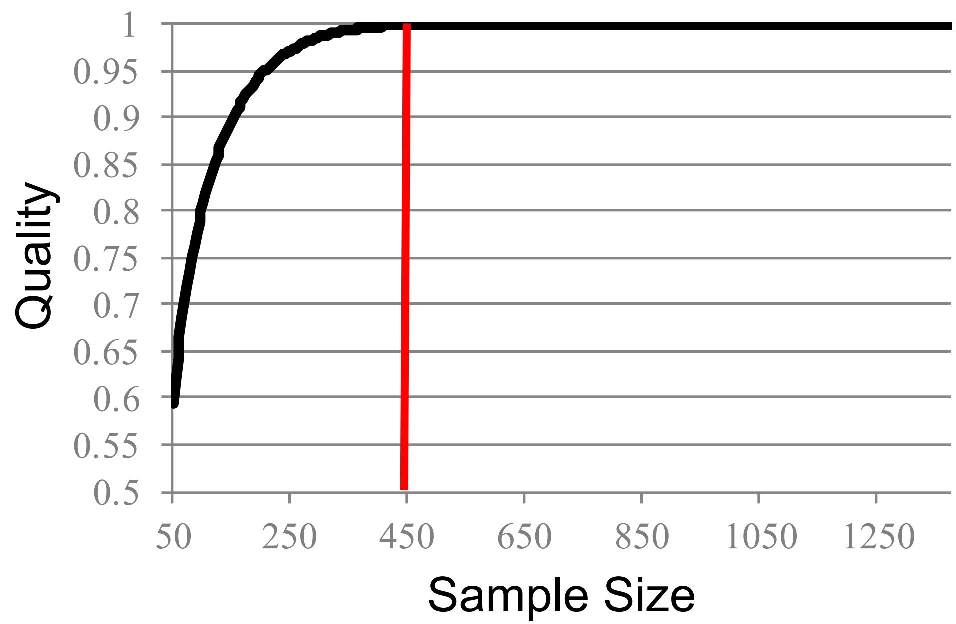
Who published more?



211

255

173



Jiannan Wang, Sanjay Krishnan, Michael J. Franklin, Ken Goldberg, Tim Kraska, Tova Milo: A sample-and-clean framework for fast and accurate query processing on dirty data. SIGMOD Conference 2014: 469-480

Sampling (Cont'd)

- **Workflow (iterative)**
  1. Generate tasks based on a sample
  2. Collect the task answers from the Crowd
  3. Infer the results of the full data

- **Pros**
  - Provable bounds for quality (e.g., the paper count is 211±5 with 95% probability)
- **Cons**
  - Limited to certain applications (e.g., it does not work for max)

Classification of Existing Techniques

- **How to reduce n?**
  - Task Pruning
  - Answer Deduction
  - Task Selection
  - Sampling

The Database Community

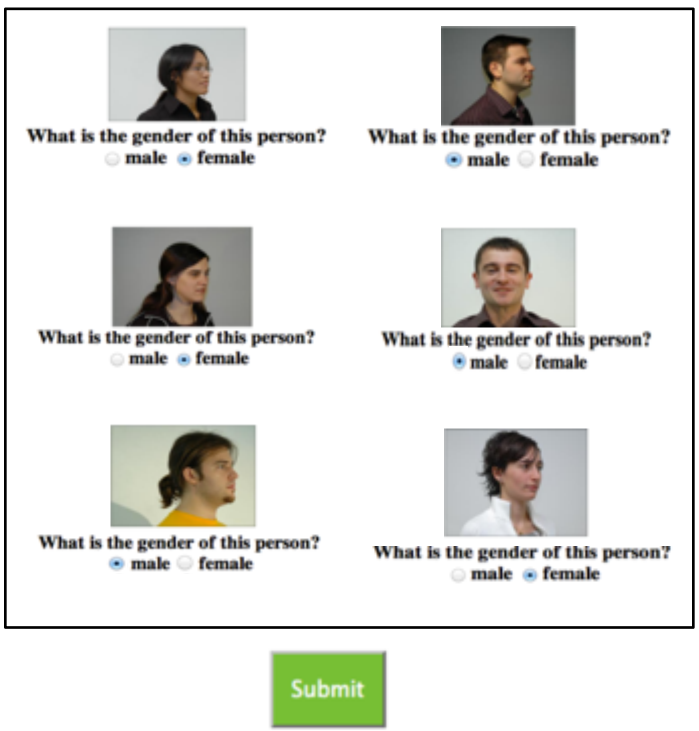
- **How to reduce c?**
  - Task Design

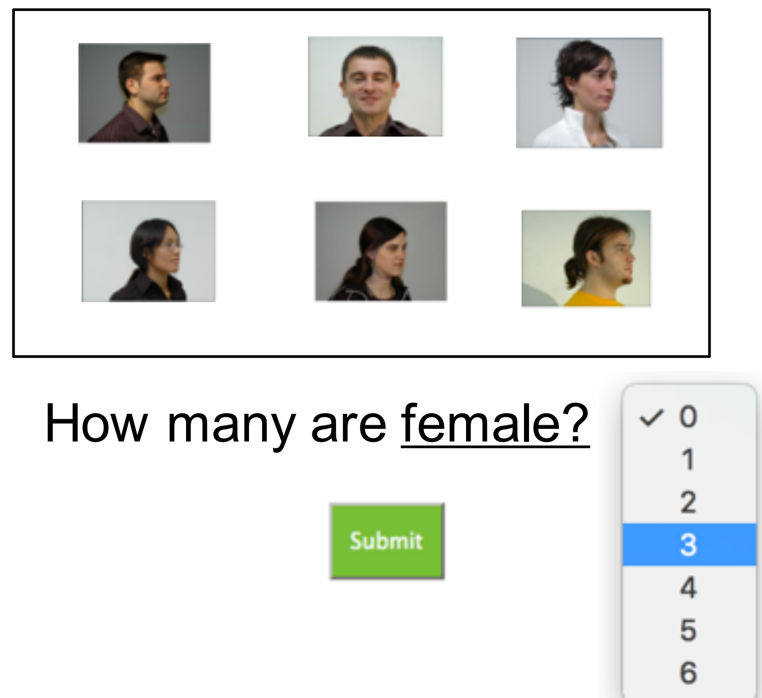
The HCI Community

Task Design (Cont'd)

- **Key Idea**
  - Optimize User Interface

- **Example 1: Count**





Adam Marcus, David R. Karger, Samuel Madden, Rob Miller, Sewoong Oh: Counting with the Crowd. PVLDB :

Task Design (Cont'd)

- **Key Idea**
  - Optimize User Interface

- **Example 2: Image Labeling**

Luis von Ahn, Laura Dabbish: Labeling images with a computer game. CHI 2004: 319-326

Summary of Cost Control

- **Two directions**
  - How to reduce n? ← DB
  - How to reduce c? ← HCI
- **DB and HCI should work together**
- **Non-iterative and iterative workflows are both widely used**

Outline

- Crowdsourcing Overview (30min)
  - Motivation (5min)
  - Workflow (15min)
  - Platforms (5min)
  - Difference from Other Tutorials (5min)
- Fundamental Techniques (100min)
  - Quality Control (60min)
  - Cost Control (20min)
  - Latency Control (20min)
- Crowdsourced Database Management (40min)
  - Crowdsourced Databases (20min)
  - Crowdsourced Optimizations (10min)
  - Crowdsourced Operators (10min)
- Challenges (10min)

Part 1

Part 2

Latency Control

- **Goal**
  - How to reduce latency?

- **Latency = $n \times t$**
  - n: number of tasks
  - t: latency of each task

- **Latency =** The completion time of the last task

Classification of Latency Control

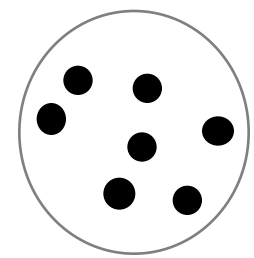
1. Single Task

– Reduce the latency of a single task

Single task

2. Single Batch

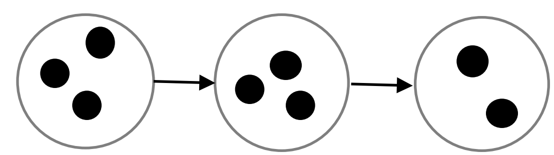
– Reduce the latency of a batch of tasks



Single batch

3. Multiple Batches

– Reduce the latency of multiple batches of tasks



Multiple batches

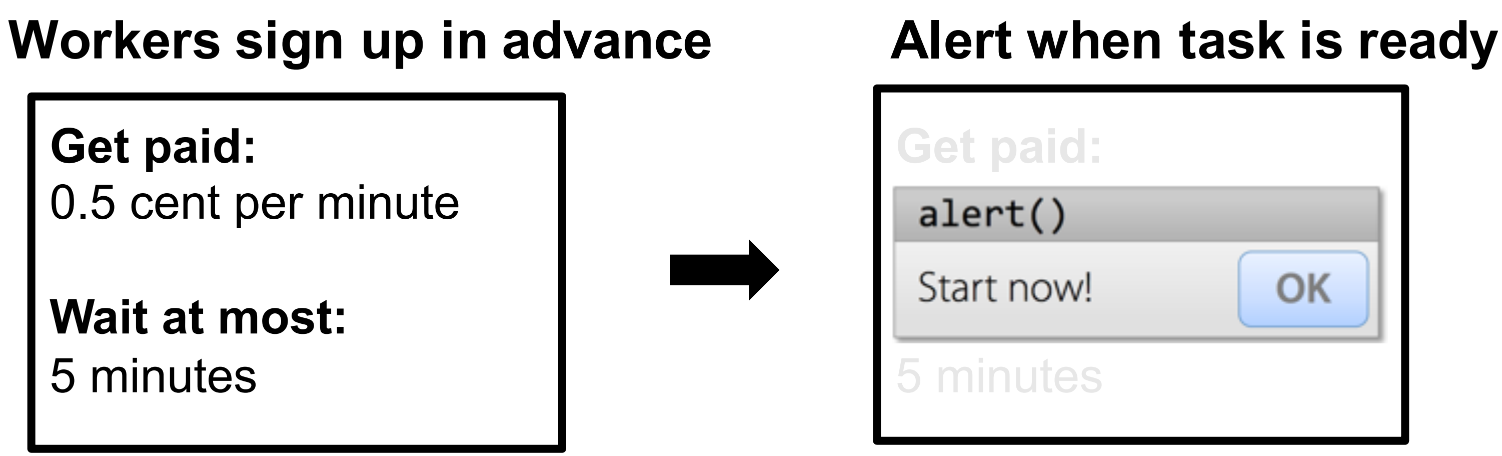
Daniel Haas, Jiannan Wang, Eugene Wu, Michael J. Franklin: CLAMShell: Speeding up Crowds for Low-latency Data Labeling. PVLDB 2015

Single-Task Latency Control

- **Latency consists of**
  - Phase 1: Recruitment Time
  - Phase 2: Qualification and Training Time
  - Phase 3: Work Time
- **Improve Phase 1**
  - See the next slide
- **Improve Phase 2**
  - Remove this phase by applying other quality control techniques (e.g., worker elimination)
- **Improve Phase 3**
  - Better User Interfaces

Reduce Recruitment Time

- **Retainer Pool**
  - Pre-recruit a pool of crowd workers



Michael S. Bernstein, Joel Brandt, Robert C. Miller, David R. Karger: Crowds in two seconds: enabling realtime crowd-powered interfaces. UIST 2011

Classification of Latency Control

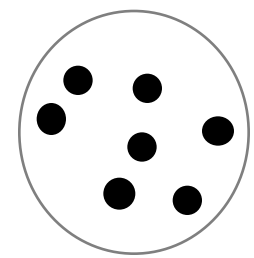
1. Single Task

– Reduce the latency of a single task

Single task

2. Single Batch

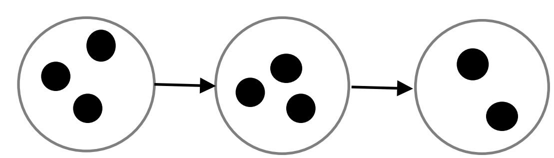
– Reduce the latency of a batch of tasks



Single batch

3. Multiple Batches

– Reduce the latency of multiple batches of tasks



Multiple batches

Daniel Haas, Jiannan Wang, Eugene Wu, Michael J. Franklin: CLAMShell: Speeding up Crowds for Low-latency Data Labeling. PVLDB 2015

Single-Batch Latency Control

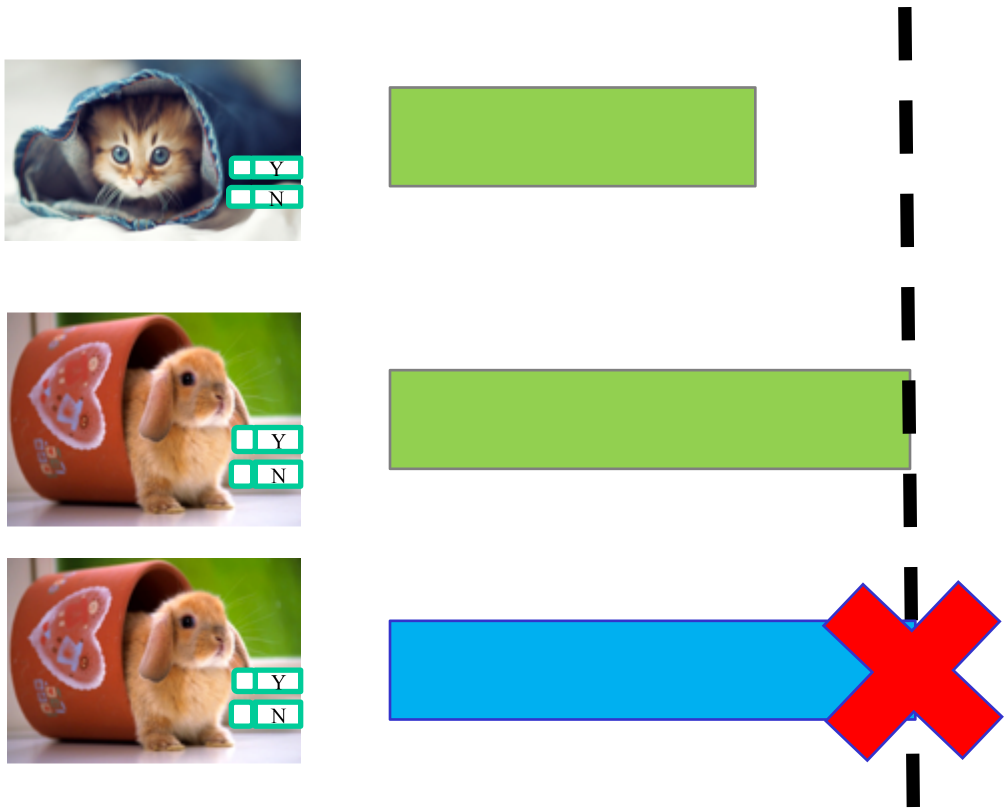
- **Idea 1: Pricing Model**
  - Model the relationship between task price and completion time

- **Predict worker behaviors** [1,2]
  - Recruitment Time
  - Work Time
- **Set task price**
  - Fixed Pricing [2]
  - Dynamic Pricing [3]

[1]. Wang et al. Estimating the completion time of crowdsourced tasks using survival analysis models. CSDM 2011
[2]. S. Faradani, B. Hartmann, and P. G. Ipeirotis. What's the right price? pricing tasks for finishing on time. In AAAI Workshop, 2011.
[3]. Y. Gao and A. G. Parameswaran. Finish them!: Pricing algorithms for human computation. PVLDB 2014.

Single-Batch Latency Control

- **Idea 2: Straggler Mitigation**
  - Replicate a task to multiple workers and return the result of the fastest worker



Straggler mitigation (e.g., MapReduce, Spark)

Daniel Haas, Jiannan Wang, Eugene Wu, Michael J. Franklin: CLAMShell: Speeding up Crowds for Low-latency Data Labeling. PVLDB 2015

Classification of Latency Control

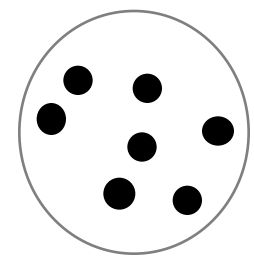
1. Single Task

– Reduce the latency of a single task

Single task

2. Single Batch

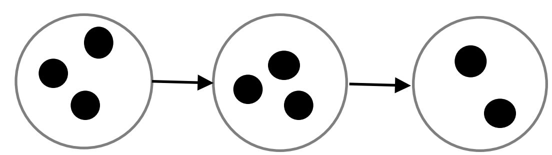
– Reduce the latency of a batch of tasks



Single batch

3. Multiple Batches

– Reduce the latency of multiple batches of tasks

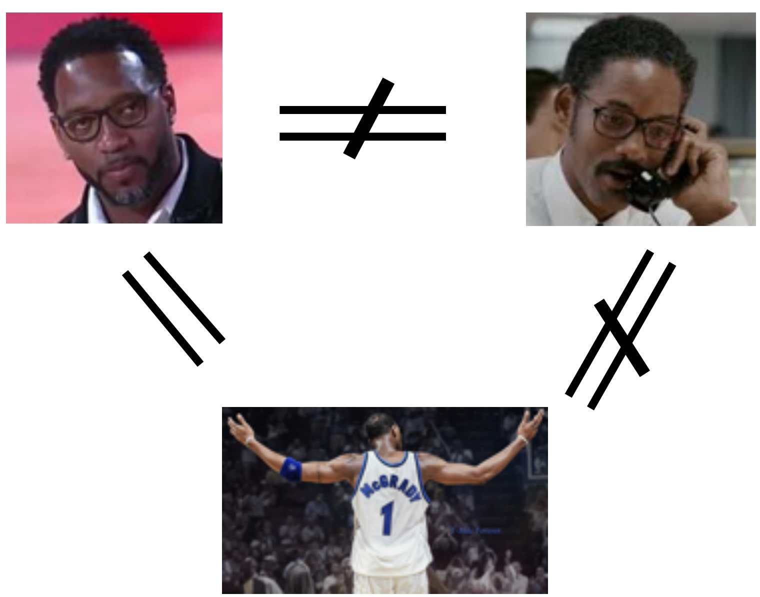


Multiple batches

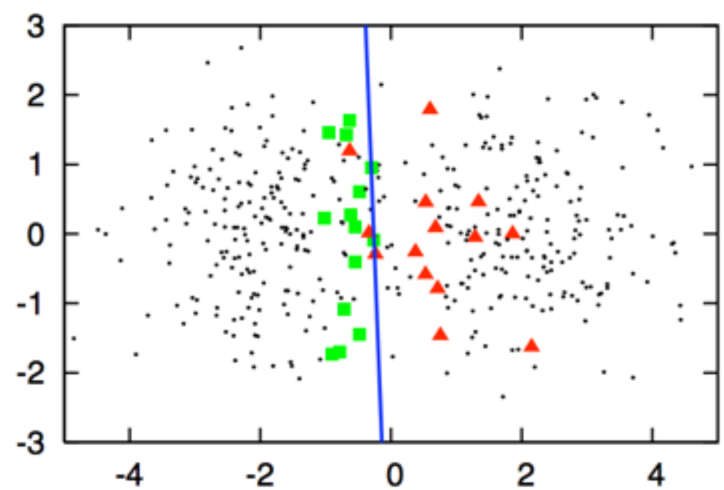
Daniel Haas, Jiannan Wang, Eugene Wu, Michael J. Franklin: CLAMShell: Speeding up Crowds for Low-latency Data Labeling. PVLDB 2015

Multiple-Batches Latency Control

- **Why multiple batches?**
  - To save cost
    - Answer Deduction (e.g., leverage transitivity)
    - Task Selection (e.g., active learning)



Active Learning



Multiple-Batches Latency Control

- **Two extreme cases**
  - Single task per batch: high latency
  - All tasks in one batch: high cost

- **Idea 1**
  - Choose the maximum batch size that does not hurt cost [1,2]
- **Idea 2**
  - Model as a latency budget allocation problem [3]

1. Jiannan Wang, Guoliang Li, Tim Kraska, Michael J. Franklin, Jianhua Feng: Leveraging transitive relations for crowdsourced joins. SIGMOD 2013
2. D. Sarma, A. G. Parameswaran, H. Garcia-Molina, and A. Y. Halevy. Crowd-powered find algorithms. ICDE 2014.
3. Verroios et al.. tdp: An optimal latency budget allocation strategy for crowdsourced MAXIMUM operations. SIGMOD 2015

Summary of Latency Control

- **Latency**
  - The completion time of the last task

- **Classification of Latency Control**
  - Single-Task
    - Retainer Pool
    - Better UIs
  - Single-Batch
    - Pricing Model
    - Straggler Mitigation
  - Multiple-Batches
    - Batch size

Two Take-Away Messages

- **There is no free lunch**
  - Cost control
    - Trades off quality (or/and latency) for cost
  - Latency control
    - Trades off quality (or/and cost) for latency

- **Learn from other communities**
  - Task Design (from HCI)
  - Straggler Mitigation (from Distributed System)

Reference – Cost Control

1. Y. Amsterdamer, S. B. Davidson, T. Milo, S. Novgorodov, and A. Somech. Oassis: query driven crowd mining. In SIGMOD, pages 589–600. ACM, 2014
2. X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In WSDM, pages 193–202, 2013
3. G. Demartini, D. E. Difallah, and P. Cudre-Mauroux. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In WWW, pages 469–478, 2012.
4. B. Eriksson. Learning to top-k search using pairwise comparisons. In AISTATS, pages 265–273, 2013.
5. C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. W. Shavlik, and X. Zhu. Corleone: hands-off crowdsourcing for entity matching. In SIGMOD, pages 601–612, 2014.
6. A. Gruenheid, D. Kossmann, S. Ramesh, and F. Widmer. Crowdsourcing entity resolution: When is A=B? Technical report, ETH Zurich.
7. S. Guo, A. G. Parameswaran, and H. Garcia-Molina. So who won?: dynamic max discovery with the crowd. In SIGMOD, pages 385–396, 2012.
8. H. Heikinheimo and A. Ukkonen. The crowd-median algorithm. In HCOMP, 2013.
9. S. R. Jeffery, M. J. Franklin, and A. Y. Halevy. Pay-as-you-go user feedback for dataspace systems. In SIGMOD, pages 847–860, 2008.
10. H. Kaplan, I. Lotosh, T. Milo, and S. Novgorodov. Answering planning queries with the crowd. PVLDB, 6(9):697–708, 2013.
11. A. R. Khan and H. Garcia-Molina. Hybrid strategies for finding the max with the crowd. Technical report, 2014.
12. A. Marcus, D. R. Karger, S. Madden, R. Miller, and S. Oh. Counting with the crowd. PVLDB, 6(2):109–120, 2012.
13. B. Mozafari, P. Sarkar, M. Franklin, M. Jordan, and S. Madden. Scaling up crowd-sourcing to very large datasets: a case for active learning. PVLDB, 8(2):125–136, 2014.
14. A. G. Parameswaran, A. D. Sarma, H. Garcia-Molina, N. Polyzotis, and J. Widom. Human-assisted graph search: it's okay to ask questions. PVLDB, 4(5):267–278, 2011.

Reference – Cost Control

15. T. Pfeiffer, X. A. Gao, Y. Chen, A. Mao, and D. G. Rand. Adaptive polling for information aggregation. In AAAI, 2012.
16. B. Trushkowsky, T. Kraska, M. J. Franklin, and P. Sarkar. Crowdsourced enumeration queries. In ICDE, pages 673–684, 2013.
17. V. Verroios and H. Garcia-Molina. Entity resolution with crowd errors. In ICDE, pages 219–230, 2015.
18. N. Vesdapunt, K. Bellare, and N. N. Dalvi. Crowdsourcing algorithms for entity resolution. PVLDB, 7(12):1071–1082, 2014.
19. J. Wang, T. Kraska, M. J. Franklin, and J. Feng. CrowdER: crowdsourcing entity resolution. PVLDB, 5(11):1483–1494, 2012.
20. J. Wang, S. Krishnan, M. J. Franklin, K. Goldberg, T. Kraska, and T. Milo. A sample-and-clean framework for fast and accurate query processing on dirty data. In SIGMOD, pages 469–480, 2014.
21. J. Wang, G. Li, T. Kraska, M. J. Franklin, and J. Feng. Leveraging transitive relations for crowdsourced joins. In SIGMOD, 2013.
22. S. Wang, X. Xiao, and C. Lee. Crowd-based deduplication: An adaptive approach. In SIGMOD, pages 1263–1277, 2015.
23. S. E. Whang, P. Lofgren, and H. Garcia-Molina. Question selection for crowd entity resolution. PVLDB, 6(6):349–360, 2013.
24. T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In MobiSys, pages 77–90, 2010.
25. P. Ye, U. EDU, and D. Doermann. Combining preference and absolute judgements in a crowd-sourced setting. In ICML Workshop, 2013.
26. C. J. Zhang, Y. Tong, and L. Chen. Where to: Crowd-aided path selection. PVLDB, 7(14):2005–2016, 2014.

Reference – Latency Control

1. J. P. Bigham et al. VizWiz: nearly real-time answers to visual questions. UIST, 2010.
2. M. S. Bernstein, J. Brandt, R. C. Miller, and D. R. Karger. Crowds in two seconds: enabling realtime crowd-powered interfaces. UIST, 2011.
3. M. S. Bernstein, D. R. Karger, R. C. Miller, and J. Brandt. Analytic Methods for Optimizing Realtime Crowdsourcing. Collective Intelligence, 2012.
4. Y. Gao and A. G. Parameswaran. Finish them!: Pricing algorithms for human computation. PVLDB, 7(14):1965–1976, 2014
5. S. Faradani, B. Hartmann, and P. G. Ipeirotis. What's the right price? pricing tasks for finishing on time. In AAAI Workshop, 2011.
6. D. Haas, J. Wang, E. Wu, and M. J. Franklin. Clamshell: Speeding up crowds for low-latency data labeling. PVLDB, 9(4):372–383, 2015
7. A. D. Sarma, A. G. Parameswaran, H. Garcia-Molina, and A. Y. Halevy. Crowd-powered find algorithms. In ICDE, pages 964–975, 2014
8. V. Verroios, P. Lofgren, and H. Garcia-Molina. tdp: An optimal-latency budget allocation strategy for crowdsourced MAXIMUM operations. In SIGMOD, pages 1047–1062, 2015.
9. T. Yan, V. Kumar, and D. Ganesan. Crowdsearch: exploiting crowds for accurate real-time image search on mobile phones. In MobiSys, pages 77–90, 2010.

Outline

- Crowdsourcing Overview (30min)
  - Motivation (5min)
  - Workflow (15min)
  - Platforms (5min)
  - Difference from Other Tutorials (5min)
- Fundamental Techniques (100min)
  - Quality Control (60min)
  - Cost Control (20min)
  - Latency Control (20min)
- Crowdsourced Database Management (40min)
  - Crowdsourced Databases (20min)
  - Crowdsourced Optimizations (10min)
  - Crowdsourced Operators (10min)
- Challenges (10min)

Part 1

Part 2

Why Crowdsourcing DB Systems

- Limitations of Traditional DB Systems

Table: car

make	model	body_style	price
Volve	S80	Sedan	$10K
Volve	XC60	SUV	$20K
BMW	X5	SUV	$25K
?	Prius	Sedan	$15K

```
SELECT   *
FROM     car
WHERE    make = "Toyota"
```

# of rows

0

Problem: Close world assumption

Why Crowdsourcing DB Systems

○ Limitations of Traditional DB Systems

Table: car_image

m_1 m_2 m_3 m_4 m_5

```
SELECT   *
FROM     car C, car_image M
WHERE    M.make = C.make AND
         M.model = C.model AND
         M.color = "red"
```

Table: car

make	model	body_style	price
XXX	XXX	XXX	XXX
XXX	XXX	XXX	XXX
……	……	……	……

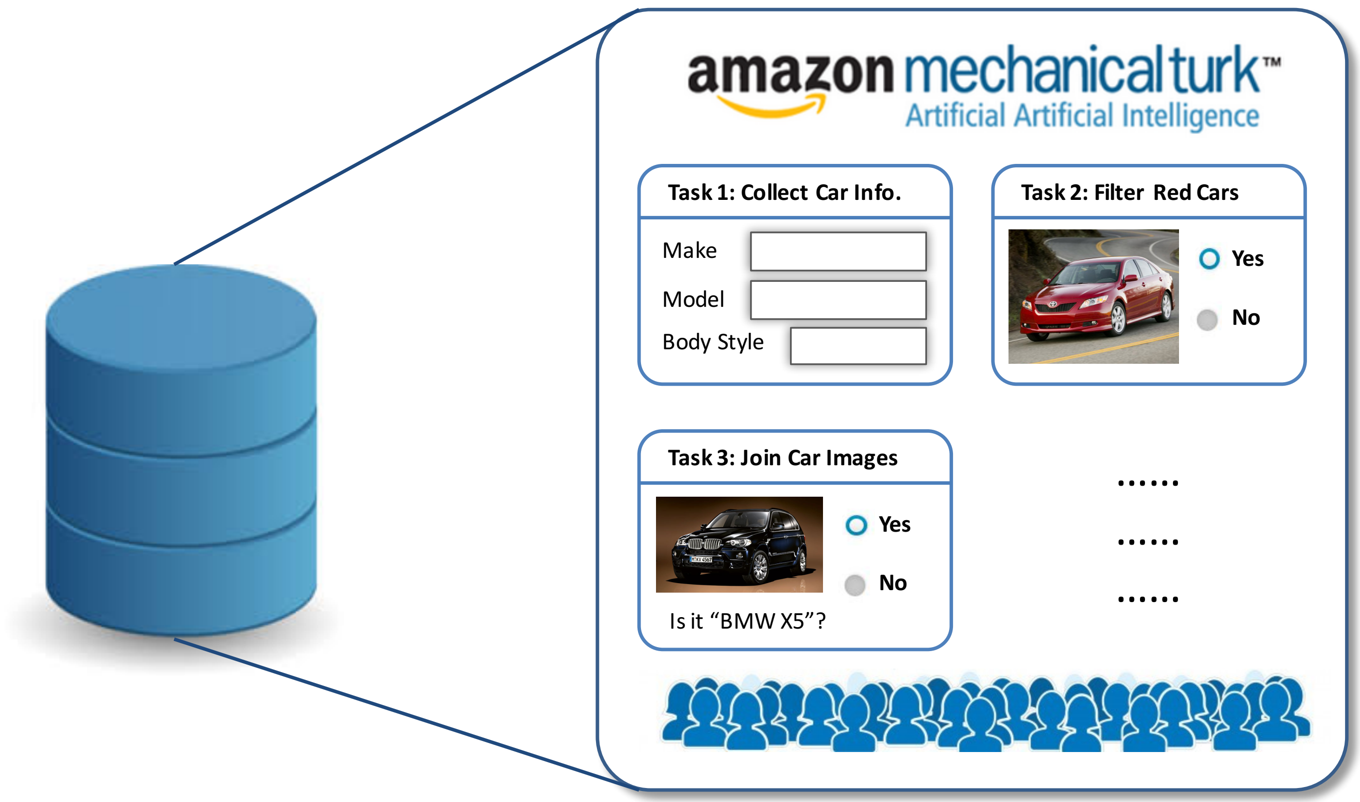
\# of rows

0

Problem: Machine-hard tasks

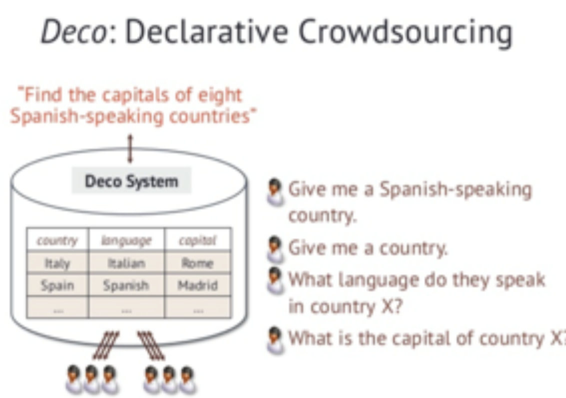
Crowdsourcing DB Systems

- **Integrating crowd functionality to DB**
  - Close world → Open world
  - Processing DB-hard queries

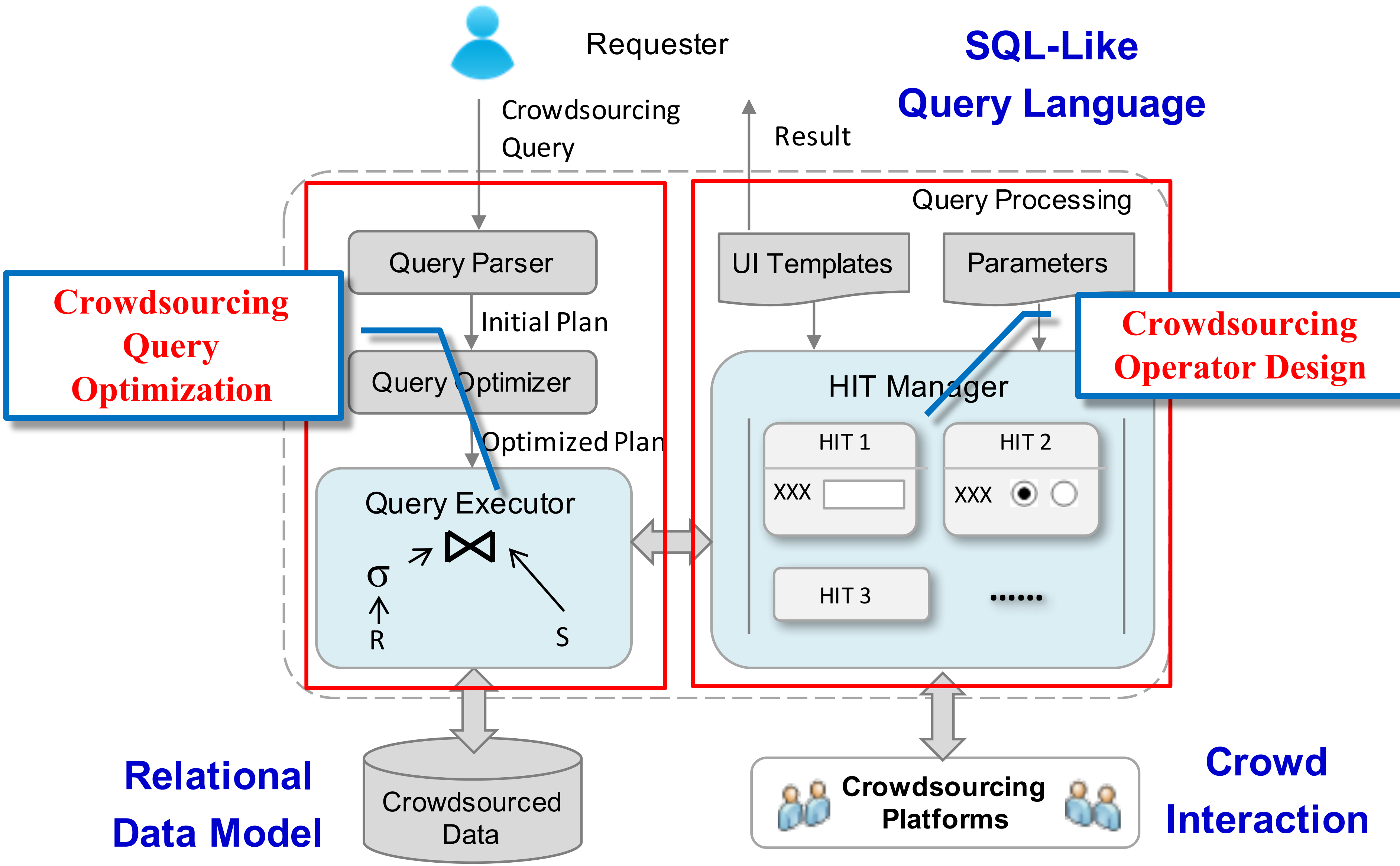


Existing Crowd DB Systems

- **CrowdDB**
  - UC Berkeley & ETH Zurich
- **Qurk**
  - MIT
- **Deco**
  - Stanford
- **CDAS**
  - NUS
- **CDB**
  - Tsinghua



System Architecture



Running Example

car_review R1

review	make	model	sentiment
r_1	...The 2014 **Volvo S80** is the flagship model for the brand...		
r_2	...**S80** is a **Volvo** model having problems in oil pump..		
r_3	...The **BMW X5** is surprisingly agile for a big SUV..		

car R2

id	make	model	style
a_1	Volvo	S80	Sedan
a_2	Toyota	Avalon	Sedan
a_3	Volvo	XC60	SUV
a_4	Toyota	Corolla	Sedan
a_5	BMW	X5	SUV
a_6	Toyota	Camry	Sedan

car_image R3

m_1

m_2

m_3

m_4

m_5

Example Query:

Find **black** **cars** with **high-quality** **images** and **positive** **reviews**

Crowdsourcing DB Systems

- **System Overview**
  - CrowdDB
  - Quirk
  - Deco
  - CDAS
  - CDB

Crowdsourcing Systems

- **Operator Design**
  - Design Principles

Crowdsourcing Operators

CrowdDB Query Language

- **CrowdSQL: Crowdsource missing data**

Missing Columns

review	make	model	sentiment
xxx	Volvo	S80	?

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

Missing Tuples

make	model	style	color
?	?	?	?

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

CrowdDB Query Language

- CrowdSQL: Crowdsource DB-hard tasks

Crowd-powered Filtering

The Vovlo S80 is the flagship model of this brand…

Is the review positive?

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

Crowd-Powered Ordering

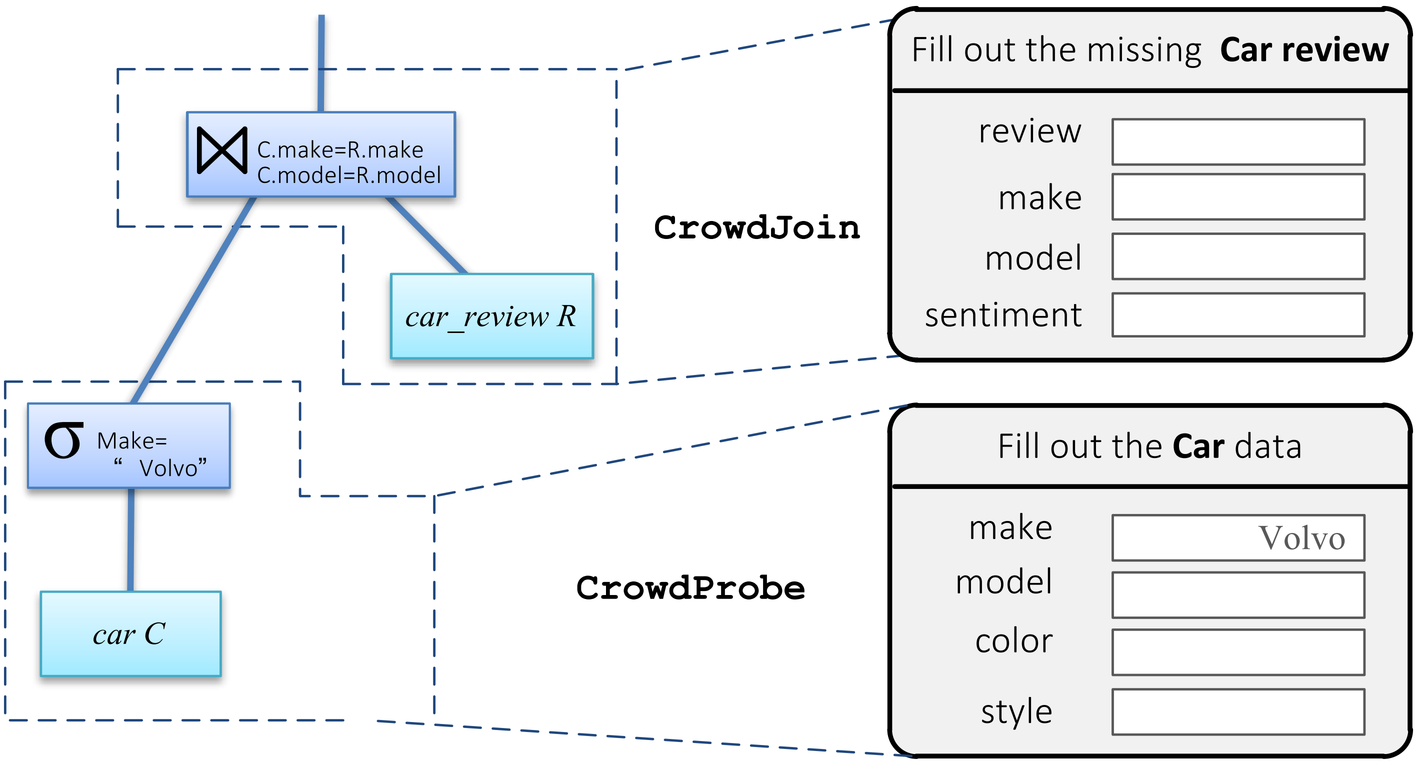
Which one is better?

```
SELECT image i
FROM car_image
WHERE subject = "Volvo S60"
ORDER BY CROWDORDER(”clarity");
```

CrowdDB Query Processing

- Crowd operators for data missing

```
SELECT *
FROM car C, car_review R
WHERE C.make = R.make AND C.model = R.model AND
      C.make = "Volvo"
```

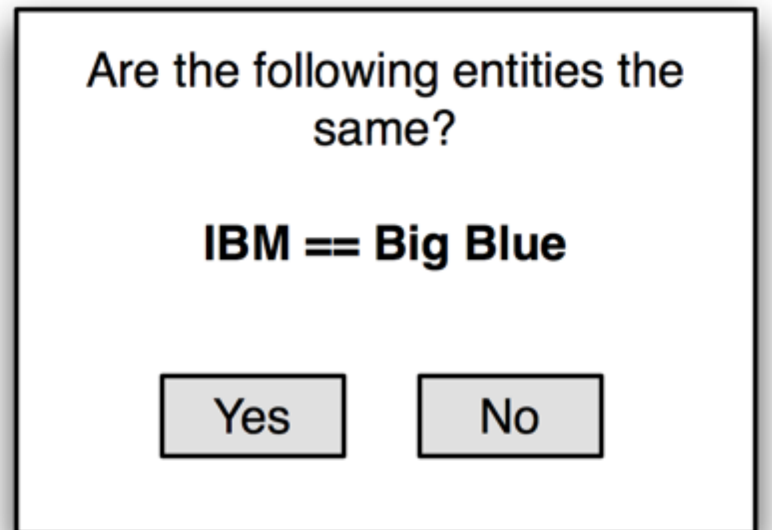


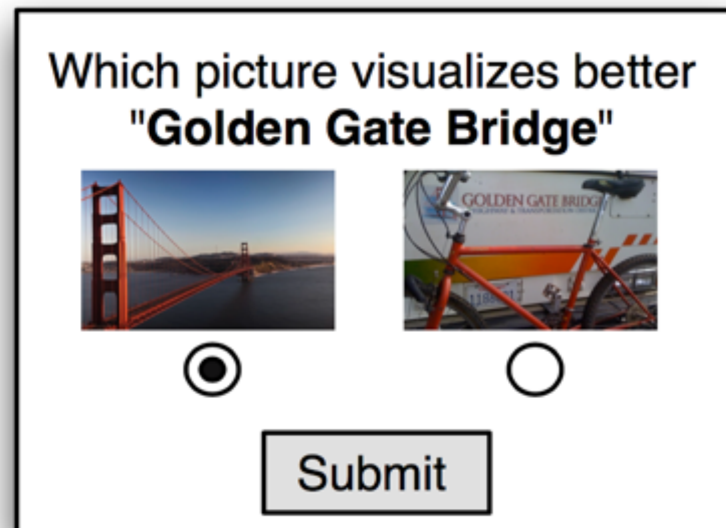
CrowdDB Query Processing

- Crowd operators for DB-hard tasks

```
SELECT *
FROM  company C1, company C2
WHERE C1.name ~= C2.name
```

```
SELECT *
FROM  image M
ORDER BY CROWDORDER ("clarity")
```

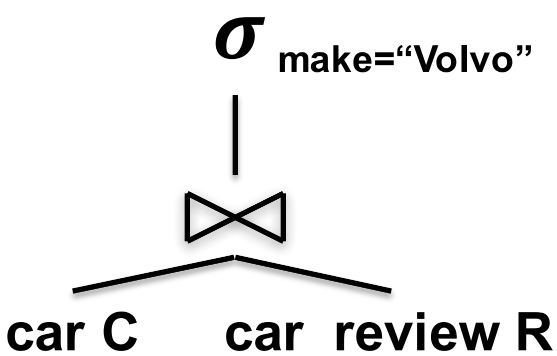


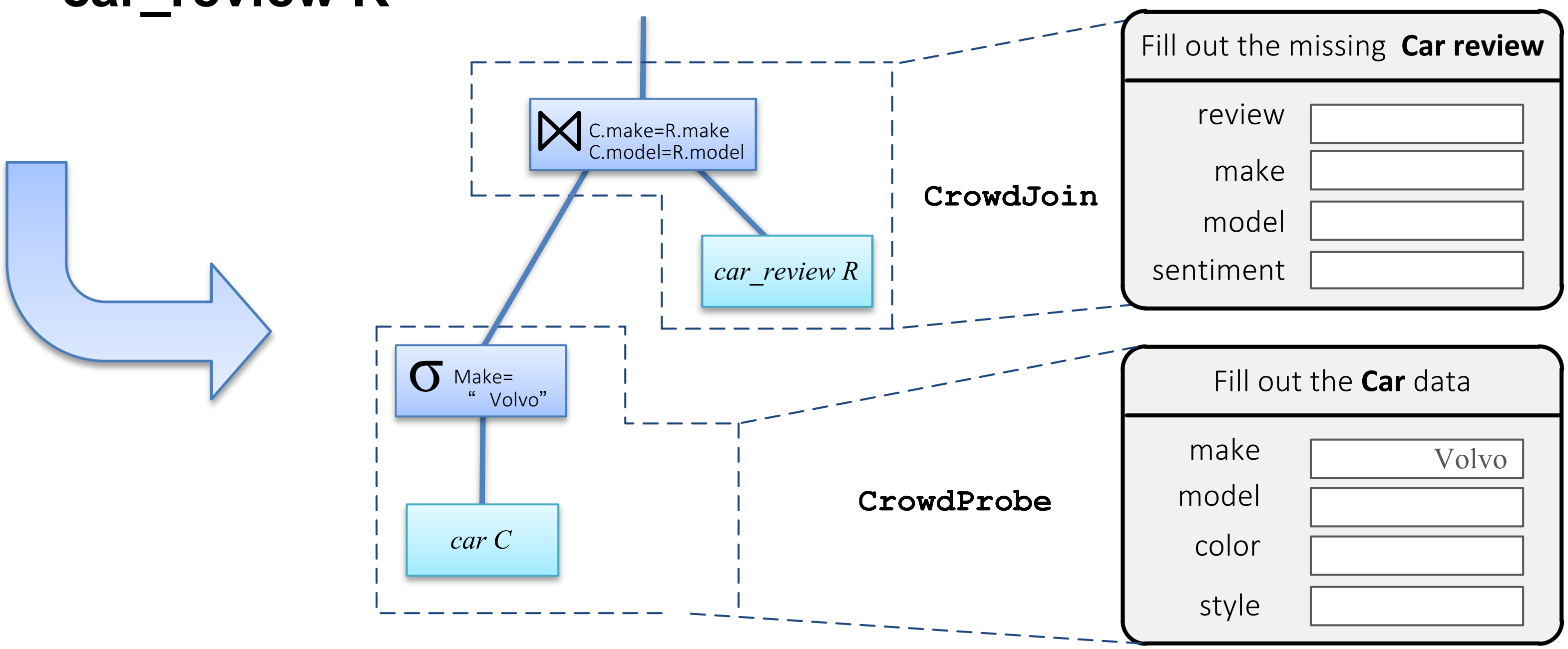


CrowdCompare

CrowdDB Query Optimization

- **Strategy: Rule-based optimizer**
  - Pushing down selects
  - Determining join orders





Crowdsourcing DB Systems

- **System Overview**
  - CrowdDB
  - Qurk
  - Deco
  - CDAS
  - CDB

Crowdsourcing Systems

- **Operator Design**
  - Design Principles

Crowdsourcing Operators

Qurk Query Language

- **SQL with User-Defined Functions (UDFs)**

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

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

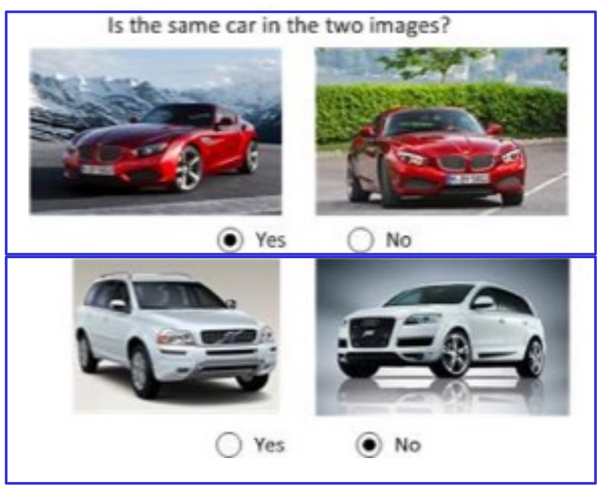
Is the car in black color?

Yes No

Qurk Query Processing

- **Designing crowd-powered operators**
  - Crowd Join: Designing better interfaces

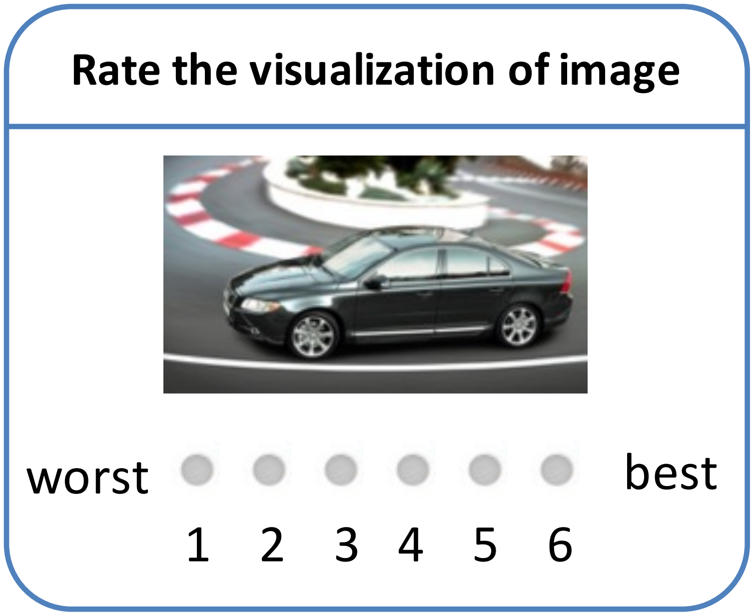
Simple Join



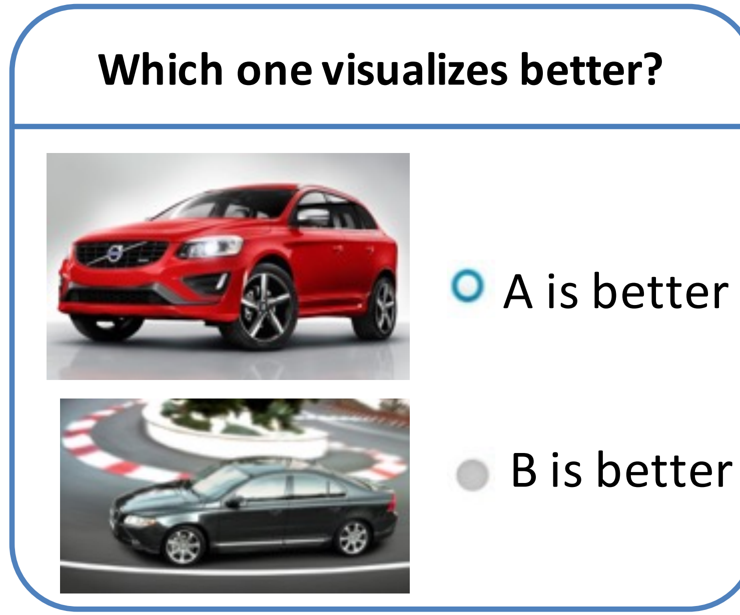
Naïve Batching **Smart Batching**

Qurk Query Processing

- **Designing crowd-powered operators**
  - Crowd Sort: Designing better interfaces



Rating-Based Interface



Comparing-Based Interface

Qurk Query Optimization

- **Join: Feature filtering optimization**

```
SELECT *
FROM car_image M1 JOIN car_image M2
ON sameCar(M1.img, M2.img) AND
POSSIBLY make(M1.img) = make(M2.img) AND
POSSIBLY style(M1.img) = style(M2.img)
```

Filtering pairs with different makes & colors

- **Is filtering feature always helpful?**
  - Filtering cost vs. join cost
    - What if all cars has the same style
  - Causing false negatives, e.g., color
  - Disagreement among the crowd

Crowdsourcing DB Systems

- **System Overview**
  - CrowdDB
  - Quirk
  - Deco
  - CDAS
  - CDB
- **Operator Design**
  - Design Principles

Crowdsourcing Systems

Crowdsourcing Operators

Deco Query Language

- **Conceptual Relation**

$$\text{Car (}\underbrace{\text{make, model,}}_{\text{Anchor Attributes}}\ \underbrace{\text{[door-num], [style]}}_{\text{Dependent Attribute-groups}}\text{)}$$

- **Raw Schema**

```
CarA ( make, model)              // Anchor table
CarD1 ( make, model, door-num) //Dependent table
CarD2 ( make, model, style) // Dependent table
```

- **Fetch Rules: How to collect data**

```
∅ ⇒ make, model  //Ask for a new car
make, model ⇒ door-num//Ask for d-n of a given car
make, model ⇒ style //Ask for style of a given car
```

Deco Query Language

- **Resolution rules**

```
image ⇒ style: majority-of-3  // majority vote
∅ ⇒ make,model: dupElim  //eliminate duplicates
```

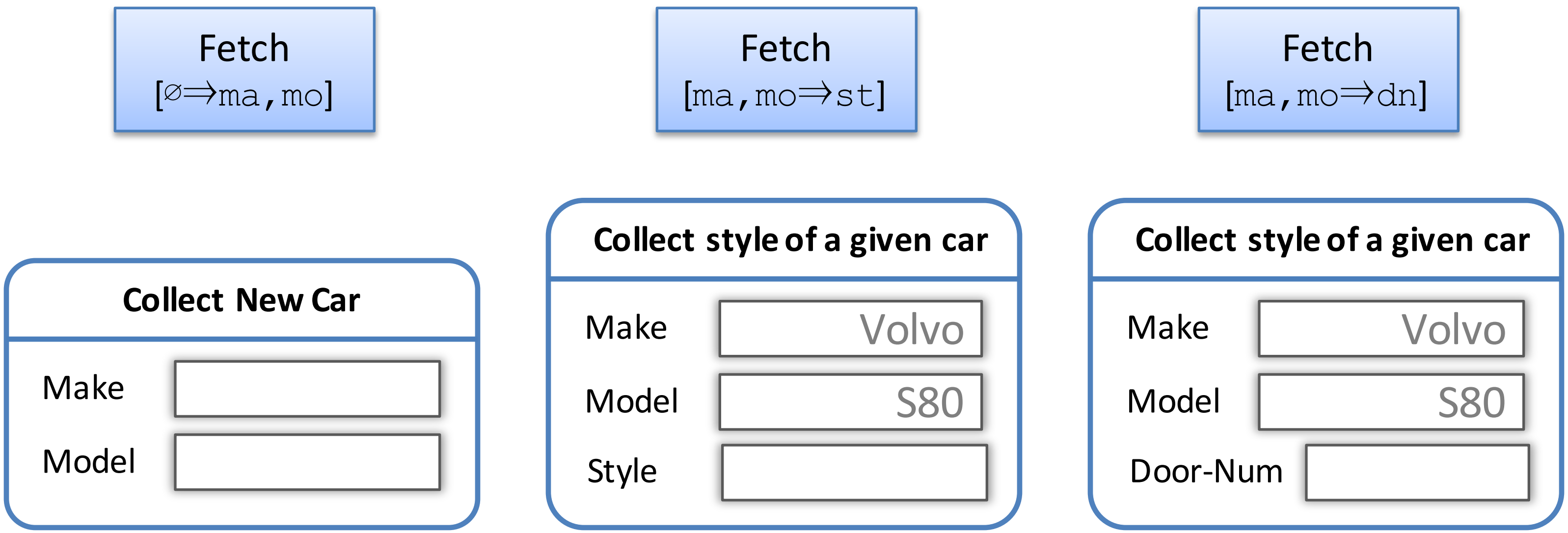
- **Query**
  - Collecting style and color of at least 8 SUV cars
  - SQL Query:

```
SELECT make, model, door-num, style
FROM Car
WHERE style = "SUV" MINTUPLES 8
```

    - Standard SQL Syntax and Semantics
    - New keyword: MINTUPLES

Deco Query Processing

- **Crowd Operator: Fetch**



- **Machine Operators**
  - Scan: insert a collected tuple into raw table
  - Resolve: e.g., `majority-of-3, dupElim`
  - DLOJoin: traditional join

Deco Query Optimization

- **Example**
  - Current Status of the database

CarA

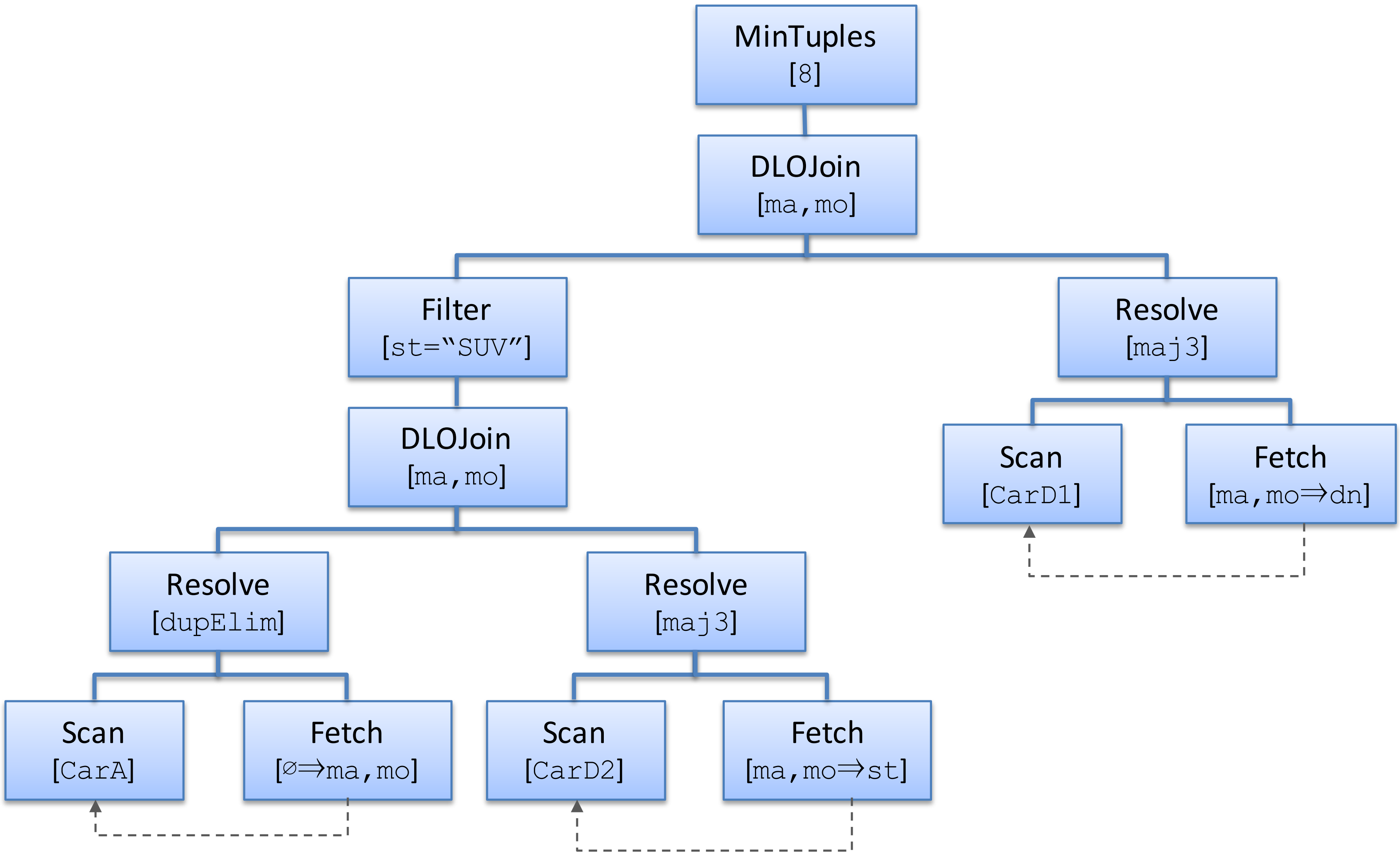
make	model
Volve	S80
Toyota	Corolla
BMW	X5
Volvo	XC60

CarD2

make	model	Style
Volve	XC60	SUV
BMW	X5	SUV
Volvo	S80	Sedan

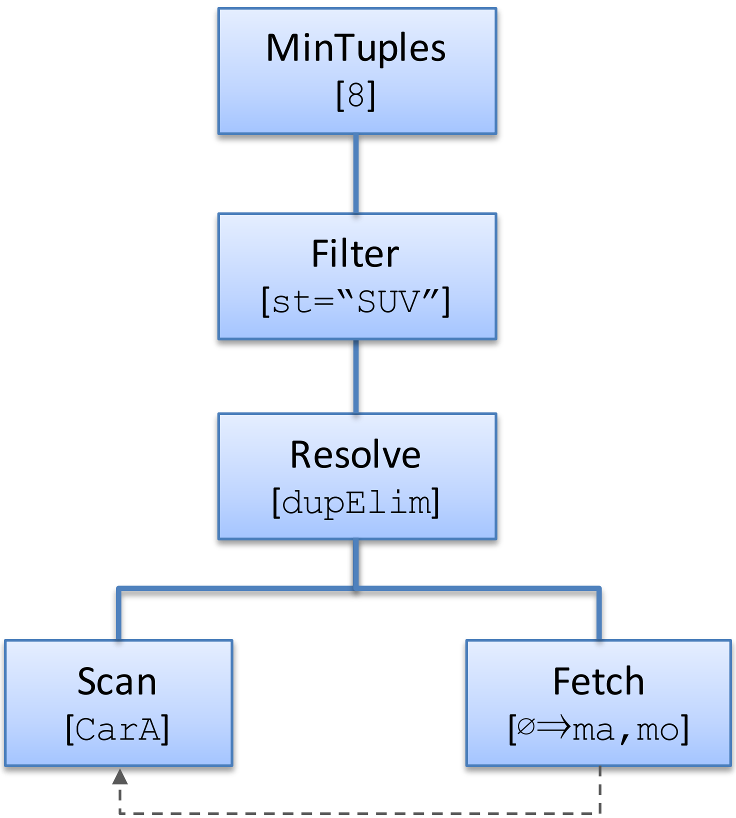
  - Selectivity of [style=‘SUV’] = 0.1
  - Selectivity of dupElim = 1.0
  - Each fetch incurs $0.05
- **How will a query be evaluated?**

Deco Query Processing



Deco Query Optimization

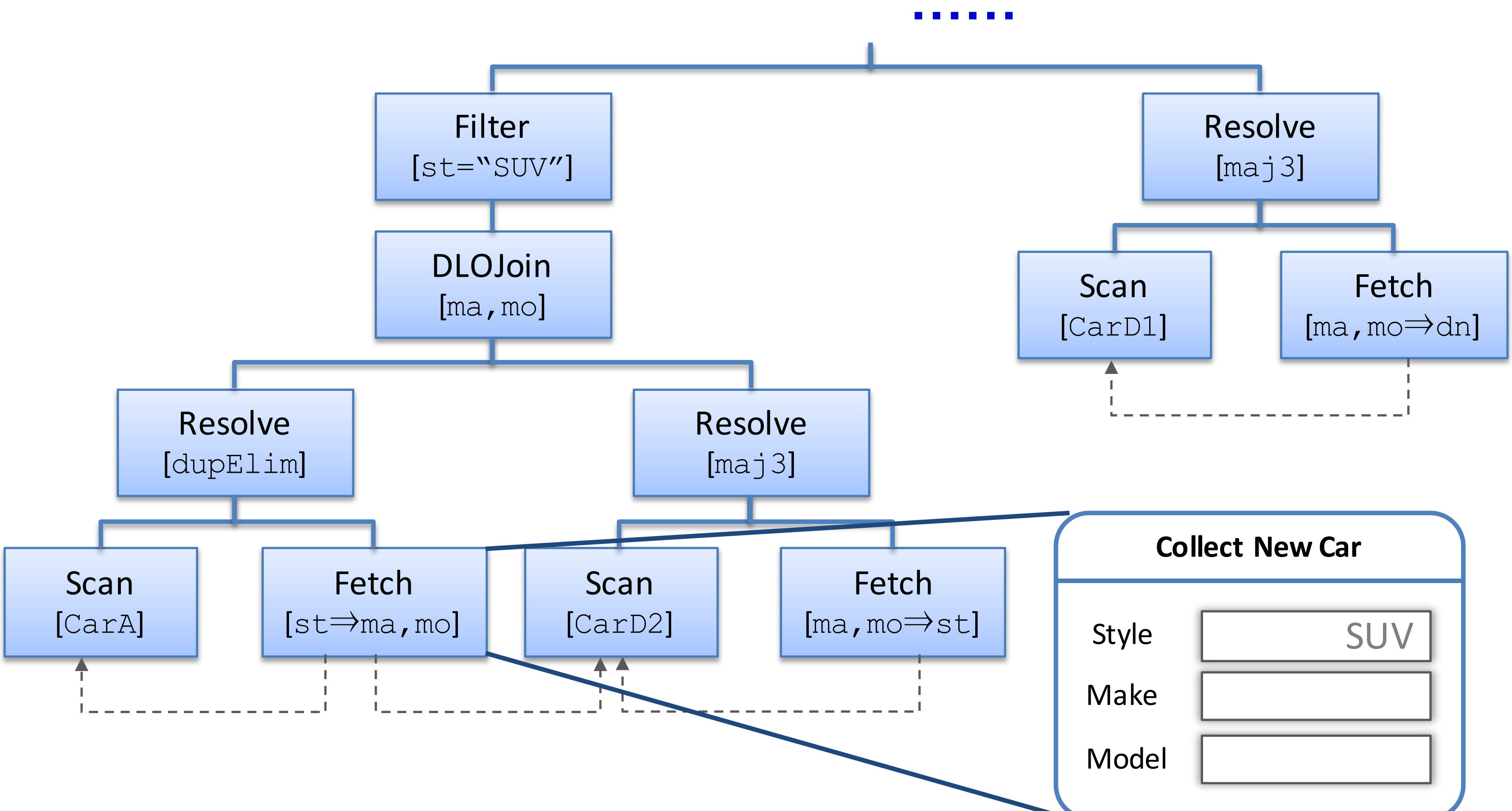
- **Cost Estimation**
  - Let us consider a simple case



- Resolve [dupElim]
  - Target: 8 SUV cars
  - DB: 2 SUV cars, 1 Sedan car, and 1 unknown car
  - Estimated: 2.1 SUV
- Fetch
  - Target: (8 – 2.1) SUV cars
  - Sel [style='SUV'] = 0.1
  - Fetch 59 cars
- Cost: 59 * $0.05 = $2.95

Deco Query Optimization

- Better Plan: Reverse Query Plan



Reverse Plan incurs less cost in this query

Crowdsourcing DB Systems

- **System Overview**
  - CrowdDB
  - Qurk
  - Deco
  - CDAS
  - CDB

Crowdsourcing Systems

- **Operator Design**
  - Design Principles

Crowdsourcing Operators

CDAS Query Language

- **SQL with Crowdsourcing on demand**
  – Crowdsourcing when columns are unknown

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

Is the review matching with the image?

The Vovlo S80 is the flagship model of this brand…

Is the review positive?

Is the car in black?

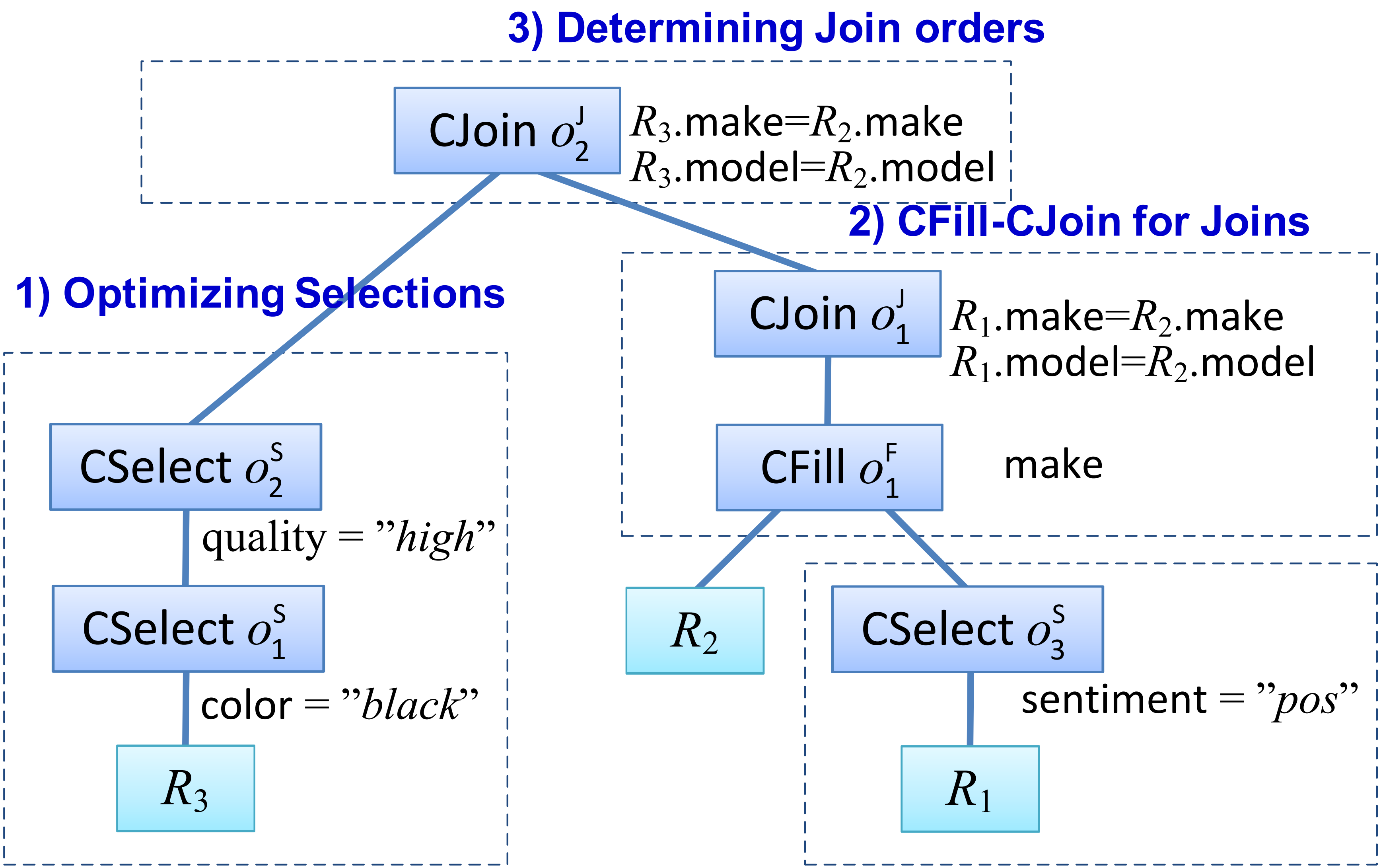
CDAS Query Processing

- **Designing Crowd Operators**
  - CrowdFill: filling missing values
  - CrowdSelect: filtering items
  - CrowdJoin: matching items from multiple sources

CDAS Query Processing

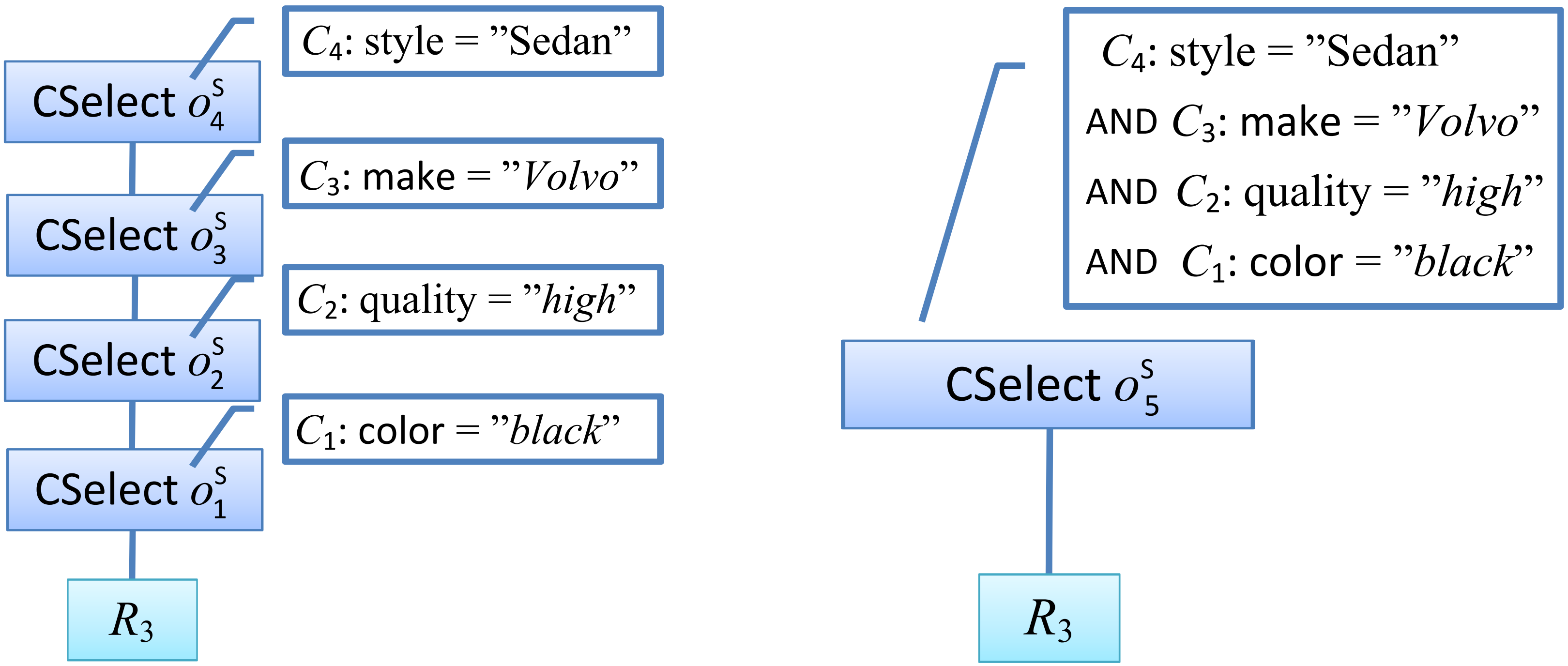
- **Performance metrics**
  - Monetary cost: Unit price * # of HITs
  - Latency: # of crowdsourcing rounds
- **Optimization Objectives:**
  - Cost Minimization: finding a query plan minimizing the monetary cost
  - Cost Bounded Latency Minimization: finding a query plan with bounded cost and the minimum latency
- **Key Optimization Idea**
  - Cost-based query optimization
  - Balance the tradeoff between cost and latency

CDAS Query Processing



CDAS Query Optimization

- **Cost-Latency Tradeoff**



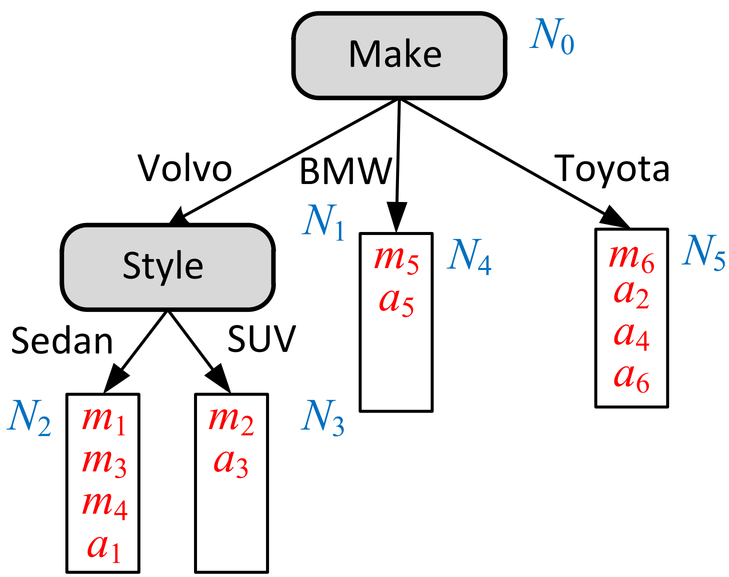
Less cost, higher latency **More cost, lower latency**

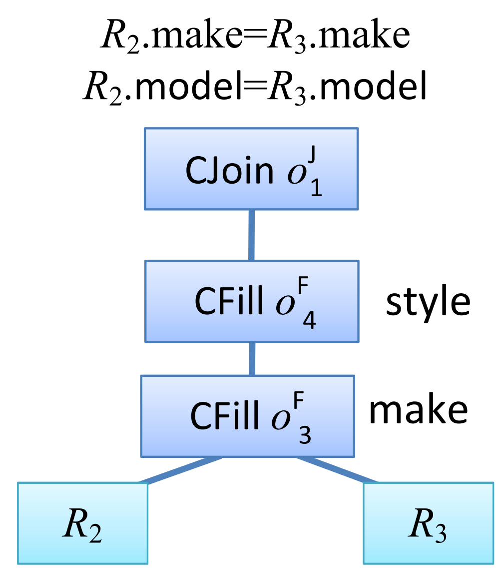
How to balance cost-latency tradeoff?

CDAS Query Optimization

- How to implement Join
  - CJoin: Compare every pairs
  - CFill: Fill missing join attributes
- A Hybrid CFill-CJoin Optimization

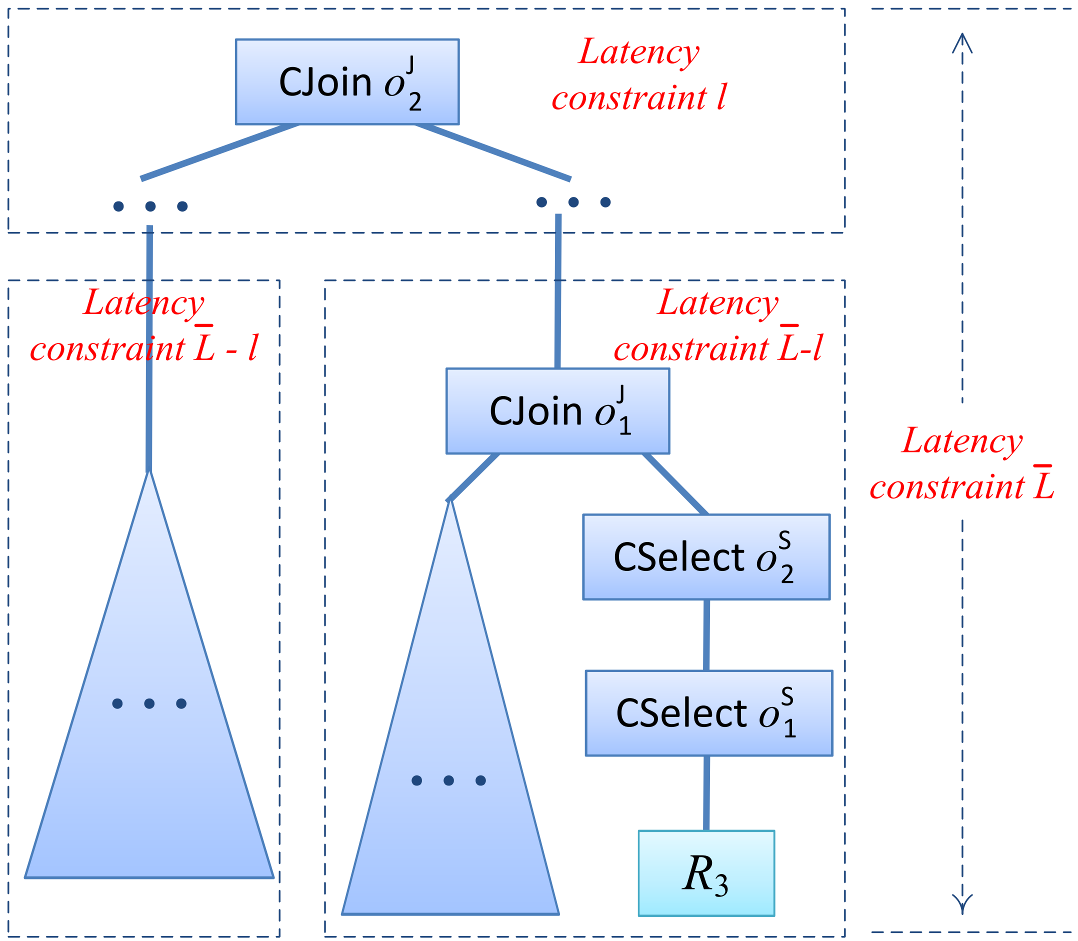
```
SELECT * FROM car R2, car_image R3
WHERE R2.make = R3.make AND R2.model = R3.model
```





CDAS Query Optimization

- **Complex query optimization**
  - The latency constraint allocation problem



Crowdsourcing DB Systems

- **System Overview**
  - CrowdDB
  - Qurk
  - Deco
  - CDAS
  - CDB

Crowdsourcing Systems

- **Operator Design**
  - Design Principles

Crowdsourcing Operators

CDB Query Language

- Collect Semantics
  - Fill Semantics

```
FILL car_image.color
WHERE car_image.make = "Volvo";
```

  - Collect Semantics

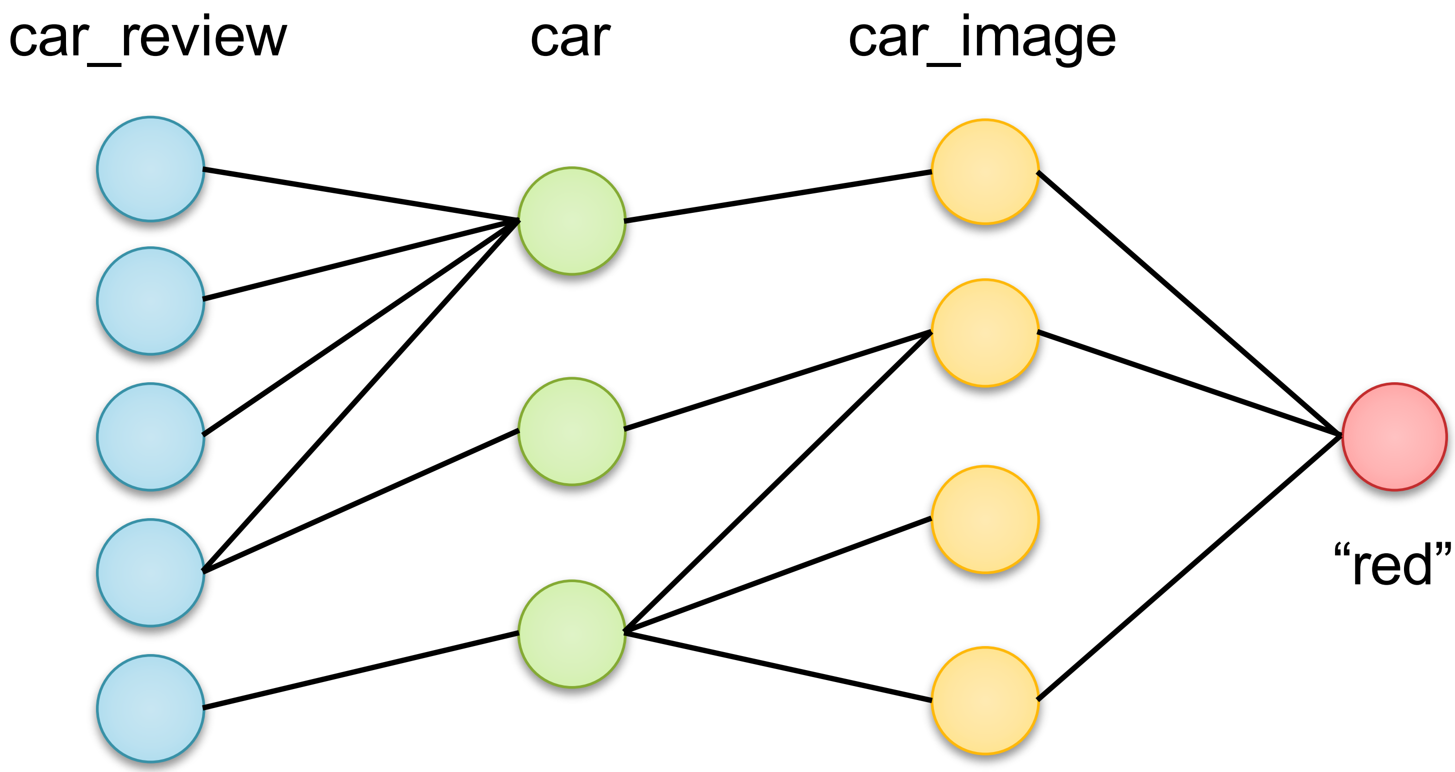
```
COLLECT car.make, car.model
WHERE car.style = "SUV";
```

- Query Semantics

```
SELECT *
FROM car_image M, car C, car_review R
WHERE M.(make,model) CROWDJOIN C.(make,model)
AND R.(make, model) CROWDJOIN C.(make,model)
AND M.color CROWDEQUAL "red"
```

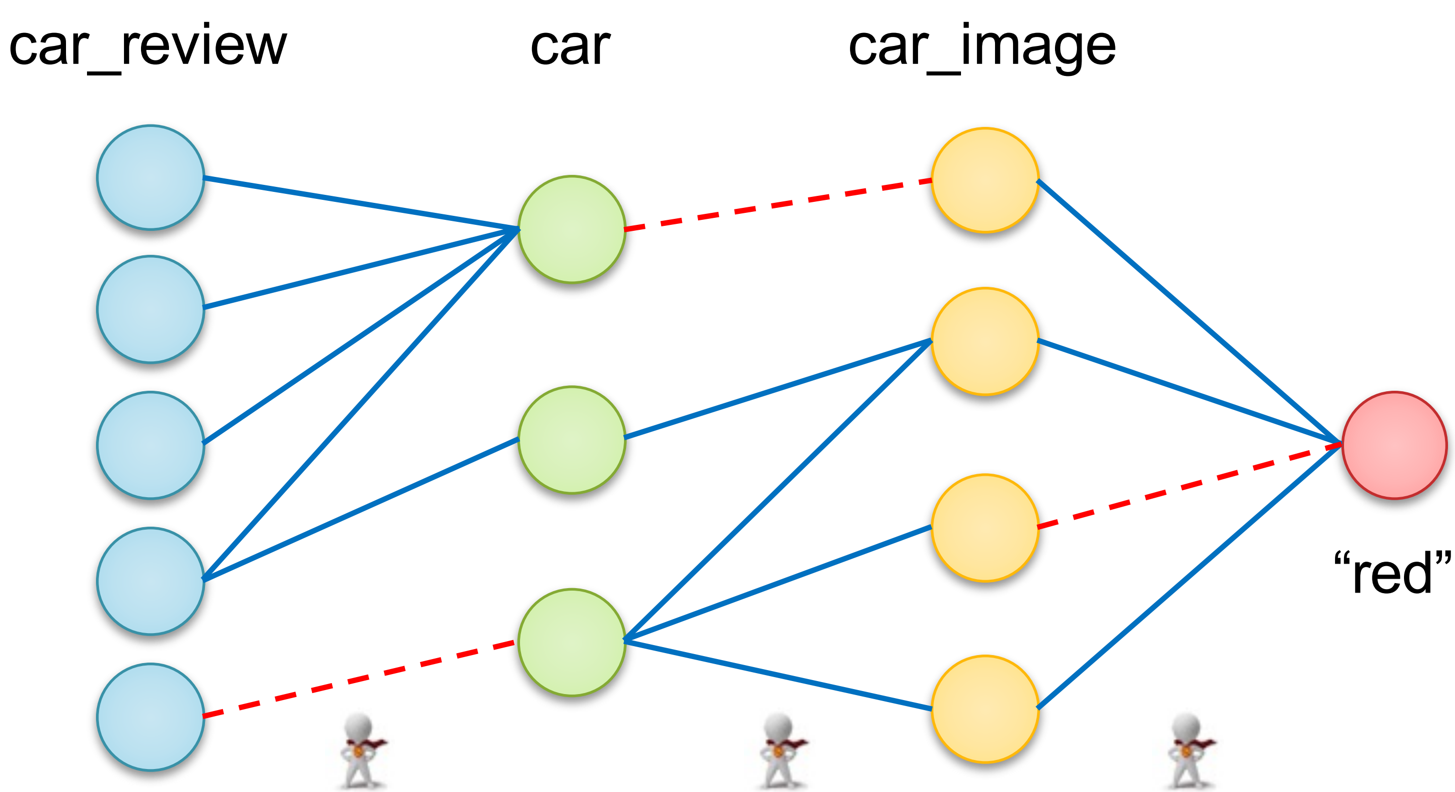
CDB Query Processing

- Graph-Based Query Model
  - Computing matching probabilities each CROWDJOIN
  - Building a query graph that connects tuple pairs with matching probabilities larger than a threshold



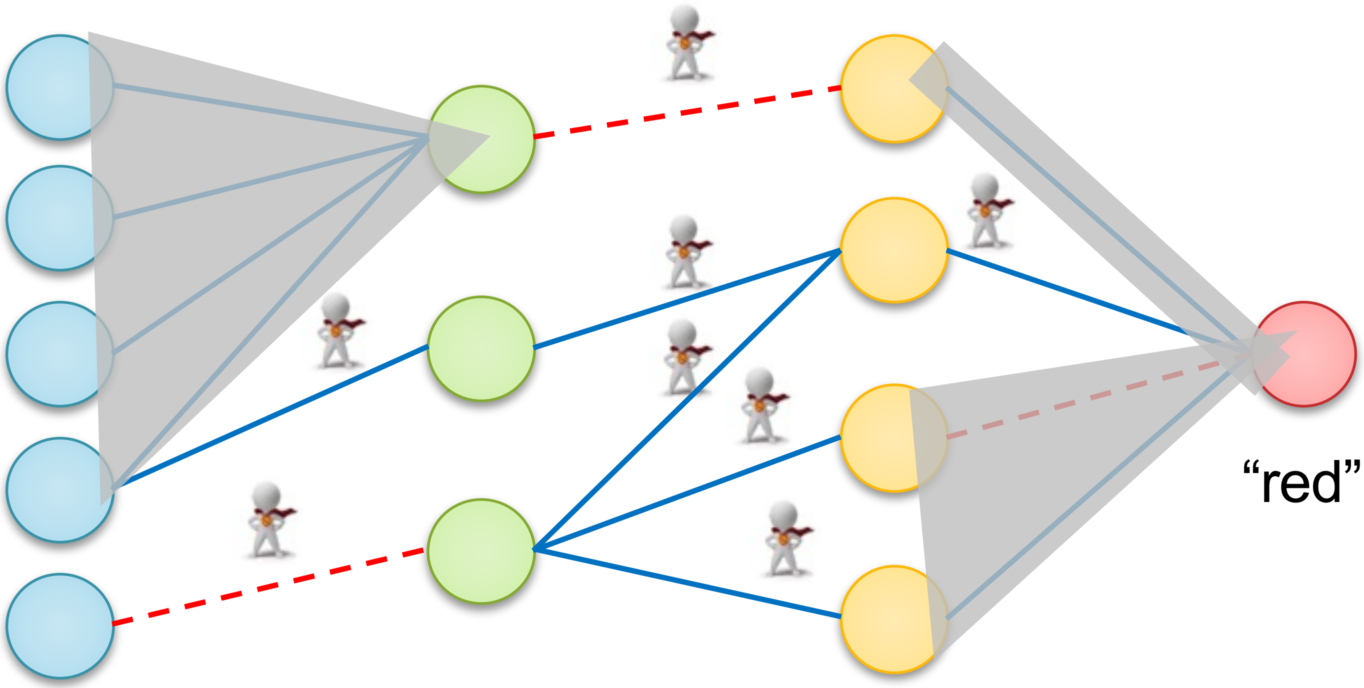
CDB Query Processing

- Graph-Based Query Model
  - Crowdsource all edges (Yes/No tasks)
  - Coloring edges by the crowd answers
  - Result tuple: a path containing all CROWDJOINs



CDB Query Optimization

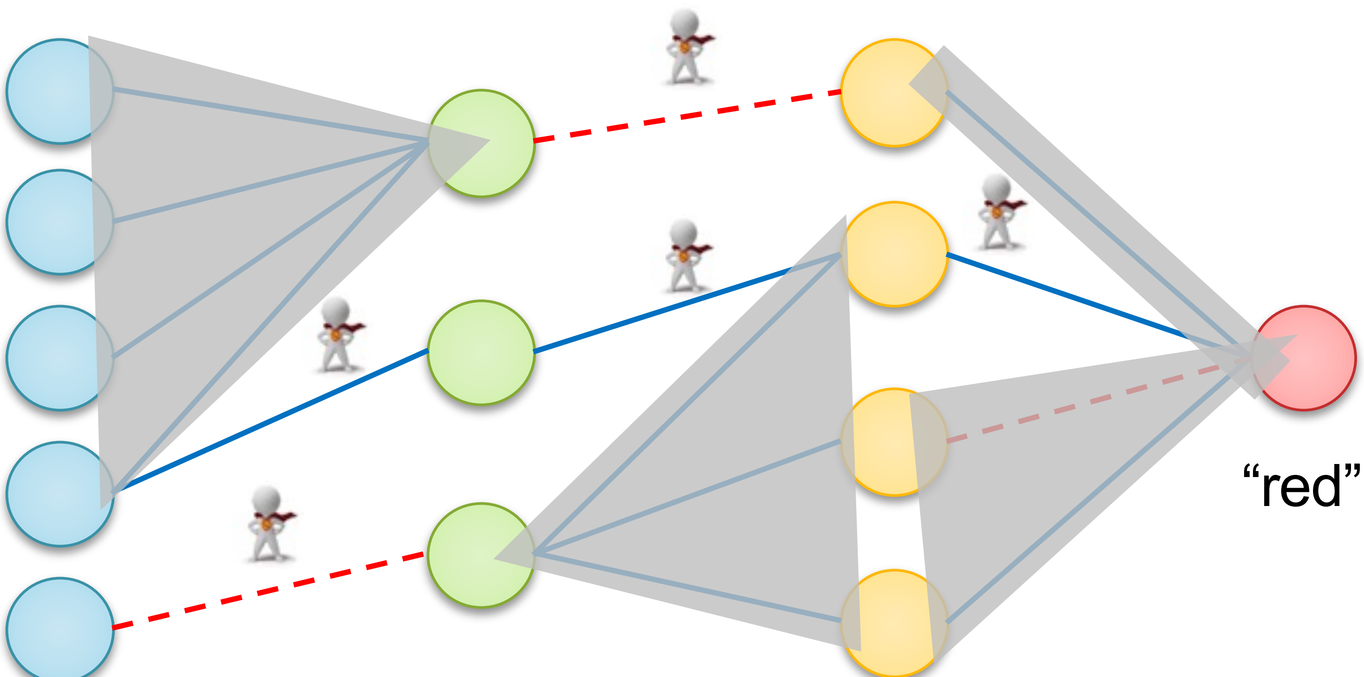
- Monetary cost control
  - Traditional goal: finding an optimal join order
  - CDB goal: selecting minimum number of edges



Traditional 2 tasks + 5 tasks + 1 task = 8 tasks

CDB Query Optimization

- Monetary cost control
  - Traditional goal: finding an optimal join order
  - CDB goal: selecting minimum number of edges

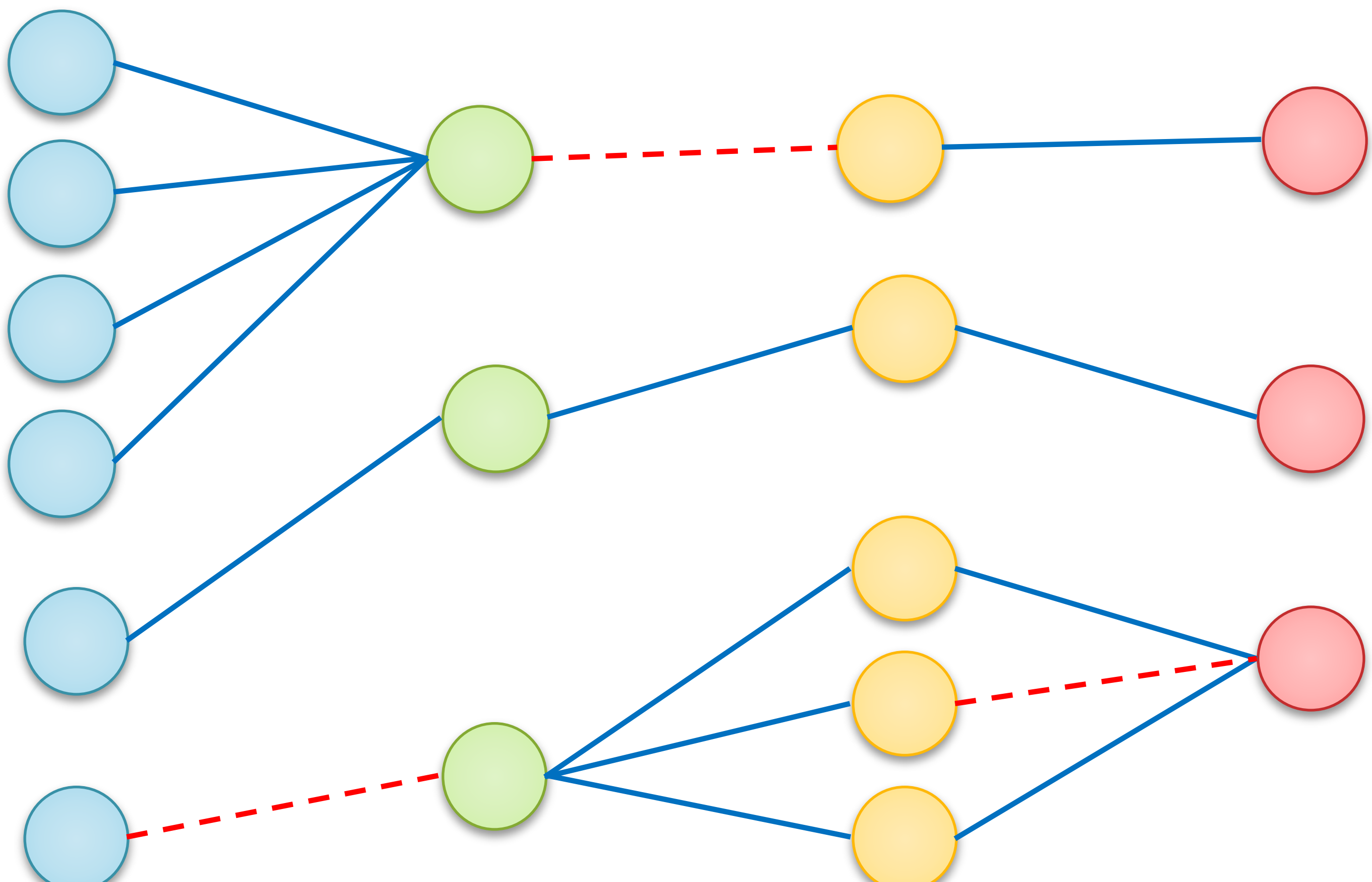


Traditional 2 tasks + 5 tasks + 1 task = 8 tasks

CDB 5 tasks **NP-HARD → Various Heuristics**

CDB Query Optimization

- Latency control
  - Partitioning the graph into connected components
  - Crowdsourcing each components in parallel



CDB Query Optimization

- Quality control
  - Probabilistic truth inference model

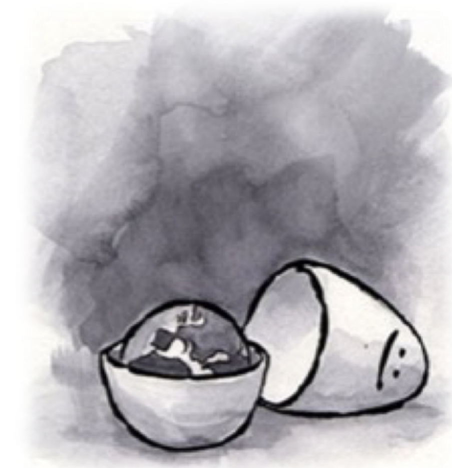
$$p_i = \frac{\prod_{(w,a)\in V_t} (q_w)^{\mathbb{1}\{i=a\}} \cdot (\frac{1-q_w}{\ell-1})^{\mathbb{1}\{i\neq a\}}}{\sum_{j=1}^{\ell} \prod_{(w,a)\in V_t} (q_w)^{\mathbb{1}\{j=a\}} \cdot (\frac{1-q_w}{\ell-1})^{\mathbb{1}\{j\neq a\}}}$$

  - Entropy-based task assignment model

$$\mathcal{I}(t) = \mathcal{H}(\vec{p}) - \sum_{i=1}^{\ell} [\, p_i \cdot q_w + (1-p_i) \cdot \frac{1-q_w}{\ell-1} \,] \cdot \mathcal{H}(\vec{p'}).$$

- Other Task Types
  - Single-choice & Multi-choice tasks
  - Fill-in-blank tasks
  - Collection tasks

Take-Away for System Design



- **Data Model**
  - Relational model
  - Open world assumption
- **Query Language**
  - Extending SQL
  - Supporting interactions with the crowd
- **Query Processing**
  - Tree-based vs. Graph-based
  - Crowd-powered operators
  - Optimization: Quality, Cost, and Latency

Crowdsourcing DB Systems

- **System Overview**
  - CrowdDB
  - Qurk
  - Deco
  - CDAS
  - CDB

Crowdsourcing Systems

- **Operator Design**
  - Design Principles

Crowdsourcing Operators

Design Principles

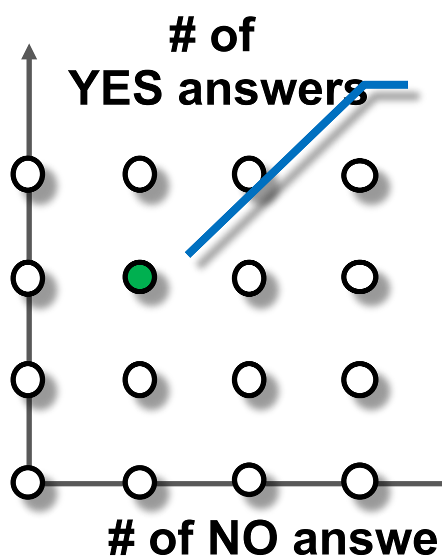
- **Leveraging crowdsourcing techniques**
  - **Quality Controlling**
    - **Truth Inference**: inferring correct answers
    - **Task Assignment**: assigning tasks judiciously
  - **Cost Controlling**
    - **Answer Deduction**: avoiding unnecessary costs
    - **Task Selection**: selecting most beneficial tasks
  - **Latency Controlling**
    - **Round Reduction:** reducing # of rounds
  - **Task Design**
    - **Interface Design:** interacting with crowd wisely

Crowdsourced Selection

- **Objective**
  - Identifying items satisfying some conditions
- **Key Idea**
  - Task Assignment: cost vs. quality

```
Find all images containing SUV cars from an image set
```

For each image



- ***(x,y)***: x YES, y No
- **Truth Inference**
  - Output PASS?
  - Output FAIL?
- **Task Assignment**
  - Ask one more?

A. G. Parameswaran et al.: CrowdScreen: algorithms for filtering data with humans. SIGMOD Conference 2012: 361-372

Crowdsourced Selection

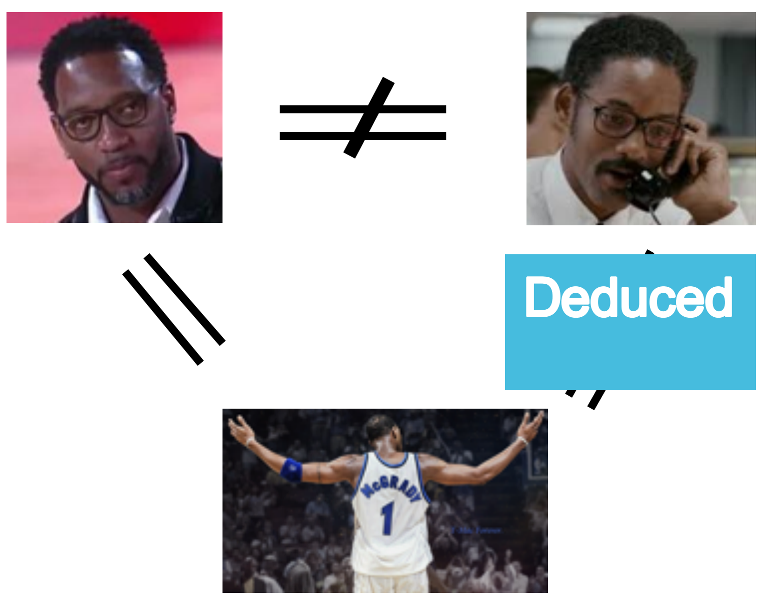
- **Key Idea**
  - Latency Controlling: cost vs. latency

```
Find 2 images with SUV cars from 100 images
```

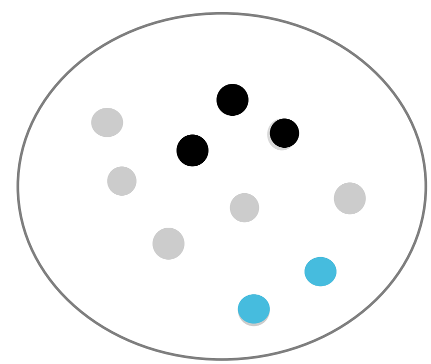
A. D. Sarma et al.: Crowd-powered find algorithms. ICDE 2014: 964-975

Crowdsourced Join

- **Objective**
  - Identifying record pairs referring to same entity
- **Key Idea**
  - Answer Deduction, e.g., using Transitivity



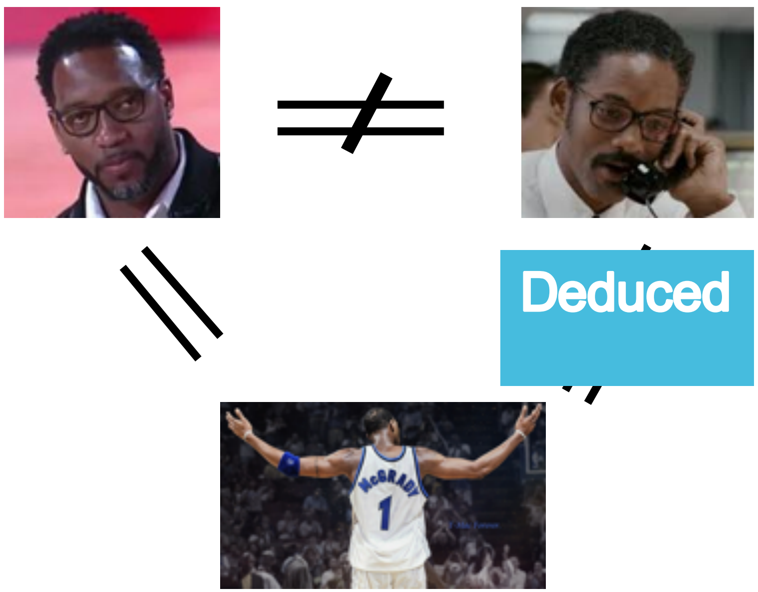
Task Pool



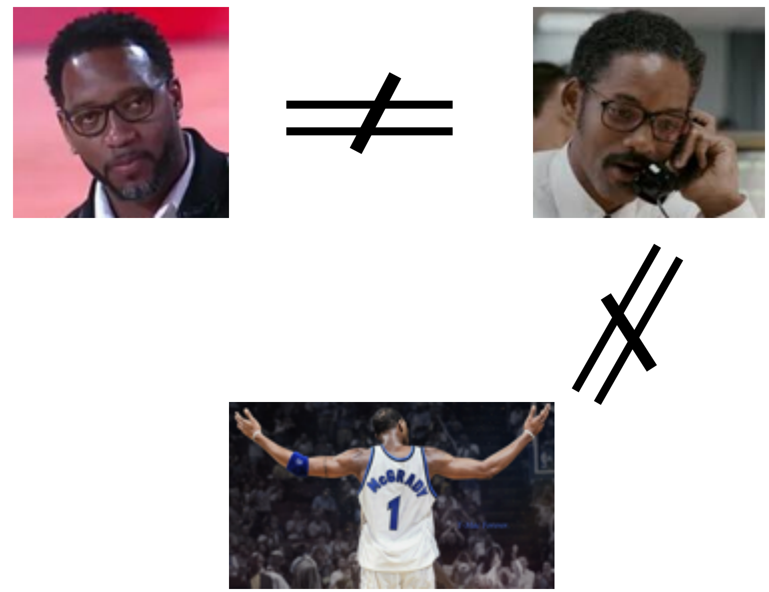
- Jiannan Wang, Guoliang Li, Tim Kraska, Michael J. Franklin, Jianhua Feng: Leveraging transitive relations for crowdsourced joins. SIGMOD 2013
- Donatella Firmani, Barna Saha, Divesh Srivastava: Online Entity Resolution Using an Oracle. PVLDB 2016

Crowdsourced Join

- **Key Idea**
  - Task Selection, e.g., selecting beneficial tasks



One task deduced



No task deduced

- Jiannan Wang, Guoliang Li, Tim Kraska, Michael J. Franklin, Jianhua Feng: Leveraging transitive relations for crowdsourced joins. SIGMOD 2013
- S. E. Whang, P. Lofgren, H. Garcia-Molina: Question Selection for Crowd Entity Resolution. PVLDB 6(6): 349-360 (2013)

Crowdsourced TopK/Sort

- **Objective**
  - Finding top-k items (or a ranked list) wrt. Criterion
- **Key Idea**
  - Truth Inference: Resolve conflicts among crowd

Which picture visualizes the best SFU Campus?

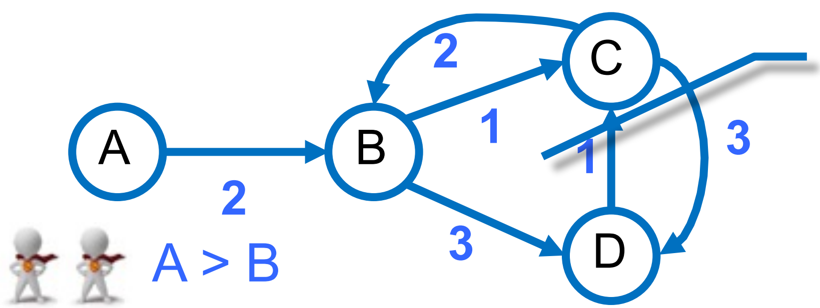
A

B

C

D

Pair-wise Voting



- **Ranking Inference over conflicts among crowd**
  - Max Likelihood Inference
  - NP-hard

S. Guo, et al. : So who won?: dynamic max discovery with the crowd. SIGMOD Conference 2012: 385-396

Crowdsourced TopK/Sort

- **Key Idea**
  - Task Selection: Most beneficial for getting the top-k results

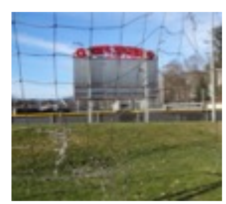
What are the top-2 picture that visualizes the best SFU Campus?

Rank by computers



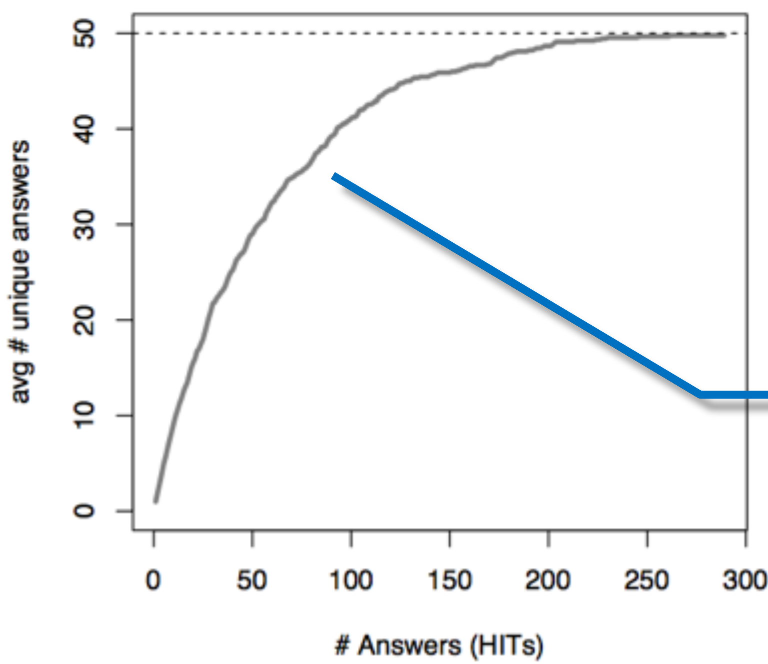


The most beneficial task:
Difficult to computers


vs.


Crowdsourced Collection

- **Objective**
  - **Collecting a set of new items**
- **Key Idea**
  - **Truth Inference: Inferring item coverage**



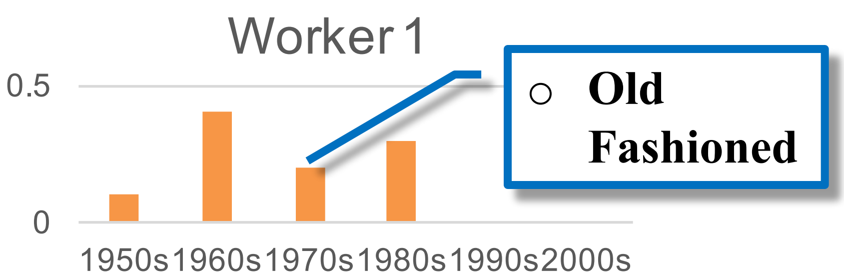
- **Species Estimation Algo.**
  - Observing the rate at which new species are identified over time
  - inferring how close to the true number of species you are

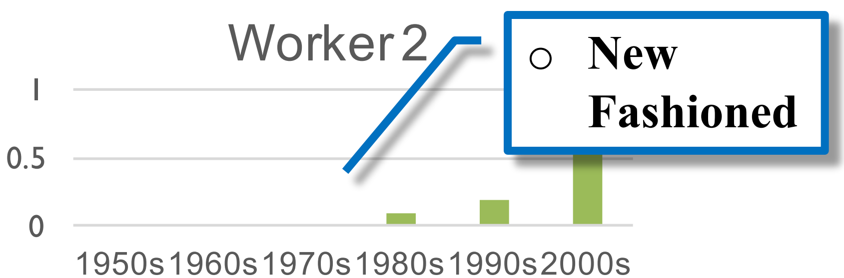
B. Trushkowsky et al.: Crowdsourced enumeration queries. ICDE 2013: 673-684

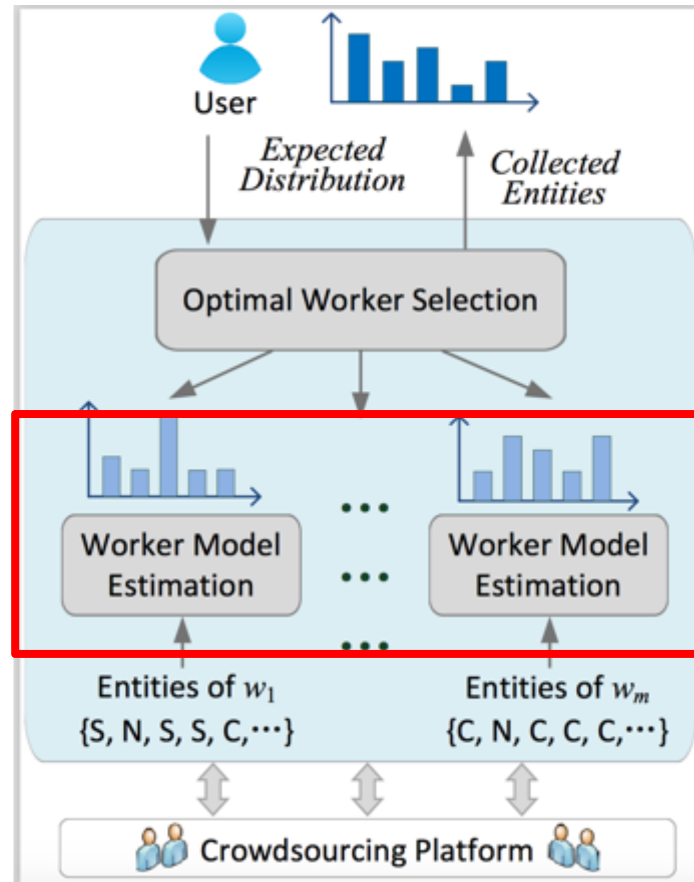
Crowdsourced Collection

- **Key Idea**
  - Task Assignment: satisfying result distribution

- **Diverse distributions among workers**
  - E.g., collecting movies with publishing decades







J. Fan et al.: Distribution-Aware Crowdsourced Entity Collection. TKDE 2017

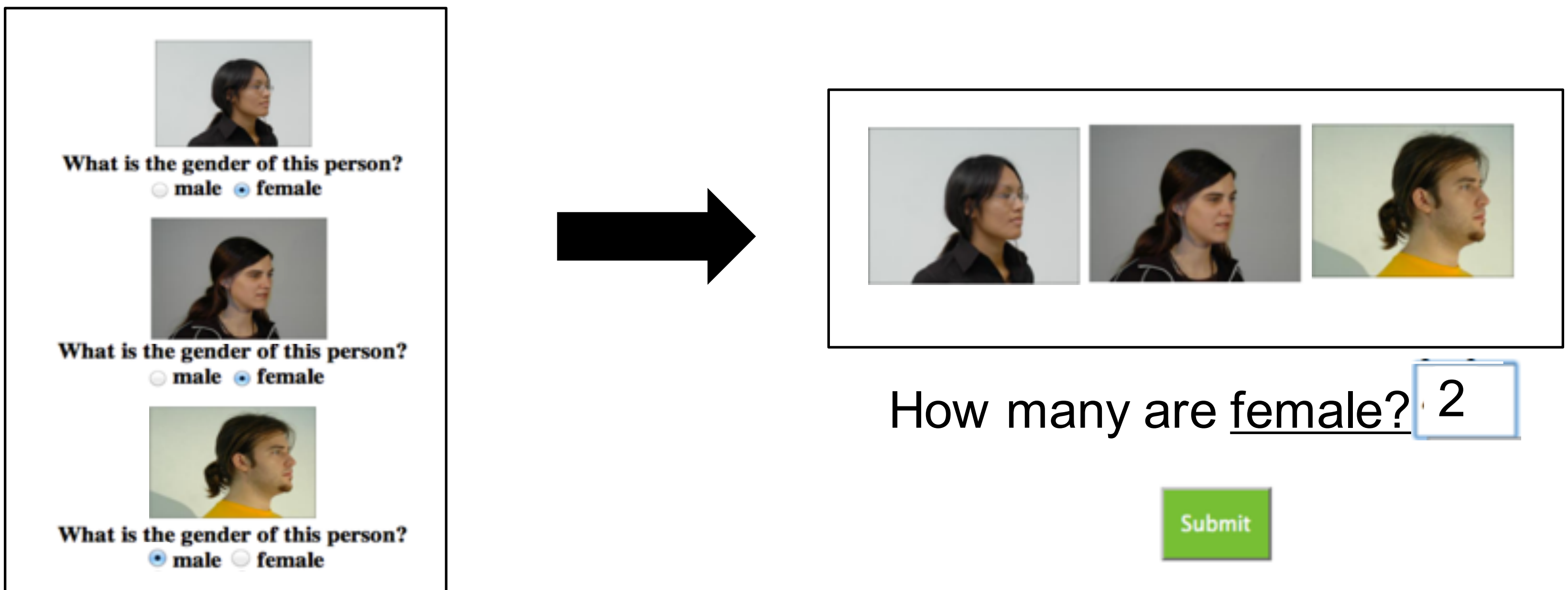
Crowdsourced Fill

- **Objective**
  - Filling missing cells in a table
- **Key Idea: Task Design**
  - Microtask vs. partially-filled table with voting
  - Real-Time collaboration for concurrent workers
  - Compensation scheme with budget

name $0.03	*nationality* $0.01	*position* $0.01	*caps* $0.05	*goals* $0.01	$0.02
Lionel Messi	**Argentina**	**FW**	**83**		
Ronaldinho	Brazil	MF	*Empty*	*Empty*	
Neymar	Brazil	FW	*Empty*	*Empty*	
Iker Casillas	Spain	FW	**150**	**0**	
Ronaldinho	Brazil	FW	*Empty*	33	

Crowdsourced Count

- **Objective**
  - **Estimating number of certain items**
- **Key Idea**
  - **Task Design: Leveraging crowd to estimate**



Adam Marcus, David R. Karger, Samuel Madden, Rob Miller, Sewoong Oh: Counting with the Crowd. PVLDB 2012

Take-Away for Crowd Operators

	CrowdSelect	CrowdJoin	CrowdSort	CrowdCollect	CrowdFill	CrowdCount
Truth Inference	√	√	√	√	×	×
Task Assignment	√	×	√	√	×	×
Answer Deduction	×	√	×	×	×	×
Task Selection	×	√	√	×	×	×
Round Reduction	√	√	×	×	×	×
Interface Design	×	√	√	×	√	√

System Comparison

		CrowdDB	Qurk	Deco	CDAS	CDB
Crowd Powered Operators	CrowdSelect	√	√	√	√	√
	CrowdJoin	√	√	√	√	√
	CrowdSort	√	√	×	×	√
	CrowdTopK	√	√	×	×	√
	CrowdMax	√	√	×	×	√
	CrowdMin	√	√	×	×	√
	CrowdCount	×	×	×	×	√
	CrowdCollect	√	×	√	×	√
	CrowdFill	√	×	√	√	√

System Comparison

		CrowdDB	Qurk	Deco	CDAS	CDB
Optimization Objectives	Cost	√	√	√	√	√
	Latency	×	×	×	√	√
	Quality	√	√	√	√	√
Design Techniques	Truth Inference	√	√	√	√	√
	Task Assignment	×	×	×	×	√
	Answer Reasoning	×	×	×	×	√
	Task Design	√	√	√	√	√
	Latency Reduction	×	×	×	√	√

Reference

1. M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. Crowddb: answering queries with crowdsourcing. In SIGMOD, pages 61–72, 2011.
2. A. Marcus, E. Wu, S. Madden, and R. C. Miller. Crowdsourced databases: Query processing with people. In CIDR, pages 211–214, 2011.
3. H. Park, R. Pang, A. G. Parameswaran, H. Garcia-Molina, N. Polyzotis, and J. Widom. Deco: A system for declarative crowdsourcing. PVLDB, 2012.
4. J. Fan, M. Zhang, S. Kok , M. Lu, and B. C. Ooi. Crowdop: Query optimization for declarative crowdsourcing systems. IEEE Trans. Knowl. Data Eng., 27(8):2078–2092, 2015.
5. G. Li, C. Chai, J. Fan, X. Weng, J. Li, Y. Zheng, Y. Li, X. Yu, X. Zhang, H. Yuan. CDB: Optimizing Queries with Crowd-Based Selections and Joins. in SIGMOD, 2017.
6. A. G. Parameswaran et al.: CrowdScreen: algorithms for filtering data with humans. SIGMOD Conference 2012: 361-372.
7. A. D. Sarma et al.: Crowd-powered find algorithms. ICDE 2014: 964-975.
8. Jiannan Wang, Guoliang Li, Tim Kraska, Michael J. Franklin, Jianhua Feng: Leveraging transitive relations for crowdsourced joins. SIGMOD 2013.
9. Donatella Firmani, Barna Saha, Divesh Srivastava: Online Entity Resolution Using an Oracle. PVLDB 2016.
10. S. E. Whang, P. Lofgren, H. Garcia-Molina: Question Selection for Crowd Entity Resolution. PVLDB 6(6): 349-360 (2013).
11. S. Guo, et al. : So who won?: dynamic max discovery with the crowd. SIGMOD Conference 2012: 385-396.
12. Xiaohang Zhang, Guoliang Li, Jianhua Feng: Crowdsourced Top-k Algorithms: An Experimental Evaluation. PVLDB 2016.
13. B. Trushkowsky et al.: Crowdsourced enumeration queries. ICDE 2013: 673-684.
14. J. Fan et al.: Distribution-Aware Crowdsourced Entity Collection. TKDE 2017.
15. H. Park, J. Widom: CrowdFill: collecting structured data from the crowd. SIGMOD Conference 2014: 577-588.
16. Adam Marcus, David R. Karger, Samuel Madden, Rob Miller, Sewoong Oh: Counting with the Crowd. PVLDB 2012.

Outline

- Crowdsourcing Overview (30min)
  - Motivation (5min)
  - Workflow (15min)
  - Platforms (5min)
  - Difference from Other Tutorials (5min)
- Fundamental Techniques (100min)
  - Quality Control (60min)
  - Cost Control (20min)
  - Latency Control (20min)
- Crowdsourced Database Management (40min)
  - Crowdsourced Databases (20min)
  - Crowdsourced Optimizations (10min)
  - Crowdsourced Operators (10min)
- Challenges (10min)

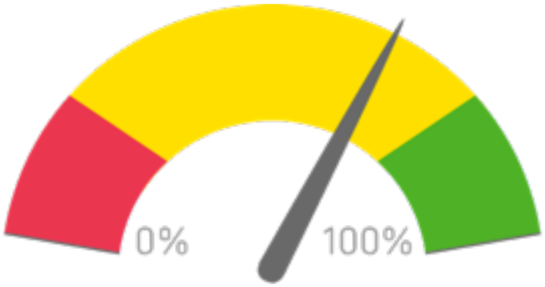
Part 1

Part 2

The 6 Crowdsourcing Challenges

- Benchmarking
- Scalability
- Truth Inference
- Privacy
- Macro-Tasks
- Mobile Crowdsourcing

1. Benchmarking



- Database Benchmarks

  TPC-C, TPC-H, TPC-DI,…

- Crowdsourcing
  No standard benchmarks

- Existing public datasets (link) are inadequate

1. Benchmarking

- Existing public datasets are inadequate, because:

- Each task often receives 5 or less answers
- Most tasks are single-label tasks
- Very few numeric tasks
- Lack ground truth
  - Expensive to get ground truth for 10K tasks

2. Scalability

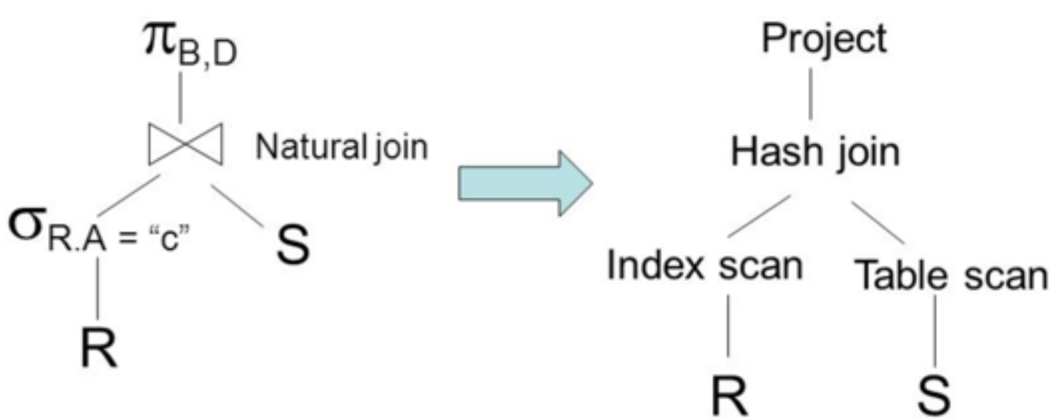
- Hard to Scale in Crowdsourcing to tackle the 3Vs of Big Data?

- (1) workers are **expensive**;
  (2) answers can be **erroneous**;
  (3) existing works focus on **specific problems**, e.g., active learning (Mozafari et al. VLDB14), entity matching (Gokhale et al. SIGMOD14).

2. Scalability: Query Optimization

- Query Processing in Traditional RDBMS

Parser → Logical Query Plan → Query Rewriter → Physical Query Plan → Query Optimization

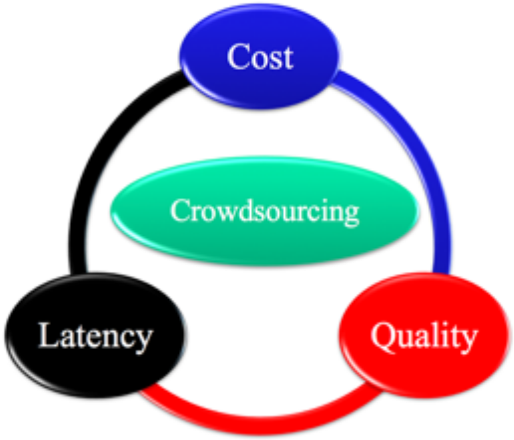


IBM DB2
PostgreSQL
MySQL
SYBASE
Microsoft SQL Server

2. Scalability: Query Optimization

- Query optimization in crowdsourcing is challenging:

(1) handle 3 optimization objectives



(2) humans are more unpredictable than machines

3. Truth Inference

- **Not fully solved**
  (Zheng et al. VLDB17)

- **We have surveyed 20+ methods:**

  (1) No best method;

  (2) The oldest method (David & Skene JRSS 1979) is the most robust;

  (3) No robust method for numeric tasks (the baseline "Mean" performs the best !)

4. Privacy

- (1) **Requester**

  Wants to protect the privacy of their tasks from workers

  e.g., tasks may contain sensitive attributes, e.g., medical data.

4. Privacy

- (2) **Workers**

  Want to have privacy-preserving requirement & worker profile

  e.g., personal info of workers can be inferred from the worker's answers, e.g., location, gender, etc.

5. Macro-Tasks

- Existing works focus on simple **micro-tasks**

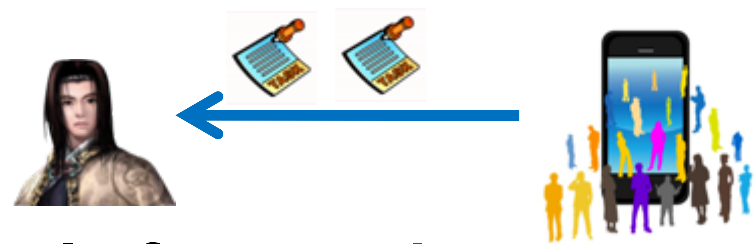
Is Bill Gates currently the CEO of Microsoft ?	Identify the sentiment of the tweet: ……
○ Yes ○ No	○ Pos ○ Neu ○ Neg

- Hard to perform big and complex tasks, e.g., writing an essay

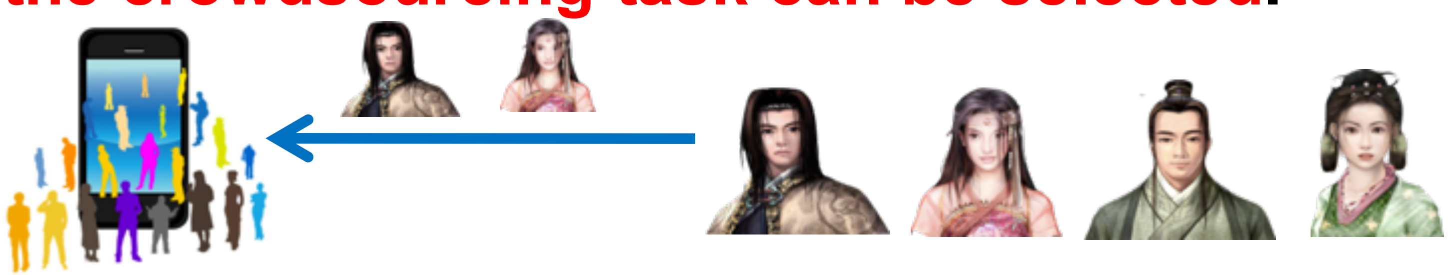
  (1) macro-tasks are **hard to be split** and accomplished by multiple workers;
  (2) workers may not be interested to perform a **time-consuming** macro-task.

6. Mobile Crowdsourcing

- **Emerging mobile crowdsourcing platforms**
  e.g., gMission (HKUST), ChinaCrowd (Tsinghua)
- **Challenges**
  (1) Other factors (e.g., spatial distance, mobile user interface) affect workers' latency and quality;
- **(2) Different mechanisms**
  traditional crowdsourcing platforms: workers request tasks from the platform;



for mobile crowdsourcing platform: only workers close to the crowdsourcing task can be selected.



Thanks !
Q & A

Guoliang Li
Tsinghua University

Yudian Zheng
Hong Kong University

Ju Fan
Renmin University

Jiannan Wang
SFU

Reynold Cheng
Hong Kong University