Data-Driven Crowdsourcing: Management, Mining, and Applications

Lei Chen @ HKUST

Dongwon Lee @ Penn State

Tova Milo @ Tel Aviv

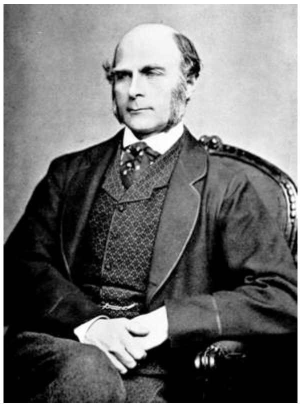
TOC

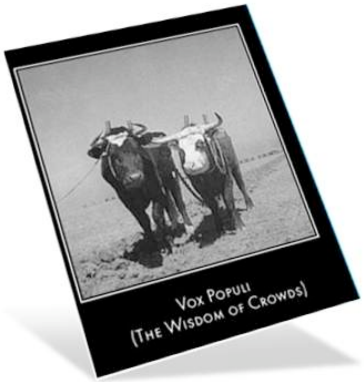
- Part I
  - Crowdsourced Data Management
- Part II
  - Crowdsourced Data Mining
- Part III
  - Crowdsourced Social Applications

PART I: CROWDSOURCED DATA MANAGEMENT

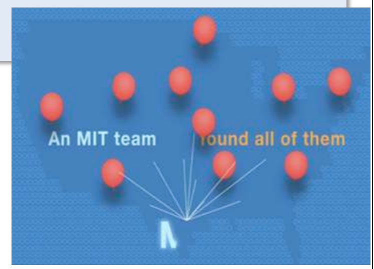
Eg, Francis Galton, 1906

Weight-judging competition:
1,197 (mean of 787 crowds) vs. 1,198 pounds (actual measurement)



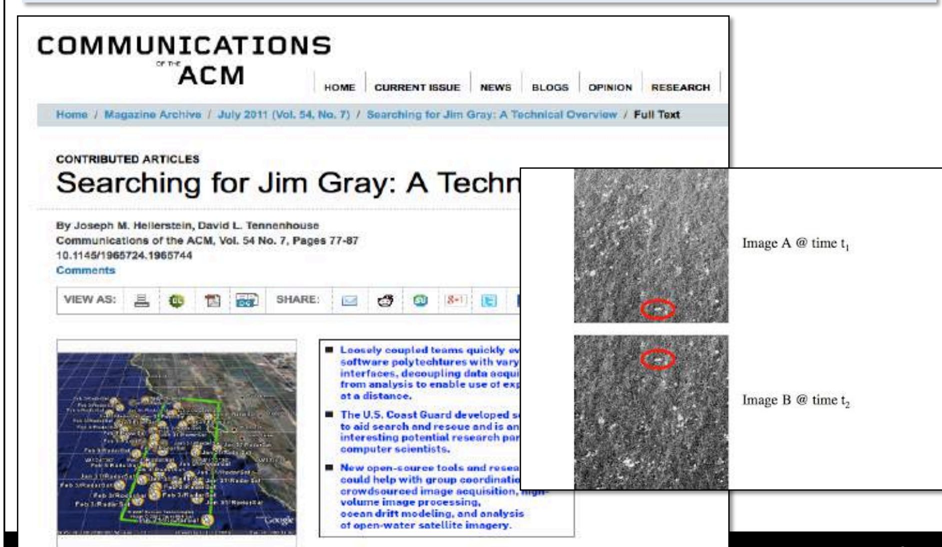


Eg, DARPA Challenge, 2009



- To locate 10 red balloons in arbitrary locations of US
- Winner gets $40K
- MIT team won the race with the strategy:
  - 2K per balloon to the first person, A, to send the correct coordinates
  - 1K to the person, B, who invited A
  - 0.5K to the person, C, who invited B, …

Eg, Finding Jim Gray, 2007

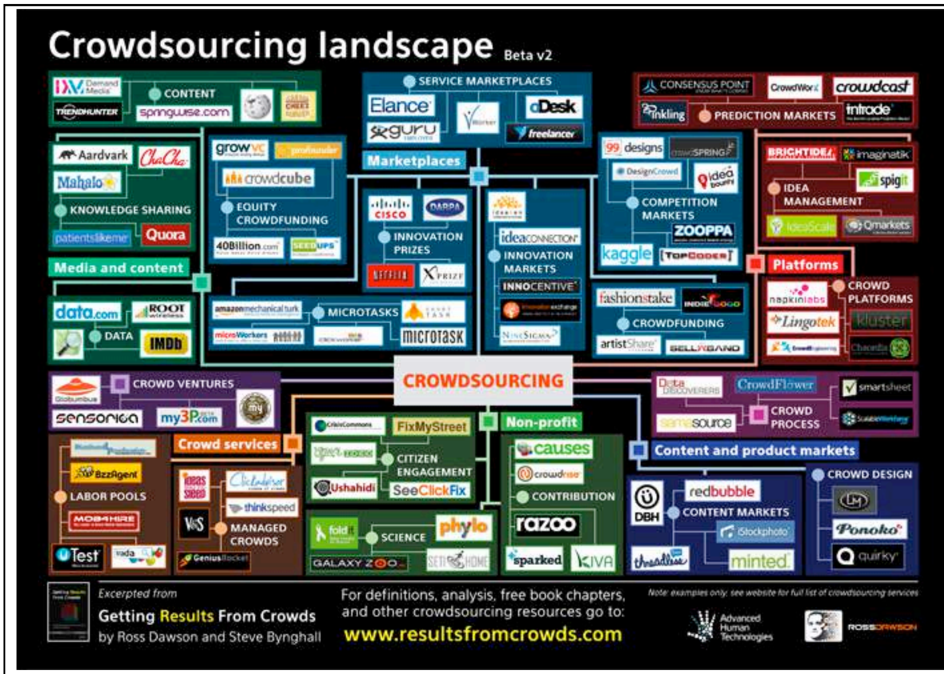


Eg, Jeff Howe, WIRED, 2006

"**Crowdsourcing** represents the act of a company or institution taking a function once performed by employees and *outsourcing* it to an undefined (and generally large) network of people in the form of an open call. … The crucial prerequisite is the use of the *open call* format and the *large* network of potential laborers…"

What is Crowdsourcing?

- Many definitions
- A few characteristics
  - **Outsourced** to **human** workers
  - **Online** and **distributed**
  - Open call & right **incentive**
  - **Diversity** and **independence**
- When to use?
  1. Machine cannot do the task well
  2. Large crowds can probably do it well
  3. Task can be split to many micro-tasks



http://www.resultsfromcrowds.com/features/crowdsourcing-landscape/

9

Marketplaces

- Requesters
  - People submit some tasks
  - Pay rewards to workers

Submit tasks

- Marketplaces
  - Provide crowds with tasks

amazon mechanical turk
Artificial Artificial Intelligence
CrowdFlower
CloudCrowd

Find tasks

- Crowds
  - Workers perform tasks

ICDE 2015 Tutorial

10

Notable Marketplaces

- Mechanical Turk
- CrowdFlower
- CloudCrowd
- Clickworker
- SamaSource

amazonmechanical turk
Artificial Artificial Intelligence

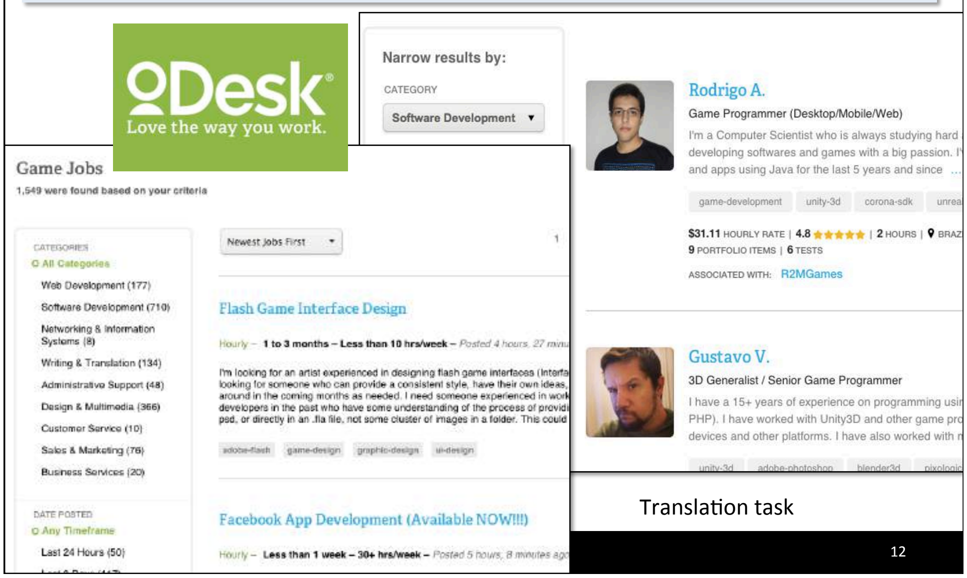
CrowdFlower

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

clickworker

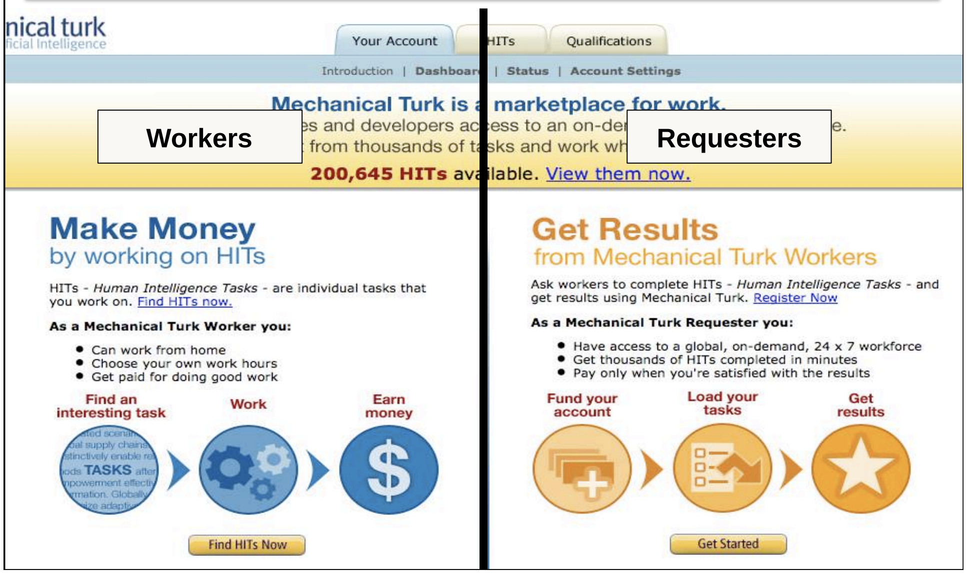
samasource

Eg, oDesk: Micro- vs. Macro-task



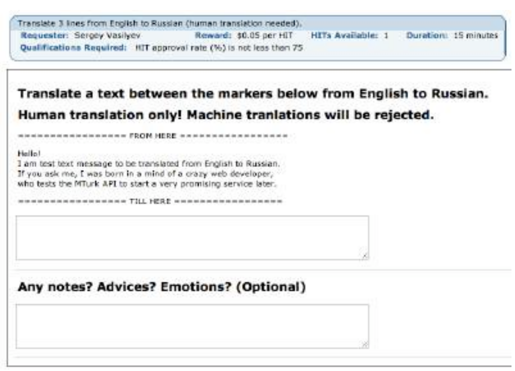
Translation task

Eg, Amazon Mechanical Turk (AMT)



AMT HIT

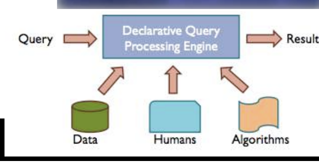
- Tasks
  - Called **HIT** (Human Intelligence Task)
  - Micro-task
- Eg
  - Data cleaning
  - Tagging / labeling
  - Sentiment analysis
  - Categorization
  - Surveying
  - Photo moderation
  - Transcription



Translation task

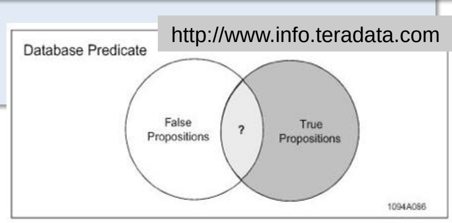
Crowdsourced DB Projects

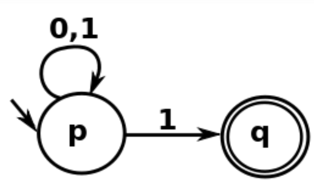
- CDAS @ NUS
- CrowdDB @ UC Berkeley & ETH Zurich
- MoDaS @ Tel Aviv U.
- Qurk @ MIT
- sCOOP @ Stanford & UCSC

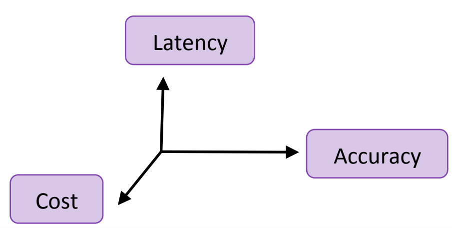


New DB Challenges

- 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





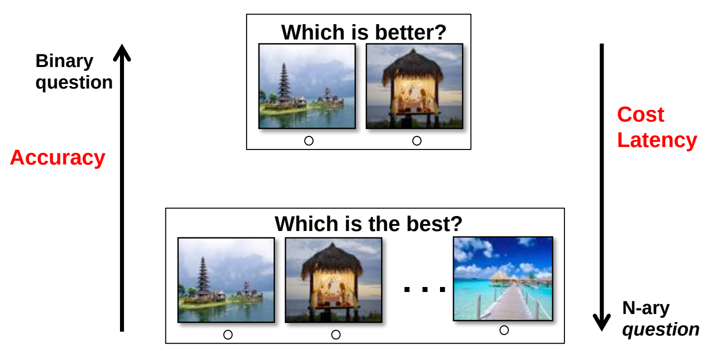


Crowdsourced DB Research Questions

- New Data Model
- New Query Language
- **New Operator Algorithm** ⇐ FOCUS
- New Query Optimization
- …

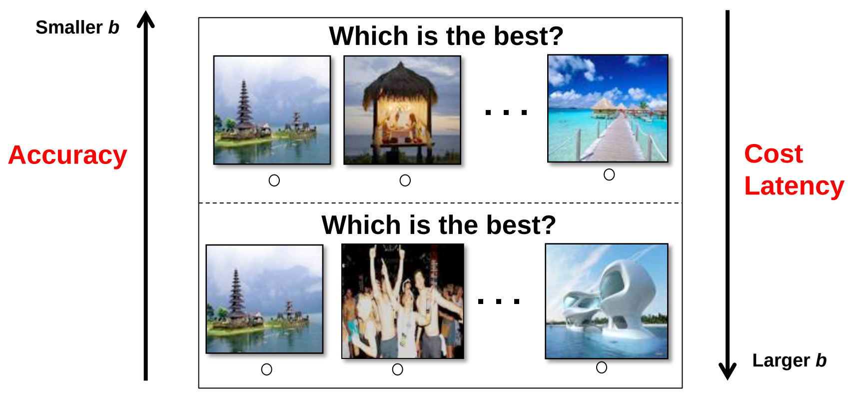
Size of Comparison

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



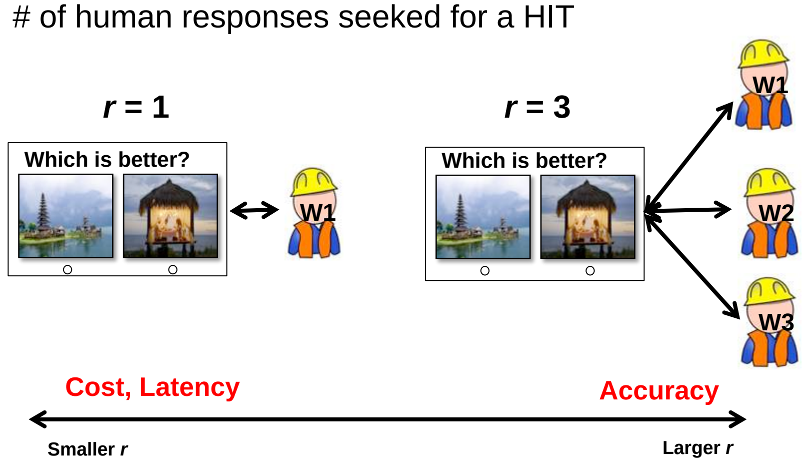
Size of Batch

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



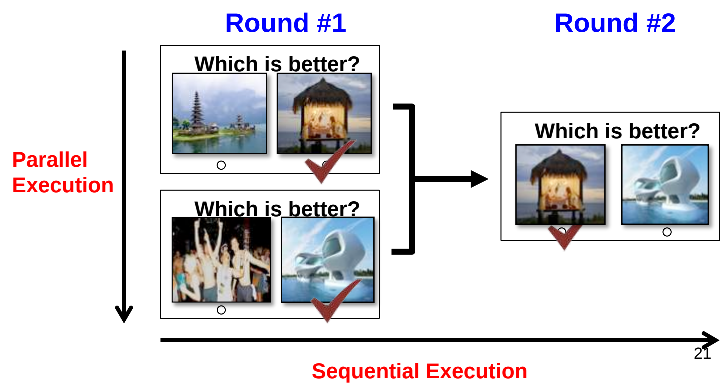
Response (r)

- # of human responses seeked for a HIT



Round (= Step)

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



DB Operations

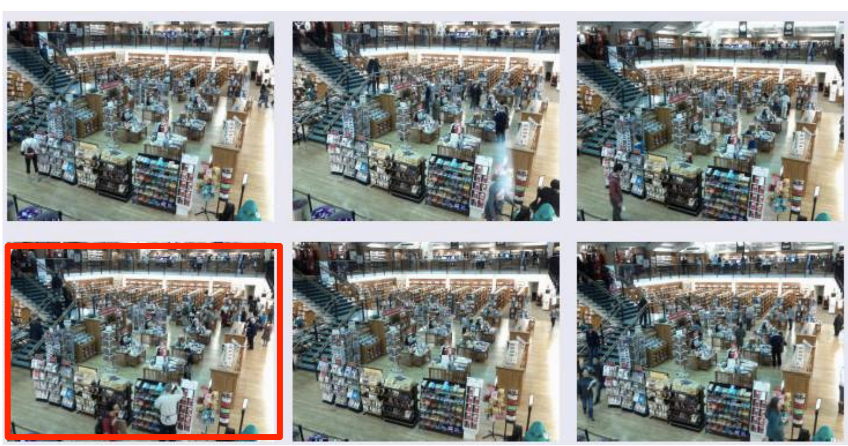
- Top-1 (= Max)
- Top-k
- Sort
- *Demo*
- Select
- Count
- Join

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

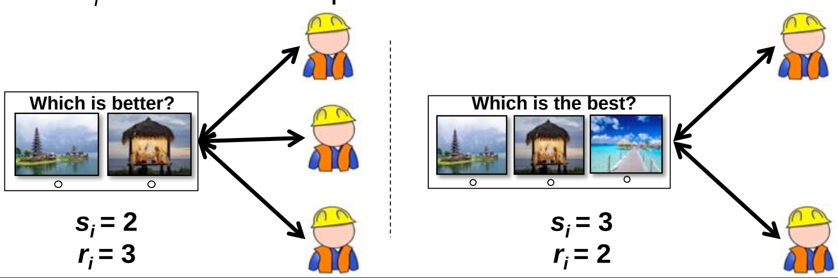
Max [Venetis-WWW12]

- Finding a peak hour

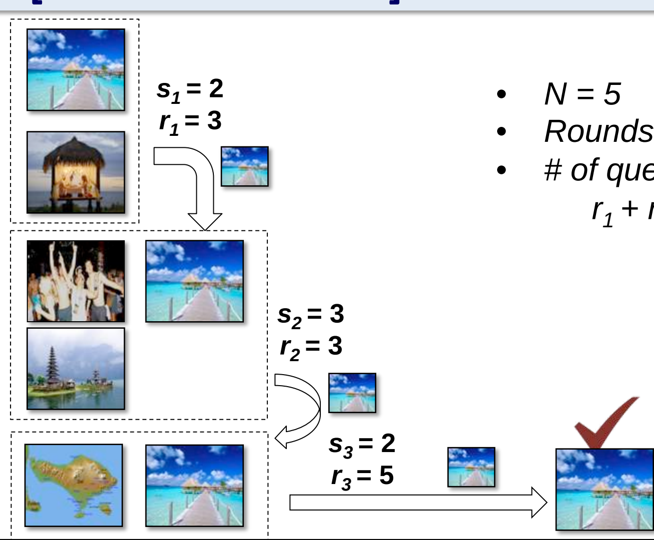


Max [Venetis-WWW12]

- 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

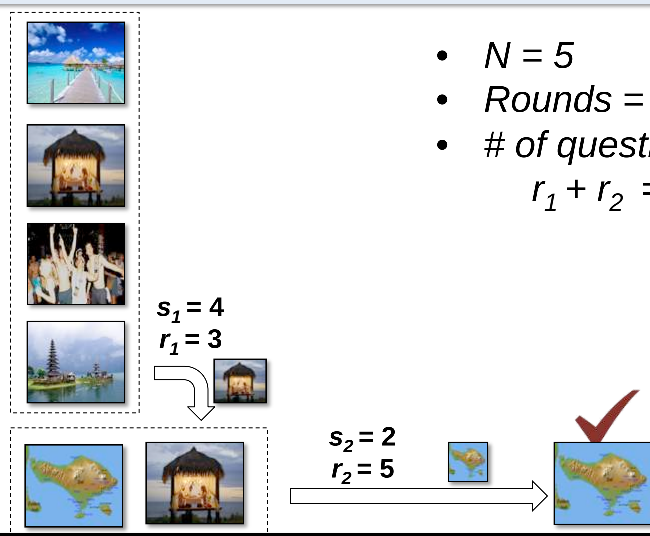


Max [Venetis-WWW12]: bubble max #1



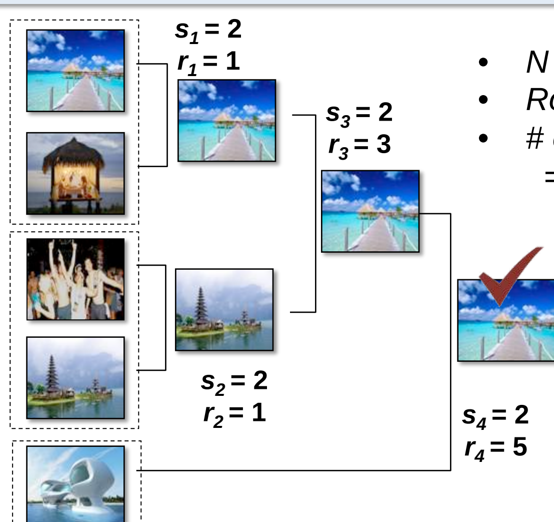
- *N = 5*
- *Rounds* = 3
- *# of questions* = $r_1 + r_2 + r_3 = 11$

Max [Venetis-WWW12]: bubble max #2



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

Max [Venetis-WWW12]: tournament max

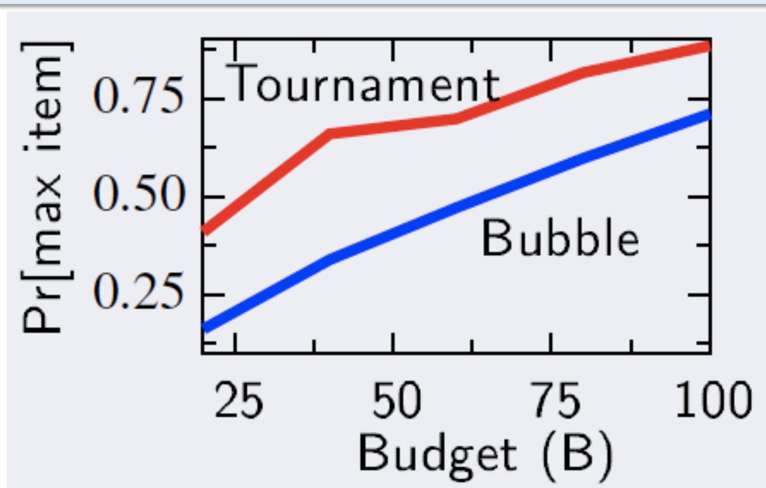


- $N = 5$
- $Rounds = 3$
- # of questions $= r_1 + r_2 + r_3 + r_4 = 10$

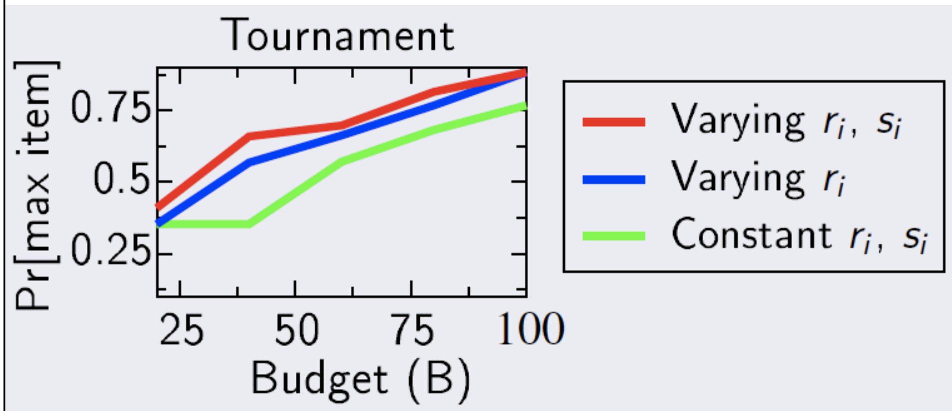
Max [Venetis-WWW12]

- 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]

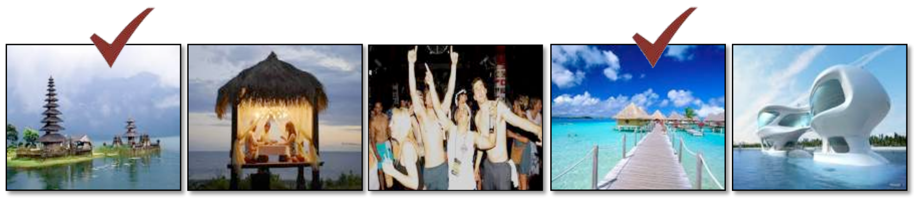


Top-K Operation

- **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*

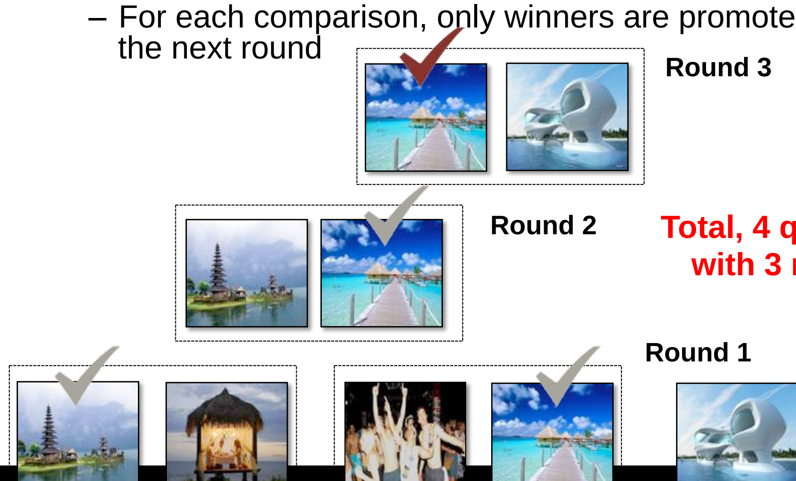
Top-K Operation

- 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 Operation: tournament (k=2)

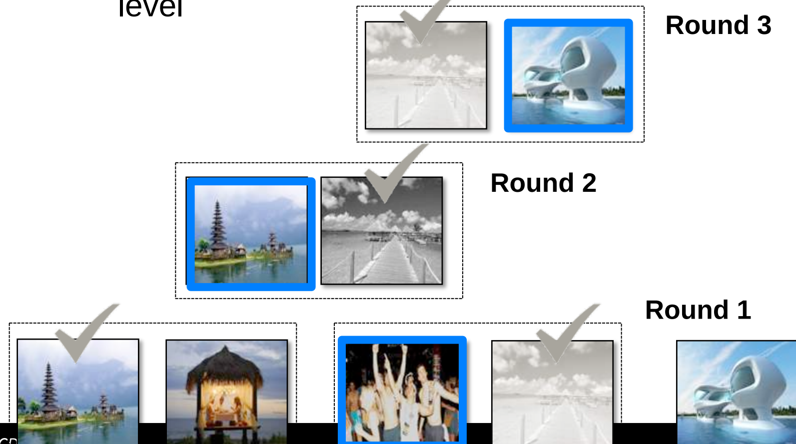
- Phase 1: **Building a tournament tree**
  - For each comparison, only winners are promoted to the next round



Total, 4 questions with 3 rounds

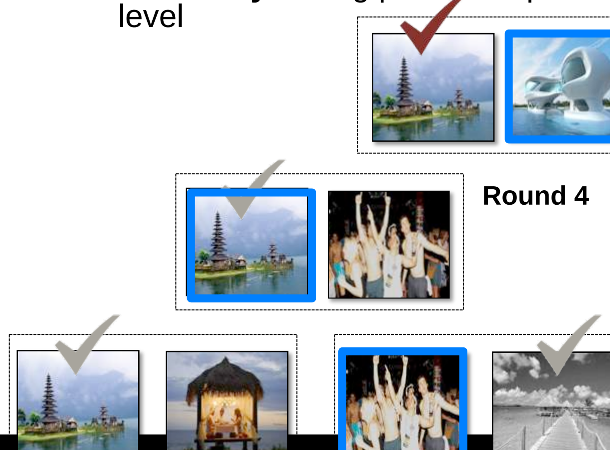
Top-K Operation: tournament (k=2)

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



Top-K Operation: tournament (k=2)

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



Total, 6 questions
With 5 rounds

Top-K Operation

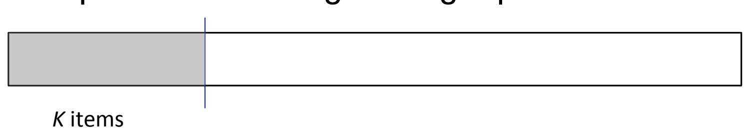
- This is a top-*k* **list** algorithm
- Analysis

	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]

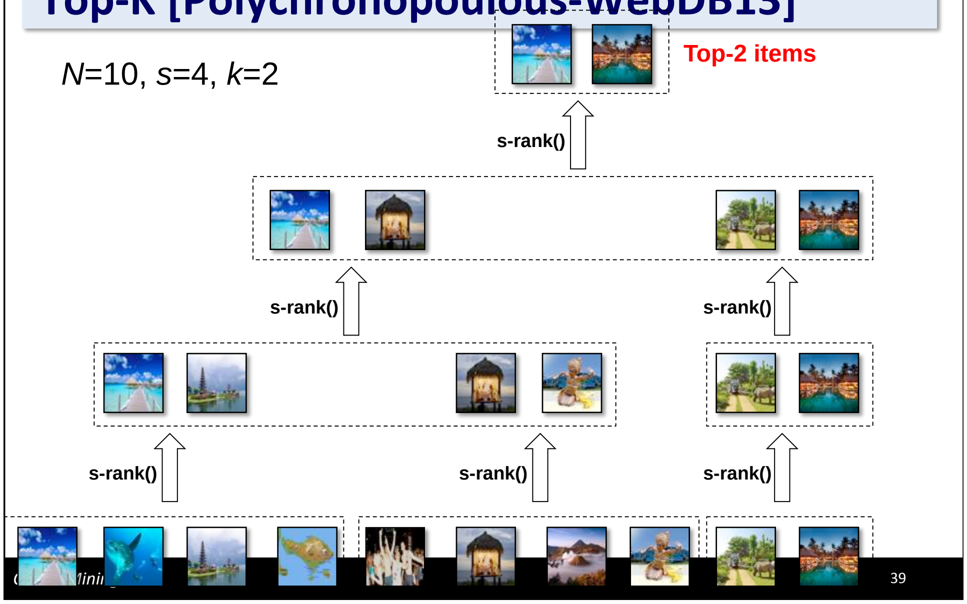
- Top-*k* **set** algorithm
  - Top-k items are "better" than remaining items
  - Capture NO ranking among top-*k* items



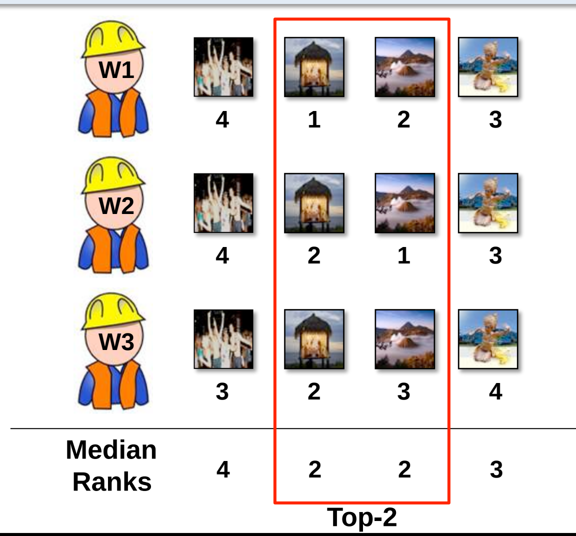
  - Effective when k is small
- Can become a Top-*k* **list** algorithm
  - Eg, Top-*k* **set** algorithm, followed by [Marcus-VLDB11] to sort *k* items

Top-K [Polychronopoulous-WebDB13]

N=10, s=4, k=2



Top-K [Polychronopoulous-WebDB13]



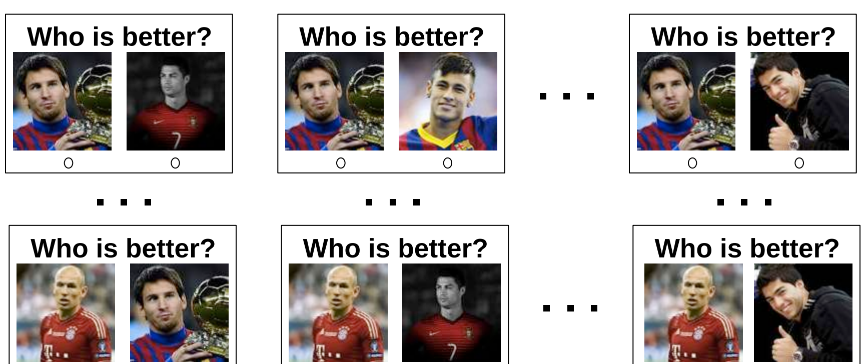
s-rank():
s=4, k=2,
w=3

Sort Operation

- Rank N items using crowdsourcing w.r.t. constraint C
  - C is subjective, fuzzy, ambiguous, and/or difficult-for-machines-to-compute

Naïve Sort

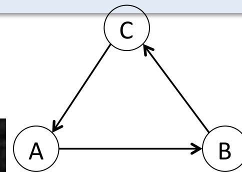
- Eg, "Which of two players is better?"
- Naïve all pair-wise comparisons takes $\binom{N}{2}$ comparisons
  - Optimal # of comparison is $O(N \log N)$

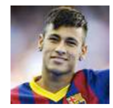


Naïve Sort

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

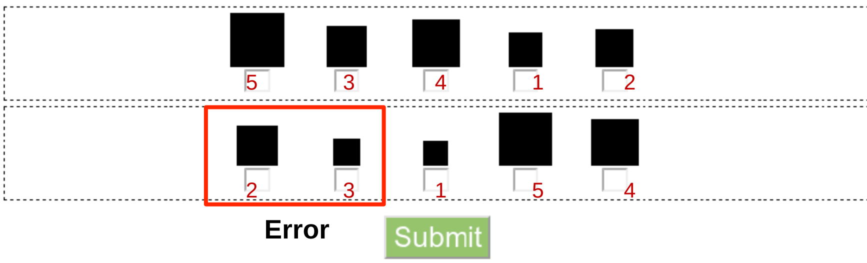
Sort [Marcus-VLDB11]

- 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]

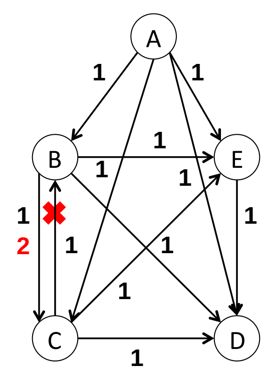
There are 2 groups of squares. We want to order the squares in each group from smallest to largest.

- 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]

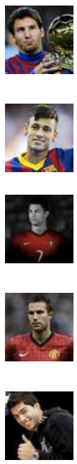
N=5, S=3

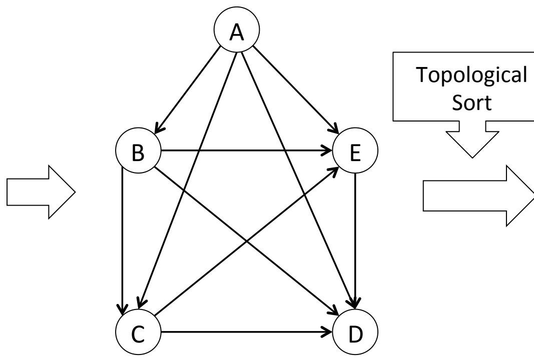


Sort [Marcus-VLDB11]

N=5, S=3

A B C D E





Sorted Result

A > B > C > E > D

ICDE 2015 Tutorial 47

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)*

Worker	Rating
W1	4
W2	3
W3	4

. . .

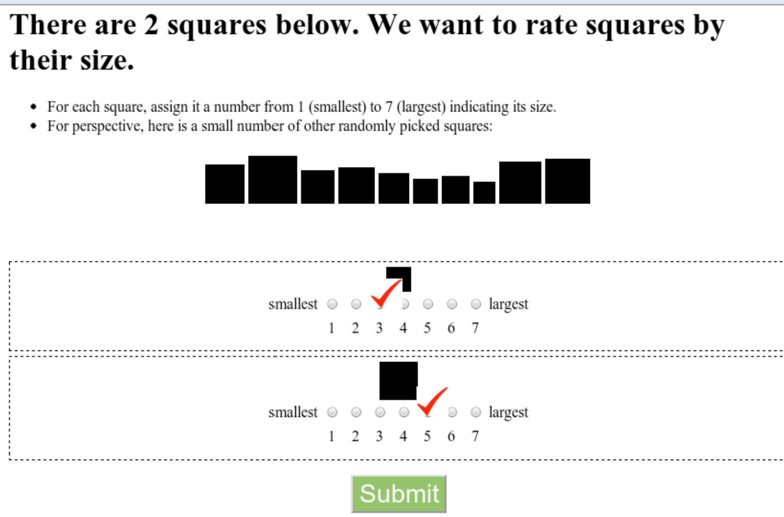
Worker	Rating
W1	1
W2	2
W3	1

Mean rating

	1.3
	3.6
. . .	
	8.2

ICDE 2015 Tutorial 48

Sort [Marcus-VLDB11]

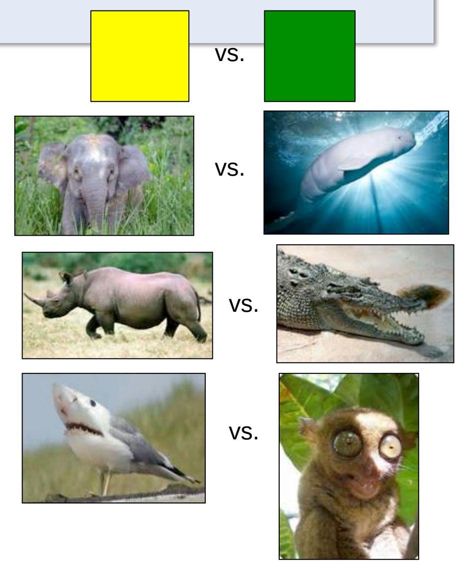


Sort [Marcus-VLDB11]

- #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

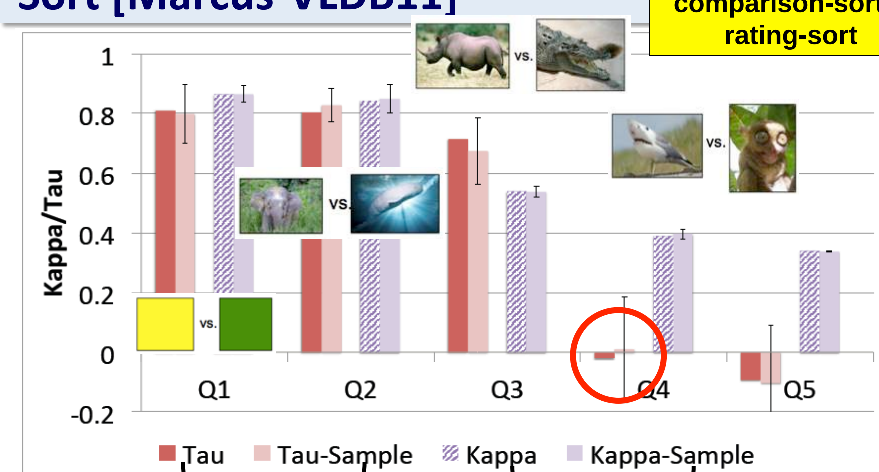
Sort [Marcus-VLDB11]

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



Sort [Marcus-VLDB11]

Finds that in general comparison-sort > rating-sort



Rank correlation btw. Comparison vs. rating

Worker agreement

Sort Demo

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

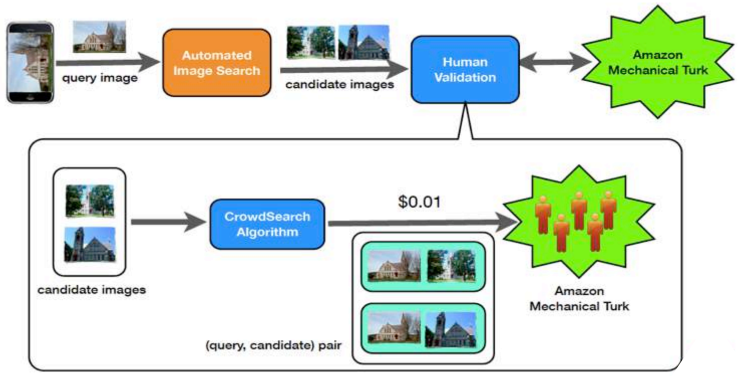
`http://goo.gl/3tw7b5`

Select Operation

- **Given *N* items, select *m* items that satisfy a predicate *P***
- ≈ Filter, Find, Screen, Search

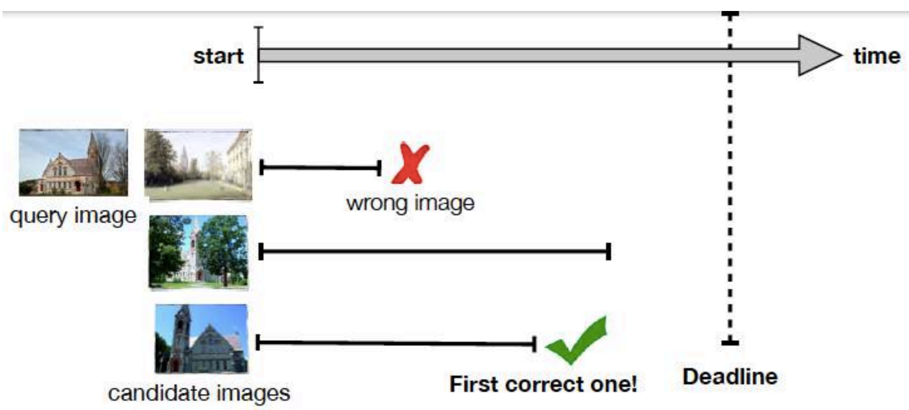
Select [Yan-MobiSys10]

- Improving mobile image search using crowdsourcing



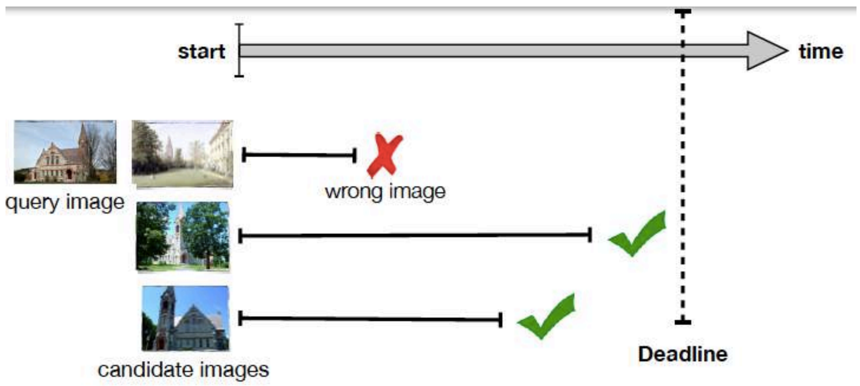
Select [Yan-MobiSys10]

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



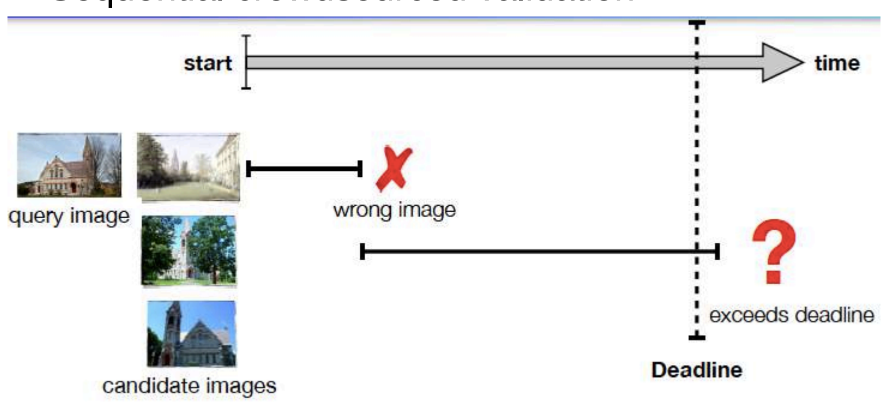
Select [Yan-MobiSys10]

- Parallel crowdsourced validation



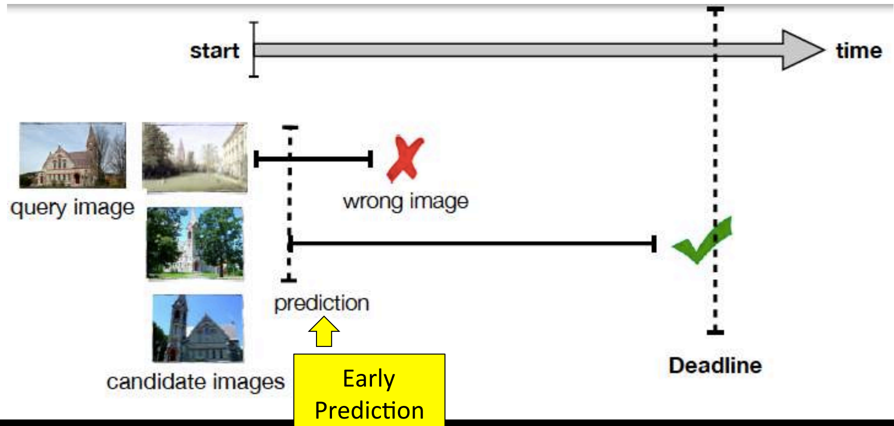
Select [Yan-MobiSys10]

- Sequential crowdsourced validation



Select [Yan-MobiSys10]

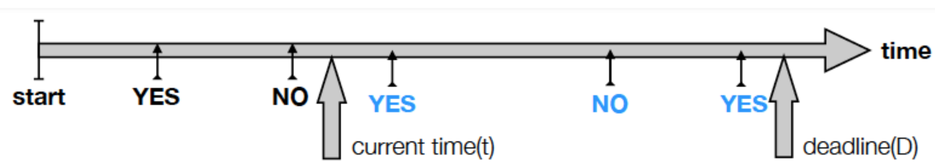
- CrowdSearch: using early prediction on the delay and outcome to start the validation of next candidate early



ICDE 2015 Tutorial 59

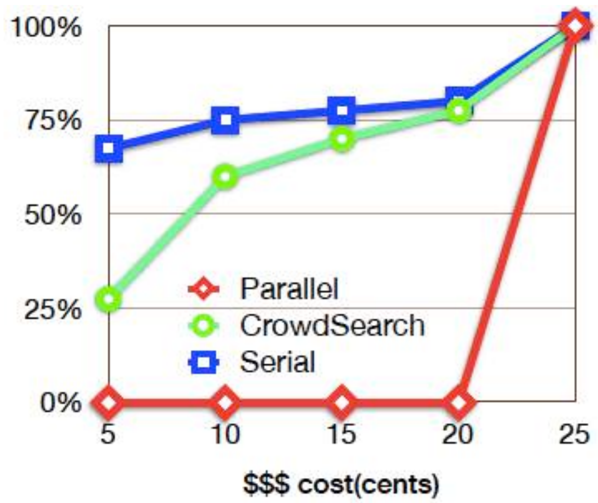
Select [Yan-MobiSys10]

- 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



ICDE 2015 Tutorial 60

Select [Yan-MobiSys10]



ICDE 2015 Tutorial 61

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

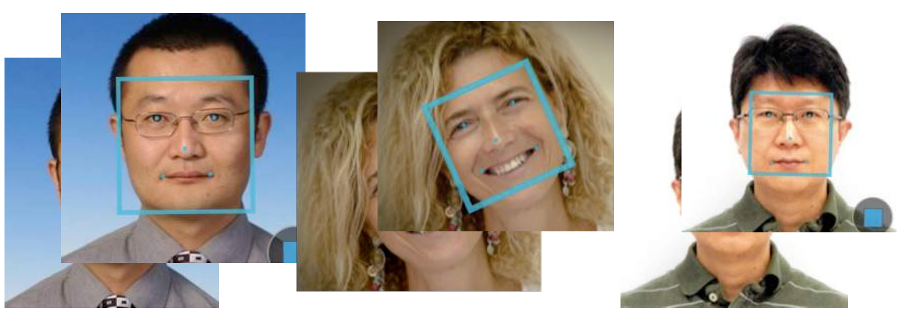
ICDE 2015 Tutorial 62

Count Operation

Q: “How many teens are participating in the Hong Kong demonstration in 2014?”

Count Operation

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



17 - 42 **7 - 34** ~~**10 - 28**~~

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

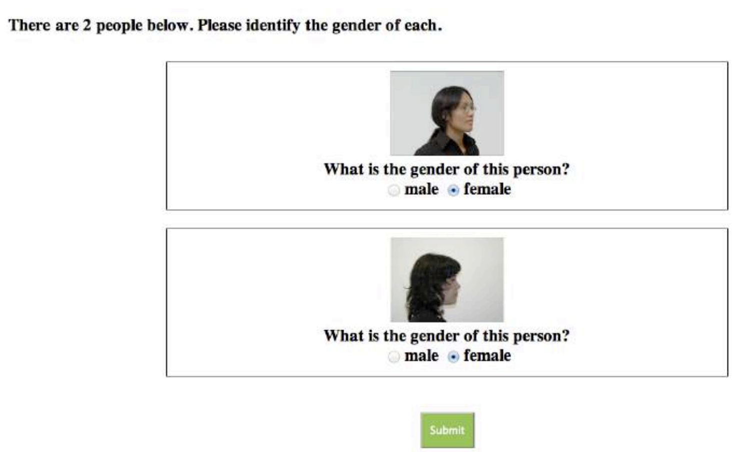
Count [Marcus-VLDB13]

- 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

ICDE 2015 Tutorial 65

Count [Marcus-VLDB13]

- Label Count (via sampling)



ICDE 2015 Tutorial 66

Count [Marcus-VLDB13]

- Batch Count

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

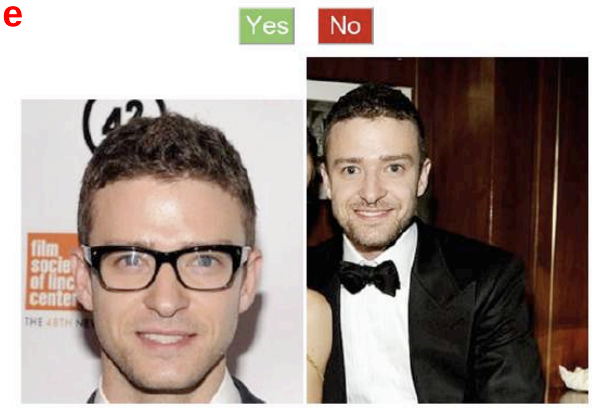
Join [Marcus-VLDB11]

- 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?

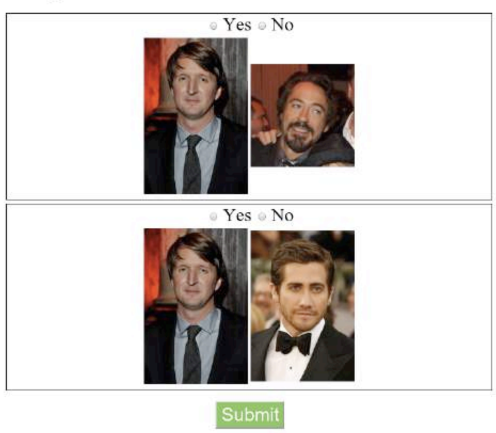
#1 Simple Join



Join [Marcus-VLDB11]

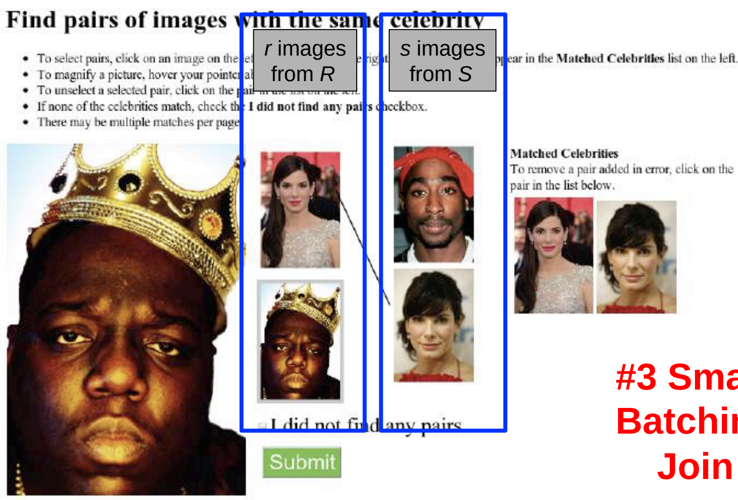
Is the same celebrity in the image on the left and the image on the right?

#2 Naïve Batching Join



Batch factor *b* = 2

Join [Marcus-VLDB11]



#3 Smart Batching Join

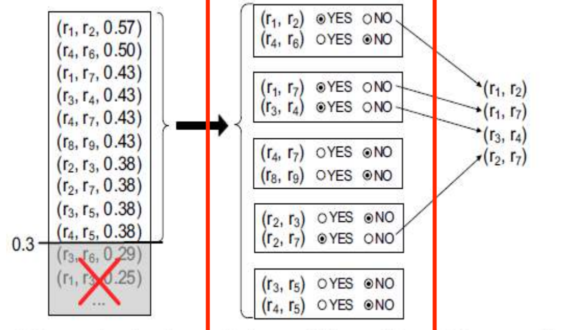
Join [Wang-VLDB12]

- [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:
  - 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
  - Algorithm to generate min # of HITs for step #2

Join [Wang-VLDB12]

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

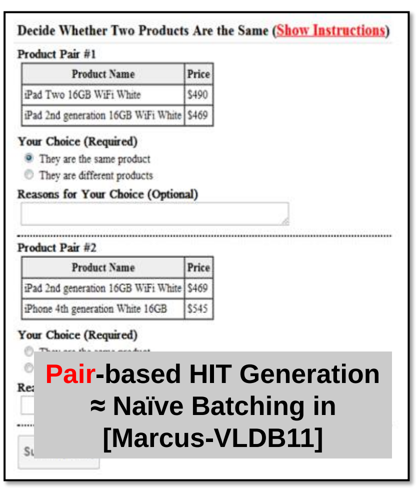
ID	Product Name	Price
r_1	iPad Two 16GB WiFi White	$490
r_2	iPad 2nd generation 16GB WiFi White	$469
r_3	iPhone 4th generation White 16GB	$545
r_4	Apple iPhone 4 16GB White	$520
r_5	Apple iPhone 3rd generation Black 16GB	$375
r_6	iPhone 4 32GB White	$599
r_7	Apple iPad2 16GB WiFi White	$499
r_8	Apple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



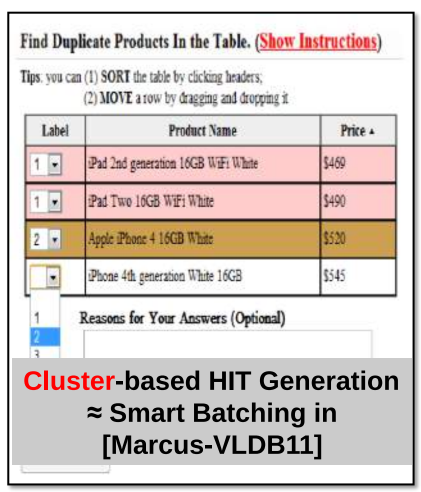
(a) Remove the pairs whose likelihood < 0.3 (b) Generate HITs to verify the pairs of records (c) Output matching pairs

Main Issue: HIT Generation Problem

Join [Wang-VLDB12]



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



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

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) *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

$(r_1, r_2, 0.57)$
$(r_4, r_6, 0.50)$
$(r_1, r_7, 0.43)$
$(r_3, r_4, 0.43)$
$(r_4, r_7, 0.43)$
$(r_8, r_9, 0.43)$
$(r_2, r_3, 0.38)$
$(r_2, r_7, 0.38)$
$(r_3, r_5, 0.38)$
$(r_4, r_5, 0.38)$

***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

Summary of Part I

- 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/

PART II: CROWDSOURCED DATA MINING

Data Everywhere

The amount and diversity of Data being generated and collected is exploding

Web pages, Sensors data, Satellite pictures, DNA sequences, …

From Data to Knowledge

Buried in this flood of data are the keys to

- New **economic** opportunities
- Discoveries in **medicine, science and the humanities**
- Improving **productivity & efficiency**

However, raw data alone is not sufficient!!!

We can only make sense of our world by turning this data into knowledge and insight.

The research frontier

- Knowledge representation.
- knowledge collection, transformation, integration, sharing.
- knowledge discovery.

We focus today on human knowledge

Think of humanity and its collective mind expanding…

Data Mining with/from the Crowd

Challenges: (very) brief overview

- What questions to ask?
  [SIGMOD13, VLDB13, ICDT14, SIGMOD14]
- How to define & determine correctness of answers?
  [ICDE11, WWW12, EDBT15]
- Who to ask? how many people? How to best use the resources?
  [ICDE12, VLDB13, ICDT13, ICDE13]

Association Rule Mining

Semi-supervised Learning

Classification

Clustering

Data Mining with/from the Crowd

Challenges: (very) brief overview

- What questions to ask?
  [SIGMOD13, VLDB13, ICDT14, SIGMOD14]
- How to define & determine correctness of answers?
  [ICDE11, WWW12, EDBT15]
- Who to ask? how many people? How to best use the resources?
  [ICDE12, VLDB13, ICDT13, ICDE13]

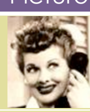
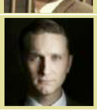
Association Rule Mining

Semi-supervised Learning

Classification

Clustering

A simple example – crowd data sourcing (Qurk)

name	Picture
Lucy	
Don	
Ken	
…	…

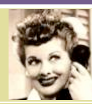
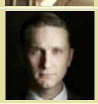
The goal:
Find the names of all the women in the **people** table

```
SELECT name
FROM people p
WHERE isFemale(p)
```

isFemale(%name, %photo)
Question: “Is %name a female?”, %photo
Answers: “Yes”/ “No”

86

A simple example – crowd data sourcing

name	Picture
Lucy	
Don	
Ken	
…	…

Crowd Mining: Crowdsourcing in an open world

- Human knowledge forms an open world
- Assume we want to find out what is *interesting* and *important* in some domain area

  Folk medicine, people's habits, …
- What questions to ask?

Back to classic databases...

- Significant data patterns are identified using data mining techniques.
- A useful type of pattern: *association rules*
  - E.g., *stomach ache* → *chamomile*
- Queries are dynamically constructed in the learning process
- **Is it possible to mine the crowd?**

Turning to the crowd

Let us model the history of every user as a *personal database*

Treated a sore throat with garlic and oregano leaves…
Treated a sore throat and low fever with garlic and ginger …
Treated a heartburn with water, baking soda and lemon…
Treated nausea with ginger, the patient experienced sleepiness…
…

- Every case = a *transaction* consisting of *items*
- Not recorded anywhere – a hidden DB
  - It is hard for people to recall many details about many transactions!
  - But … they can often provide summaries, in the form of personal rules

    "To treat a sore throat I often use garlic"

Two types of questions

- Free recollection (mostly simple, prominent patterns)

  → **Open questions**

  Tell me about an illness and how you treat it

  "I typically treat nausea with ginger infusion"

- Concrete questions (may be more complex)

  → **Closed questions**

  When a patient has both headaches and fever, how often do you use a willow tree bark infusion?

We use the two types interleavingly.

Contributions (at a very high level)

- **Formal model** for crowd mining; allowed questions and the answers interpretation; personal rules and their overall significance.
- **A Framework** of the generic components required for mining the crowd
- **Significance and error estimations.**
  [and, how will this change if we ask more questions…]
- **Crowd-mining algorithms**
- **[Implementation & benchmark.** synthetic & real data/ people]

The model: User support and confidence

- A set of users U
- Each user $u \in U$ has a **(hidden!)** transaction database D_u
- Each rule $X \rightarrow Y$ is associated with:

user support
$$\mathrm{supp}_u(X \rightarrow Y) := \frac{|\{t \in D_u | X \cup Y \subseteq t\}|}{|D_u|}$$

user confidence
$$\mathrm{conf}_u(X \rightarrow Y) := \frac{|\{t \in D_u | X \cup Y \subseteq t\}|}{|\{t \in D_u | X \subseteq t\}|}$$

Model for closed and open questions

- **Closed questions:** X $\rightarrow^{?}$ Y
  - **Answer:** (approximate) user support and confidence
- **Open questions:** ? $\rightarrow^{?}$?
  - **Answer:** an arbitrary rule with its user support and confidence

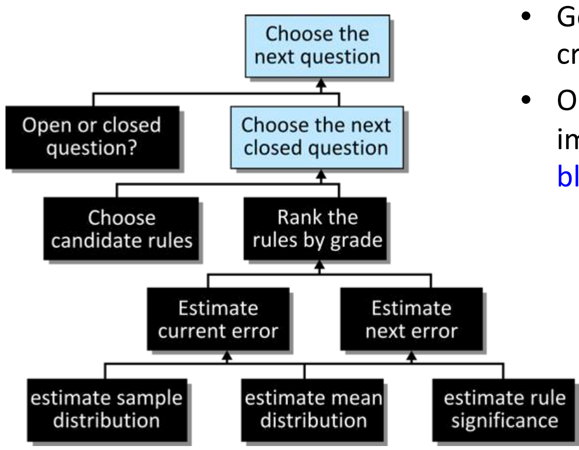
"I typically have a headache once a week. In 90% of the times, coffee helps.

$$\mathrm{supp}_u(\text{headache} \rightarrow \text{coffee}) = \frac{1}{7} \cdot \frac{9}{10} \qquad \mathrm{conf}_u(\text{headache} \rightarrow \text{coffee}) = \frac{9}{10}$$

Significant rules

- **Significant rules**: Rules were the mean user support and confidence are above some specified thresholds Θ_s, Θ_c.
- **Goal**: identifying the significant rules while asking the smallest possible number of questions to the crowd

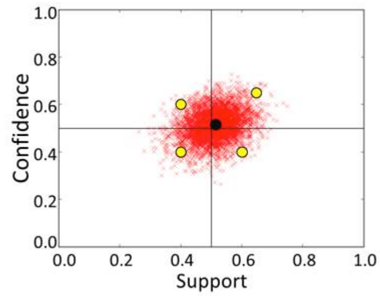
Framework components



- Generic framework for crowd-mining
- One particular choice of implementation of each black boxes

Estimating the mean distribution

- Treating the current answers as a random sample of a hidden distribution g_r, we can approximate the distribution of the hidden mean f_r
- μ – the sample average
- Σ – the sample covariance
- K – the number of collected samples



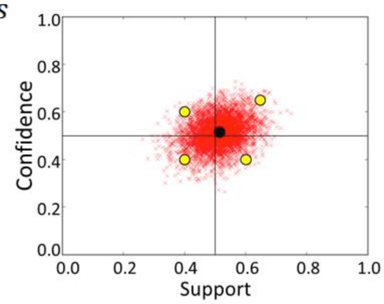
$$f_r \sim N\left(\mu, \frac{\Sigma}{K}\right)$$

- In a similar manner we estimate the hidden distribution g_r

Rule Significance and error probability

- Define M_r as the probability **mass above both thresholds** for rule r

$$M_r = \int_{\Theta_s}^{1}\int_{\Theta_c}^{1} f_r(s,c)\mathrm{d}c\mathrm{d}s$$



- r is significant if M_r is greater than 0.5
- The error prob. is the **remaining mass**
- Estimate how error will change if **another question is asked**
- Choose rule **with largest error reduction**

Completing the picture (first attempt…)

- Which rules should be considered?

  Similarly to classic data mining (e.g. Apriori)
  Start with small rules, then expend to rules similar to significant rules

- Should we ask an open or closed question?

  Similarly to sequential sampling
  Use some fixed ratio of open/closed questions to balance the tradeoff between precision and recall

Semantic knowledge can save work

Given a taxonomy of is-a relationships among items, e.g. **espresso is a coffee**

frequent({headache, espresso}) ⇒ *frequent*({headache, coffee})

Advantages

- Allows inference on itemset frequencies
- Allows avoiding semantically equivalent itemsets
  {espresso}, {espresso, coffee}, {espresso, beverage}…

Completing the picture (second attempt…)

How to measure the efficiency of Crowd Mining Algorithms ???

- Two distinguished cost factors:
  - **Crowd complexity:** # of crowd queries used by the algorithm
  - **Computational complexity:** the complexity of computing the crowd queries and processing the answers

  [Crowd comp. lower bound is a trivial computational comp. lower bound]

- There exists a **tradeoff** between the complexity measures
  - Naïve questions selection -> more crowd questions

Complexity boundaries

- Notations:
  - **$|\Psi|$** - the taxonomy size
  - **$|I(\Psi)|$** - the number of itemsets (modulo equivalences)
  - **$|S(\Psi)|$** - the number of possible solutions
  - Maximal Frequent Itemsets (**MFI**), Minimal Infrequent Itemsets (**MII**)

		W.r.t. the Input	W.r.t. the Output
Crowd	Lower	$\Theta(\log\|S(\Psi)\|)$	$\Omega(\|mfi\|+\|mii\|)$
	Upper		$O(\|\Psi\|\cdot(\|mfi\|+\|mii\|))$
Comp.	Lower	$\Omega(\log\|S(\Psi)\|)$	EQ-hard
	Upper	$O\left(\|I(\Psi)\|\cdot\left(\|\Psi\|^2+\|I(\Psi)\|\right)\right)$	$O\left(\|I(\Psi)\|\cdot\left(\|\Psi\|^2+\|mfi\|+\|mii\|\right)\right)$

$|I(\Psi)| \le 2^{O(|\Psi|)}$, $|S(\Psi)| \le 2^{O(|I(\Psi)|)}$

Now, back to the bigger picture…

The user's question in natural language:

"I'm looking for activities to do in a child-friendly attraction in New York, and a good restaurant near by"

Some of the answers:

"You can go bike riding in Central Park and eat at Maoz Vegetarian.
Tips: Rent bikes at the boathouse"

"You can go visit the Bronx Zoo and eat at Pine Restaurant.
Tips: Order antipasti at Pine.
Skip dessert and go for ice cream across the street"

Pros and Cons of Existing Solutions

- Web Search returns valuable data
  - Requires further reading and filtering
    - Not all restaurants are appropriate after a sweaty activity
  - Can only retrieve data from existing records
- A forum is more likely to produce answers that match the precise information need
  - The #of obtained answers is typically small
  - Still requires reading, aggregating, identifying consensus…
- Our new, alternative approach: **crowd mining!**

Additional examples

A dietician may wish to study the culinary preferences in some population, focusing on food dishes that are rich in fiber

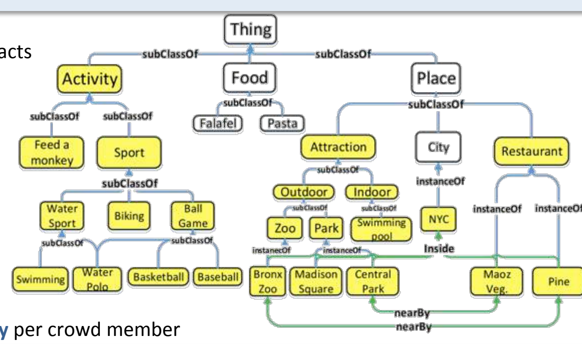
A medical researcher may wish to study the usage of some ingredients in self-treatments of bodily symptoms, which may be related to a particular disease

To answer these questions, one has to combine

- General, ontological knowledge
  - E.g., the geographical locations of NYC attractions
- And personal, perhaps unrecorded knowledge about people's habits and preferences
  - E.g., which are the most popular combinations of attractions and restaurants matching Ann's query

Formal Model Based on RDF

Ontology of general facts



DB of personal history per crowd member

T1	I visited the Bronx Zoo and ate pasta at Pine on April 5th	[Visit doAt Bronx_Zoo]. [Pasta eatAt Pine]
T2	I played basketball in Central Park on April 13th	[Basketball playAt Central_Park]
T3	I played baseball in Central Park and ate falafel at Maoz Veg. on April 27th	[Baseball playAt Central_Park]. [Falafel eatAt Maoz_Veg]
	…	…

A Declarative Mining Language

- OASSIS-QL – Ontology-ASSISted crowd mining Query Language
- For specifying information needs in a precise manner
  - Based on SPARQL, the RDF query language

```
SELECT VARIABLES
WHERE
    {$w subClassOf* Attraction.
     $x instanceOf $w.
     $x inside     NYC.
     $y subClassOf* Activity.
     $z instanceOf Restaurant.
     $z nearBy     $x}
SATISFYING
     {$y+ doAt     $x.
     []  eatAt     $z.
     MORE}
     WITH SUPPORT = 0.03
```

Evaluated over the ontology, to identify candidate data patterns

Retain the patterns that are significant for the crowd, and find additional advice

Evaluation with the Crowd

```
SELECT VARIABLES
WHERE
    {$w subClassOf* Attraction.
     $x instanceOf $w.
     $x inside     NYC.
     $y subClassOf* Activity.
     $z instanceOf Restaurant.
     $z nearBy     $x}
SATISFYING
     {$y+ doAt     $x.
     []  eatAt     $z.
     MORE}
     WITH SUPPORT = 0.03
```

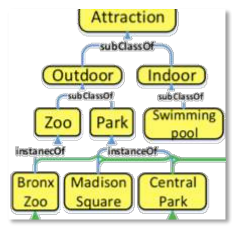
$x= Central_Park,
$y = Basketball

“How often do you play basketball in Central Park?”

“Every week.” (support = 1/7)
12/365)

Efficient Query Evaluation Algorithm

- We want to minimize the number of questions to the crowd
- We define a semantic subsumption partial order over terms, facts, and fact-sets
- Used for
  - Pruning the search space
  - Compact output representation




```
Biking doAt Park
```

```
Biking doAt Central_Park
```

```
Biking     doAt   Central_Park.
Basketball playAt Central_Park
```


Additional Aspects of the Algorithm

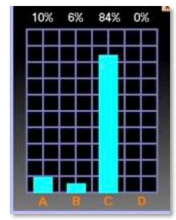
- Open questions – letting crowd members specify patterns

  "What else do you do when you play basketball in Central Park?"

  The answers help speeding up the mining process.
- Asking a sequence of questions "in context"
- Quick pruning of irrelevant items by crowd members
- Multiple crowd workers in parallel
- Output quality assurance

Can we trust the crowd ?

The common solution: ask multiple times



We may get different answers

- Legitimate diversity
- Wrong answers/lies

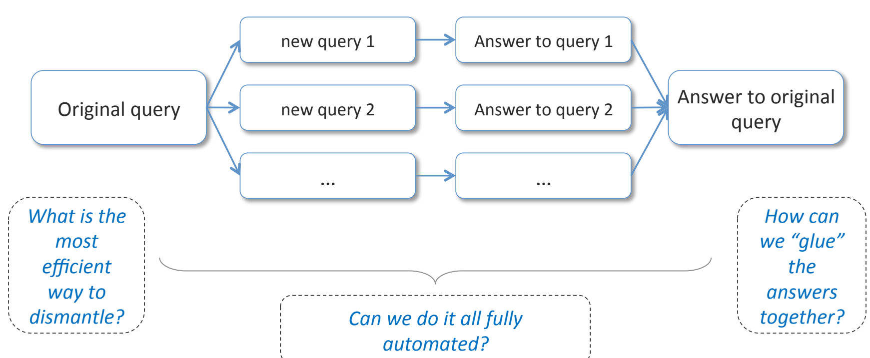
Can we trust the crowd ?

Things are non trivial …

- Different experts for different areas
- "Difficult" questions vs. "simple" questions
- Data in added and updated all the time
- Optimal use of resources… (both machines and human)

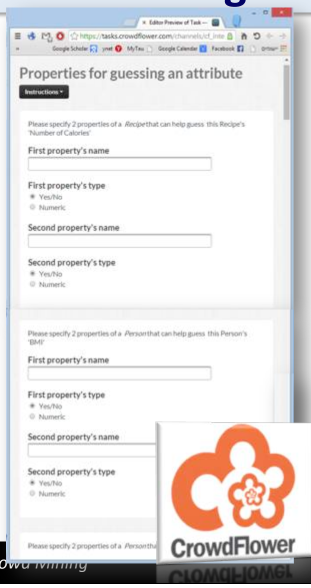
Solutions based on
- Statistical mathematical models
- Declarative specifications
- Provenance

Dismantle Queries into Easier Ones, then Reassemble



?What do we mean by "Dismantle"

Dismantling – Some Real-Life Examples



Person's age

wrinkles, grey hair, old, height, good look, children, dark skin, has work, male, over 35, weight, glasses, …

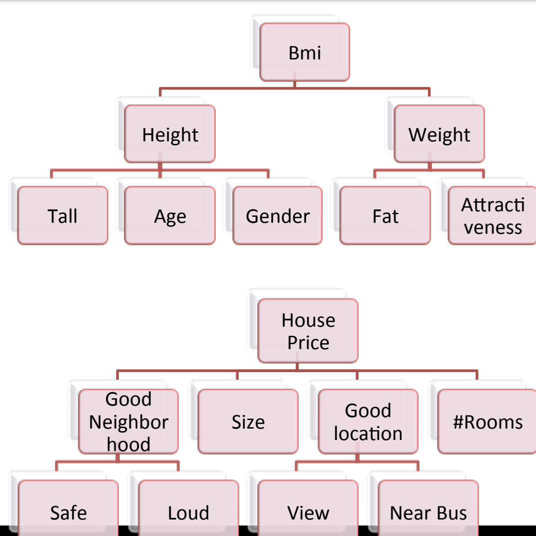
Recipe's #calories

fat amount, #ingredients, healthy, portion size, sugar amount, vegetarian, oily, dietetic,…

House's price

good location, age, size, #room, good neighborhood, good view, renovated, nice, good exterior condition, …

Dismantling – Algorithmically



The desired output consists of two parts (informal)

1. How many questions to ask on each attribute (a Budget distribution b)
2. How to compose the answers (a Linear regression l)

Q = Select name, BMI from pictures

- BMI$^{(20)}$
- 0.7BMI$^{(10)}$ + 0.1Weight$^{(6)}$ + 6.5Fat$^{(4)}$ + 4.06
- 0.2BMI$^{(4)}$ + 9.5Heavy$^{(3)}$ + 0.2Weight$^{(2)}$ + 0.4GoodBuilt$^{(2)}$ + 4.9Over200Pounds$^{(4)}$ - 0.3FairLooking$^{(1)}$ - 2.7GoodFacialFeatures$^{(1)}$ -0.2GoodPhysicalFeatures$^{(1)}$ + 0.6HasWork$^{(1)}$ - 0.1WorksOut$^{(1)}$ + 12.6

More formally

Input

- Objects $o \in O$
- Attributes $a \in A$
- Query Q
- Crowd Questions
  - Value, Dismantling, Example
- Budgets
  - per-object budget B_{obj}
  - pre-processing budget B_{prc}

Output

- Find b, l
  - $b: A \rightarrow$
  - $l: A \rightarrow$
- That minimize
  - $Er = {}_{o \in O}\left[\left(\sum_{a \in A} l(a)b(a) - Q(o)\right)^2\right]$
- Subject to
  - $\sum_{a \in A} b(a) = B_{obj}$
  - $\sum_{\text{pre-processing tasks}} \text{Cost(task)} < B_{prc}$

Solution components

Choosing Dismantling Questions

- Based on probability of new answer (attribute) and expected answer's correlation

Estimating Statistics

- Inductive solution

Calculating b

- [Sabato,Kalai ICML'13]
- Adaptations to match our scenario

Calculating l

- A well studied problem
- Collecting dataset based on calculated heuristics

Deciding When to Stop

- Minimal learning budget as a function of the number of attributes

Crowd Mining 119

Summary – Crowd Based Data Mining

The crowd is an incredible resource!

"Computers are useless, they can only give you answers"
- Pablo Picasso

But, as it seems, they can also ask us questions!

Many challenges:

- (very) interactive computation
- A huge amount of (human) data to mine
- Varying quality and trust

Crowd Mining 120

PART III: CROWDSOURCED SOCIAL APPLICATIONS

Managing Wisdom of Online Social Crowds

- **Whom to Ask [VLDB'12]**
- WiseMarket [KDD'13]
- COPE [KDD'14]
- TCS [KDD'14]

"If we drive, can we get to Victoria Peak from HKUST in one hour?"

"Yes or No?"

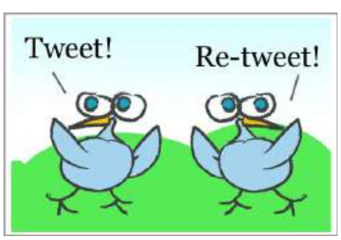
- Minor as dressing for a banquet
- Major as prediction of macro economy trends

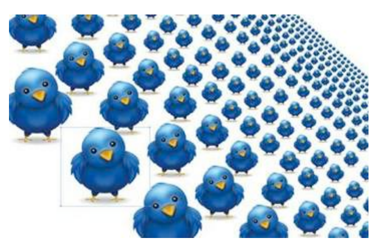
"two-option decision making tasks"

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

Microblog Users

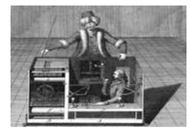
- Simple
  - 140 characters
  - ‘RT’ + ‘@’
- But comprehensive
  - Large network
  - Various backgrounds of users





Why Microblog Platform?

Twitter



AMT

	Social Media Network	General Purpose Platform
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

Whom to Ask?

- "Which venue held the latest International Film Festival in Hong Kong?"

Andy Lau

"HK Coliseum"

Cecilia Cheung

"? ? ?"

Nicholas Tse

"Hong Kong Cultural Centre"

Jackie Chan

"HK Coliseum"

Whom to Ask?

- "What's the next breakthrough in Big Data"

Andy Lau

"? ? ?"

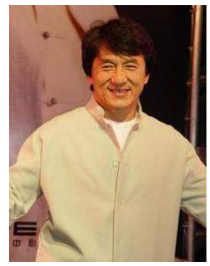
Cecilia Cheung

"? ? ?"

Nicholas Tse

"? ? ?"

Jackie Chan

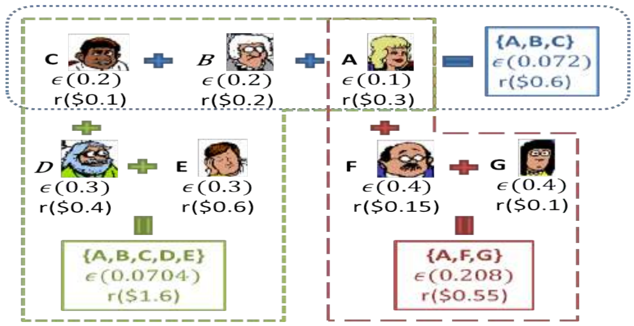


"? ? ?"

Running Example

- "Is it possible to swim in the Silverstrand Beach in August?"

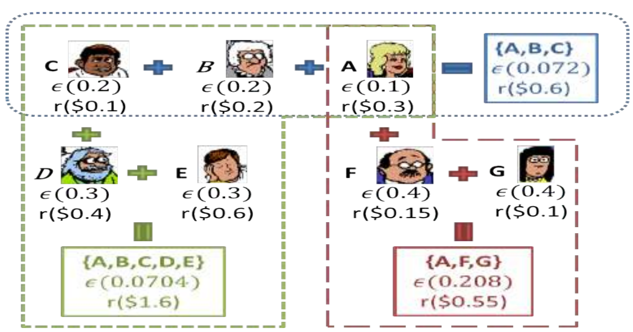
Motivation – Jury Selection Problem Running Case(1)



"Is it possible to swim in the Silverstrand Beach in August?"

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

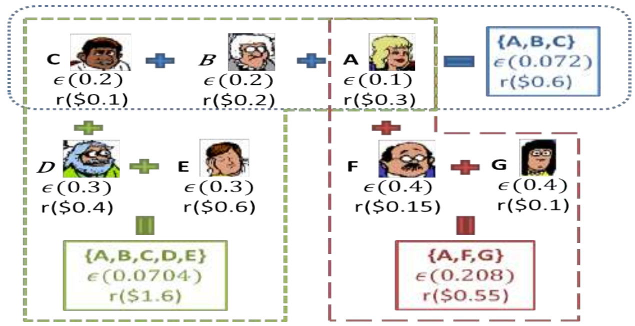
Motivation – Jury Selection Problem Running Case(2)



"Is it possible to swim in the Silverstrand Beach in August?"

- ε: error rate of an individual
- r: requirement of an individual, can be virtual
- Majority voting to achieve the final answer

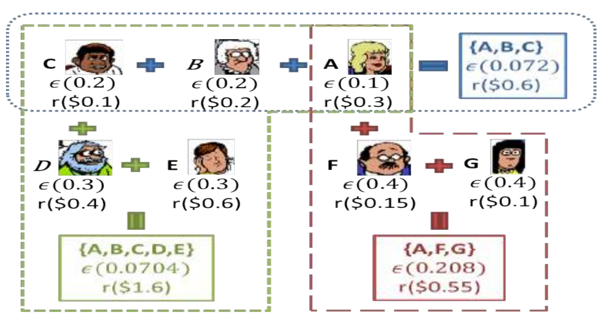
Motivation – Jury Selection Problem Running Case(3)



"Is it possible to swim in the Silverstrand Beach in August?"

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

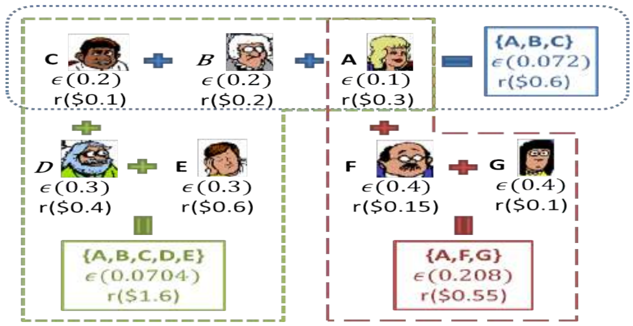
Motivation – Jury Selection Problem Running Case(4)



"Is it possible to swim in the Silverstrand Beach in August?"

- 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

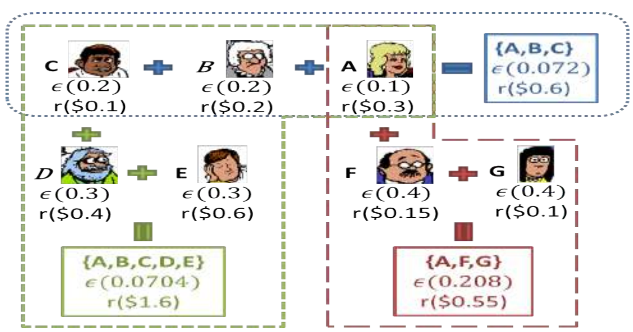
Motivation – Jury Selection Problem Running Case(5)



"Is it possible to swim in the Silverstrand Beach in August?"

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

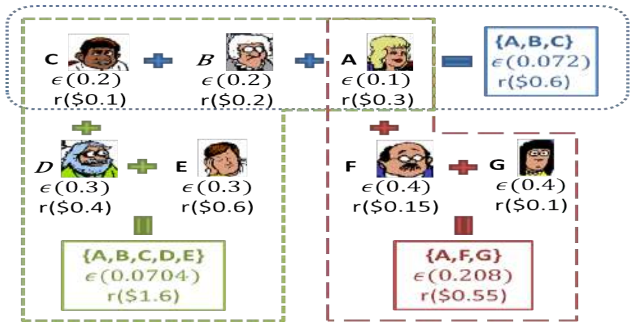
Motivation – Jury Selection Problem Running Case(6)



"Is it possible to swim in the Silverstrand Beach in August?"

- 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(7)



"Is it possible to swim in the Silverstrand Beach in August?"

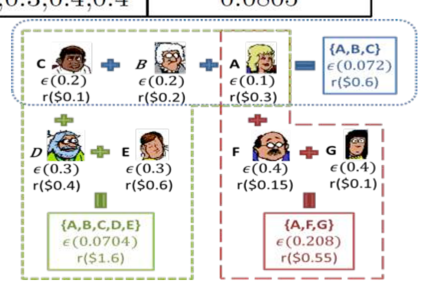
- So just pick up the best combination?
  - JER(A,B,C,D,E)=0.0704
  - R(A,B,C,D,E) = $1.6 > budget($1.0)

Motivation – Jury Selection Problem Running Case(8)

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:

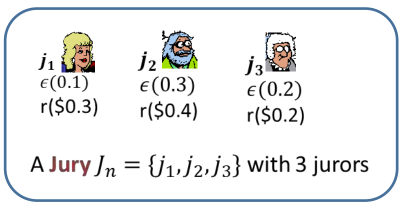
The reliability of voting.

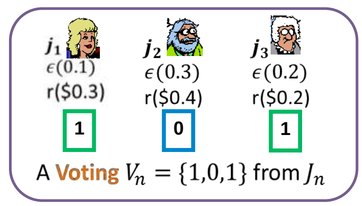


Problem Definition

- Jury and Voting

Definition 1 (Jury). *A jury $J_n = \{j_1, j_2, \cdots, j_n\} \subseteq S$ is a set of jurors with size n that can form a voting.*





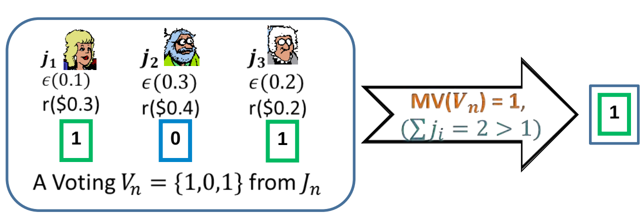
Definition 2 (Voting). *A voting V_n is a valid instance of a jury J_n with size n, which is a set of binary values.*

Problem Definition

- Voting Scheme

Definition 3 (Majority Voting - MV). *Given a voting V_n with size n, Majority Voting is defined as*

$$MV(V_n) = \begin{cases} 1 & \text{if } \sum j_i \geq \frac{n+1}{2} \\ 0 & \text{if } \sum j_i \leq \frac{n-1}{2} \end{cases}$$

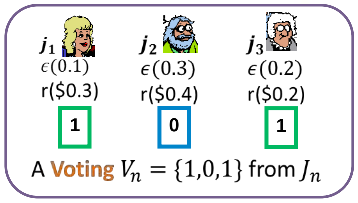


Problem Definition

- Invididual Error-rate

DEFINITION 4 (INDIVIDUAL ERROR RATE - ϵ_i). *The individual error rate ϵ_i is the probability that a juror conducts a wrong voting. Specifically*

$$\epsilon_i = Pr(vote\ otherwise | a\ task\ with\ ground\ truth\ A)$$



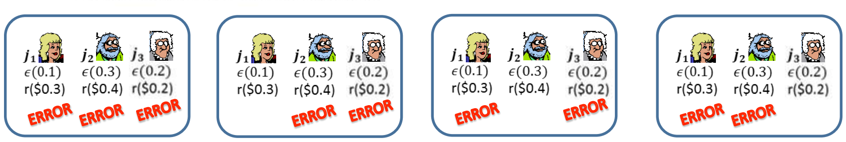
DEFINITION 5 (CARELESSNESS - C). *The Carelessness C is defined as the number of mistaken jurors in a jury J_n during a voting, where $0 \le C \le n$.*

Problem Definition

DEFINITION 6 (JURY ERROR RATE - $JER(J_n)$). *The jury error rate is the probability that the Carelessness C is greater than $\frac{n+1}{2}$ for a jury J_n, namely*

$$JER(J_n) = \sum_{k=\frac{n+1}{2}}^{n} \sum_{A \in F_k} \prod_{i \in A} \epsilon_i \prod_{j \in A^c} (1 - \epsilon_j)$$
$$= \Pr(C \ge \frac{n+1}{2} | J_n)$$

where F_k is all the subsets of S with size k and ϵ_i is the individual error rate of juror j_i.



$JER(J_3)$ = 0.1*0.3*0.2 + (1-0.1)*0.3*0.2 + 0.1*(1-0.3)*0.2 + 0.1*0.3*(1-0.2)

$JER(J_3)$ = 0.1*0.3*0.2 + (1-0.1)*0.3*0.2 + 0.1*(1-0.3)*0.2 + 0.1*0.3*(1-0.2)
= 0.029

Problem Definition

- Crowdsourcing Models(model of candidate microblog users)

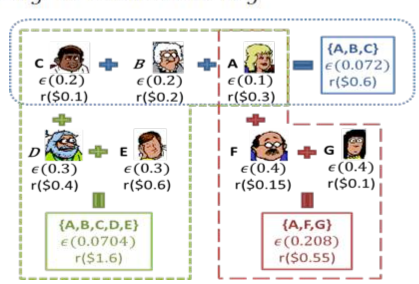
DEFINITION 7 (ALTRUISM JURORS MODEL - ALTRM). *While selecting a jury J from all candidate jurors (choosing a subset $J \subseteq S$), any possible jury is allowed.*

DEFINITION 8 (PAY-AS-YOU-GO MODEL - PAYM). *While selecting a jury J from all candidate jurors (choosing a subset $J \subseteq S$), each candidate juror j_i is associated with a payment requirement r_i where $r_i \geq 0$, the possible jury J is allowed when the total payment of J is no more than a given budget B, namely $\sum_{\forall j_i \in J} r_i \leq B$.*

Problem Definition

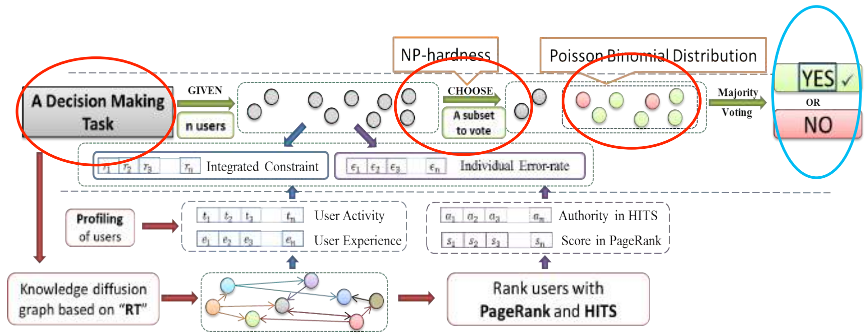
- Jury Selection Problem(JSP)

DEFINITION 9 (JURY SELECTION PROBLEM - JSP). *Given a candidate juror set S with size $|S| = N$, a budget $B \geq 0$, a crowdsourcing model(AltrM or PayM), the Jury Selection Problem(JSP) is to select a jury $J_n \subseteq S$ with size $1 \leq n \leq N$, that J_n is allowed according to crowdsourcing model and $JER(J_n)$ is minimized.*

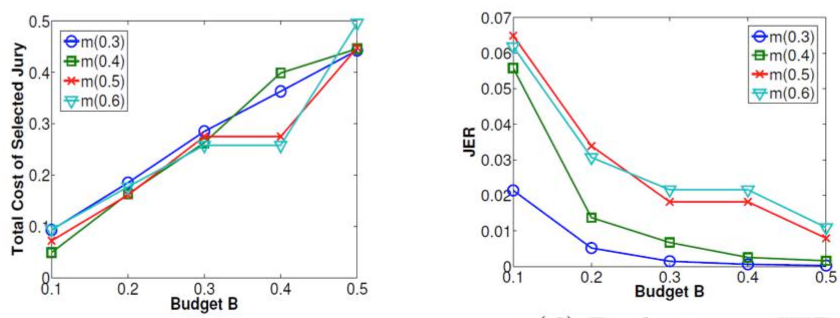


We hope to form a Jury J_n, allowed by the budget, and with lowest JER

Framework



Experimental Studies



(c) Budget v.s. Total Cost

(d) Budget v.s. JER

Managing Wisdom of Online Social Crowds

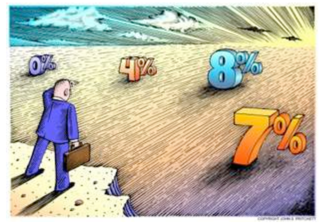
- Whom to Ask [VLDB'12]
- **WiseMarket [KDD'13]**
- COPE [KDD'14]
- TCS [KDD'14]

WiseMarket

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

Market

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



- 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

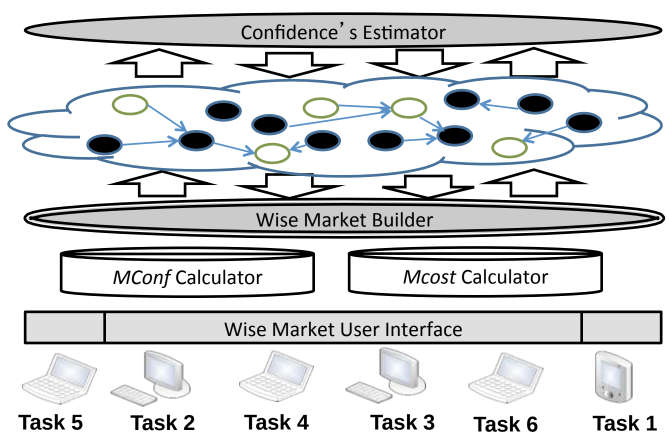
Why WiseMarket?

- 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

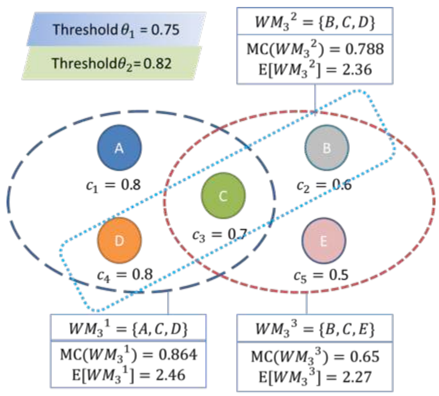
- So How Does it Run?

How Does it Run?



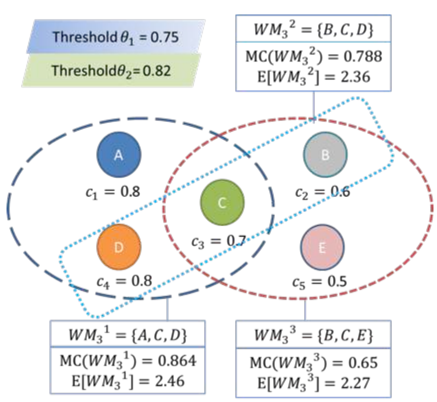
Choose the best investors to build a market

Running Example



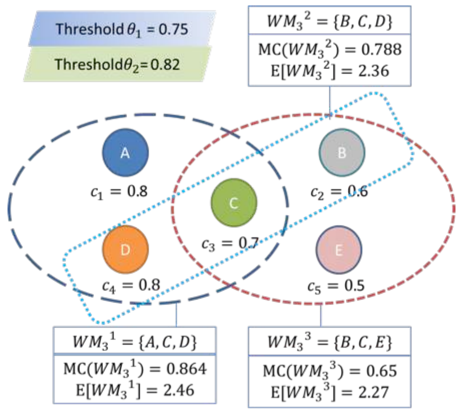
- **Based on Majority Voting, to achieve an overall confidence of θ, how should we build a most economical market?**

Running Example



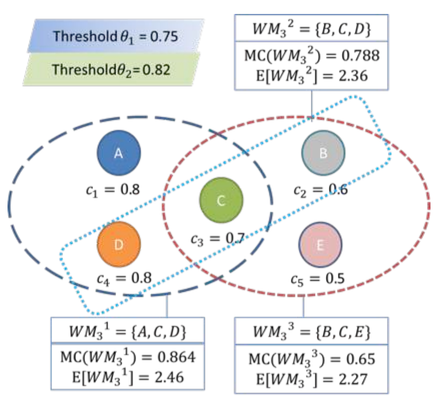
- "Economical" means minimum expected cost
- Each winner is getting a unit-reward
- Only winners get reward

Running Example



- Each investor is associated with a confidence *c*
- The market is measured according to Market Confidence *MC*

Running Example



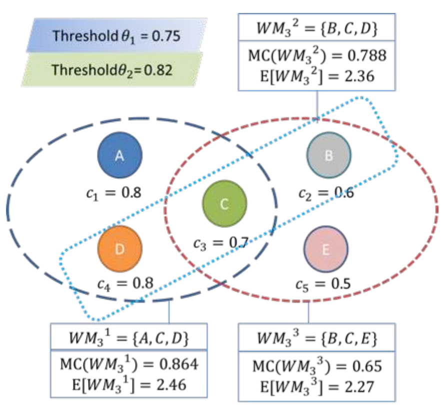
- If we choose {A,C,D}, the MC is 0.864 based on Majority Voting.
- The Cost is ?

Running Example

Case	Correct	Wrong	Prob.	Cost
1	{A,C,D}	∅	0.448	3
2	{A,C}	{D}	0.112	2
3	{A,D}	{C}	0.192	2
4	{C,D}	{A}	0.112	2
5	{A}	{C,D}	0.048	2
6	{C}	{A,D}	0.028	2
7	{D}	{A,C}	0.048	2
8	∅	{A,C,D}	0.012	3

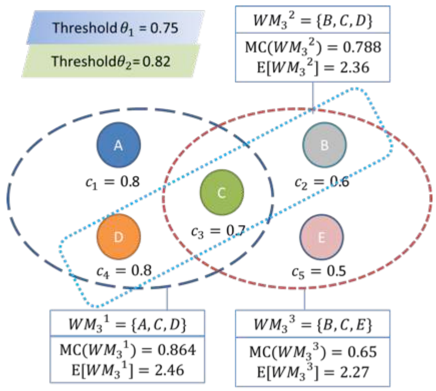
- $E[Cost] = 2.46 = 3 \cdot \sum(Prob.\, of\ Case\ 1\ and\ 8) + 2 \cdot \sum(Prob.\, of\ Case\ 2\ to\ 7)$

Running Example



- **How about {B, C, D}?**
- **The expected cost is lower (2.36), but the market confidence disagrees with the threshold θ_2=0.82.**

Running Example



- How about others?
- There are too many possible combinations, we need better solutions

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*

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

- 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:*

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

- The market size should be ODD to feature Majority Voting.

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*:*

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

- 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*:*

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

- Expected Market Cost

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

- The lower the expected cost, the more economical the market.

Problem Definition

- Effective Market Problem

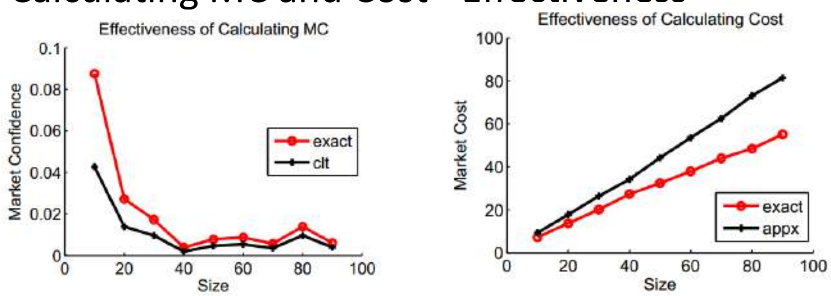
DEFINITION 6 (EFFECTIVE MARKET PROBLEM). *Given a set of investors* $I = \{\iota_1, \ldots, \iota_N\}$ *with size* N*, a Market Confidence threshold* θ*, the Effective Market Problem(*EMP*) is to find a subset of all investors* $WM_n \subseteq I$*, so that:*

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

A market *BUILDER* for tasks holders

Experimental Studies

- Calculating MC and Cost - Effectiveness



- CLT converges while size grows larger
- Appx algorithms exhibit lower appx ratio
  - $(3 - 2\theta)$

Managing Wisdom of Online Social Crowds

- Whom to Ask [VLDB'12]
- WiseMarket [KDD'13]
- **COPE [KDD'14]**
- TCS [KDD'14]

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

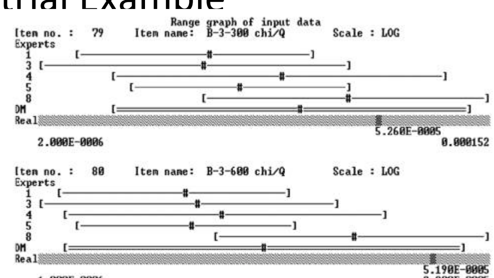


Figure 1: Example of Opinion Elicitation of five participants over two variables(NRC-EU accident uncertainty analysis [4])

  - Specifying (uniform) variable distribution over a range.
  - Multiple workers are involved to express their opinions.
  - The opinions are aggregated afterwards.

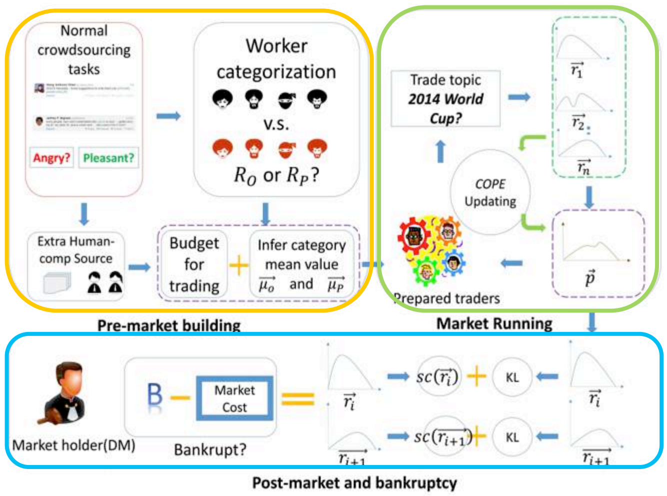
Challenges

- No ground truth
  - Intrinsic opinion on an m outcomes event
    - $d=\{d_1, d_2, \ldots, d_m\}$
  - Reported by the worker
    - $r=\{r_1, r_2, \ldots, r_m\}$
  - In some cases, $r \neq d$
- Reasons for such insincerity
  - Carelessness
  - Indifference

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

- Payoff
  - Payoff depends on the contribution on the aggregated opinion
  - COPE adopts KL-divergence to measure the contribution
    - i.e. relative entropy
    - Naturally suits the Bayesian Updating Scheme
    - In accordance to the measure of goodness (entropy) of a report

DEFINITION 3 (PAYOFF). *Given the market estimation $\vec{p}$, a trader T_i, whose proposed report is $\vec{r_i}$, will receive a payoff M_i, when the market is closed.*

$$M_i = C_i \cdot \frac{Odd}{D_i + 1} = C_i \cdot \frac{Odd}{D(\vec{r_i}||\vec{p}) + 1} \quad (2)$$

C_i is the invested capital of T_i, and Odd is the preset parameter such that at most a trader could earn $Odd \times C_i$ as payoff.

COPE – The Design

- Payoff Range
  - The traders may lose their seed capitals
  - The traders maximize their payoff when their reports approximate the global report

LEMMA 1 (PAYOFF RANGE). *The range of payoff for trader T_i is as follow:*

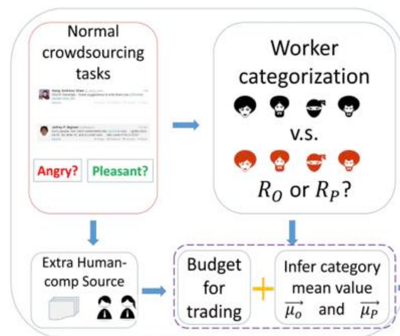
$$0 < M_i \leq Odd \times C_i \quad (3)$$

The maximum equality is observed when $\vec{r_i} = \vec{p}$.

- Goodness of a report
  - Expec $sc(\vec{r_i}) = \sum_j \vec{r_i}[j] \cdot S_j(\vec{r_i}[j]) = \sum_j \vec{r_i}[j] \log \vec{r_i}[j]$ rithm form

COPE – The Design

- Pre-market Building
  - Generate Seed Capital
    - Promised Salaries as initial funds
  - Tendency Evaluation
    - Optimistic
    - Pessimistic
    - Group Mean



Pre-market building

For bias adjustment during an upc

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

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
    - Com extremes

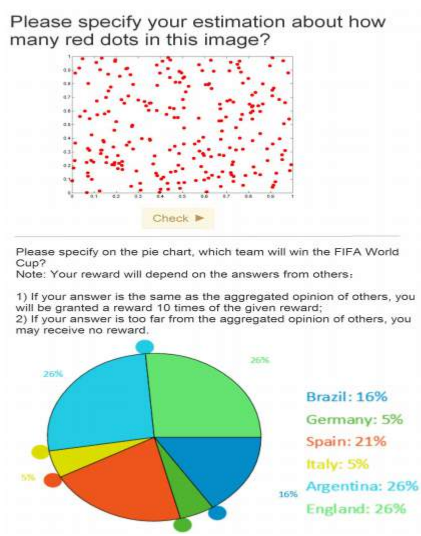
$$p^* = \Pr(\vec{p}|\vec{r}) \propto \frac{\Pr(\vec{p})L(\vec{r}|\vec{p})}{\Pr(\vec{r})}$$

* C. Genest and J. V Zidek. Combining probability distributions: A critique and an annotated bibliography. Statistical Science, 1986

COPE –The Running Mechanism

- Simple Strategy
  – Calculate market cost (MC) every time new report is updated
  – Stop when MC>B
  – May be inflexible during real-time running
  – Time complexity O(|S|n)
- Slope-based Strategy
  – Start to calculate exact MC when upper edge exceeds the budget B
  – Terminate immediately when lower edge exceeds the budget B

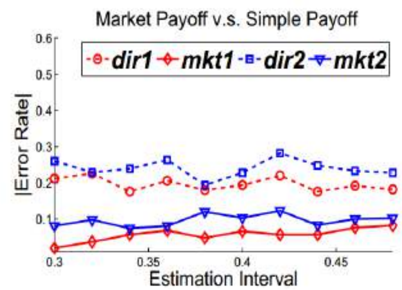
COPE – The Implementation



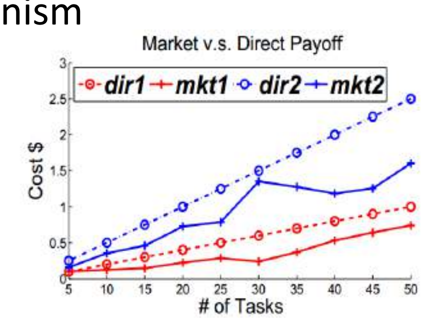
- Premarket Tasks
- Opinion Elicitation
  – Dynamic Chart
  – Kill *probability-phobia*
    - Unwilling or uncomfortable to give numerical probability
  – Workers are informed the payoff method
- Payoff Dispatch
  – Special "*payoff tasks*" are

COPE – The Evaluation

- Merits of Market Mechanism



Error Rate

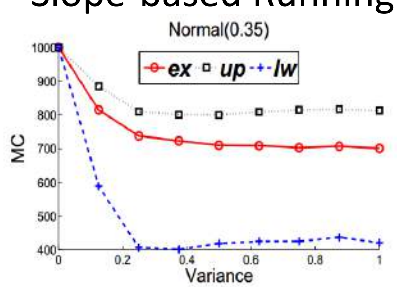


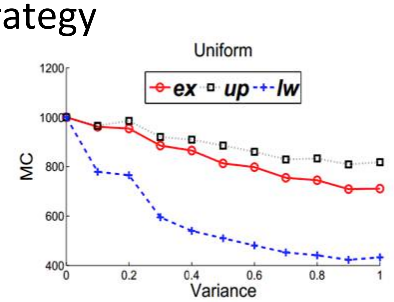
Total Cost

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

COPE – The Evaluation

- Slope-based Running Strategy





  – Tested for both normal and uniform distribution
  – Upper edge and lower edge as the slope range of *MC*
  – The lower the variance, the narrower the slope
    - i.e. when opinions are alike, the market terminates early

Managing Wisdom of Online Social Crowds

- Whom to Ask [VLDB'12]
- WiseMarket [KDD'13]
- COPE [KDD'14]
- **TCS [KDD'14]**

Big Crowd-Oriented Services

- The information services provided by crowdsourcing usually include big task-response pairs.
- It is important to discover hidden rules for the big crowd-oriented service data.

184

Crowd-Oriented Service Data

- A snippet of crowd-oriented service from Stack Overflow

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

185

Crowd-Oriented Service Data

- A snippet of crowd-oriented service from Stack Overflow

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

186

Crowd-Oriented Service Data

- A snippet of crowd-oriented service from Stack Overflow

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

187

Crowd-Oriented Service Data

- A snippet of crowd-oriented service from Stack Overflow

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

188

Crowd-Oriented Service Data

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 18:23:01
R2	T1	Save the image to your sdcard. ...	2014-02-01 15:01:53
R3	T1	Storing images in your database will…	2014-02-01 16:38:17

189

Crowd-Oriented Service Data

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 18:23:01
R2	T1	Save the image to your sdcard. ...	2014-02-01 15:01:53
R3	T1	Storing images in your database will…	2014-02-01 16:38:17

190

Crowd-Oriented Service Data

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 18:23:01
R2	T1	Save the image to your sdcard. ...	2014-02-01 15:01:53
R3	T1	Storing images in your database will…	2014-02-01 16:38:17

191

Crowd-Oriented Service Data

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 18:23:01
R2	T1	Save the image to your sdcard. ...	2014-02-01 15:01:53
R3	T1	Storing images in your database will…	2014-02-01 16:38:17

192

Characteristic of Crowd-Oriented Service Data-I

- Task-Response Pairs
  - Task-Response Correlation

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 18:23:01
R2	T1	Save the image to your sdcard. ...	2014-02-01 15:01:53
R3	T1	Storing images in your database will…	2014-02-01 16:38:17

Characteristic of Crowd-Oriented Service Data-II

- Big volume
  - Each task may have large amount of responses

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33
…	…	…

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 18:23:01
…	…	…	…
R100	T1	Storing images in your database will…	2014-02-04 11:36:02
…	…	…	…

Characteristic of Crowd-Oriented Service Data-III

- Dynamic Evolution with Time

Timeline

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 17:30:33
R2	T1	Save the image to your sdcard. ...	2014-02-01 10:31:25
R3	T1	Storing images in your database will…	2014-02-01 21:40:01

Task ID	Tasks	Timestamp
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25

2nd Bucket

3rd Bucket

1st Bucket

Characteristic of Crowd-Oriented Service Data-III

- Dynamic Evolution with Time
  - Accumulates as Multiple Consecutive Buckets

Timeline

Task ID	Tasks	Timestamp
T1	Android application database to save images ...	2014-01-31 17:30:33

Response ID	Task ID	Responses	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 17:30:33
R2	T1	Save the image to your sdcard. ...	2014-02-01 10:31:25
R3	T1	Storing images in your database will…	2014-02-01 21:40:01

Task ID	Tasks	Timestamp
T2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
…	…….	….

Challenges

- How to model big crowd-oriented service data

Task-Response Correlation

Responses ID	Task ID	Tasks	Timestamp
R1	T1	Android SQLite database with multiple...	2014-01-31 17:30:33
R2	T1	Save the image to your sdcard. ...	2014-02-01 10:31:25
R3	T1	Storing images in your database will…	2014-02-01 21:40:01

- High training efficiency is important for the big data

- Topics over big crowd-oriented service data are

Problem Definitions

Task ID	Tasks	Timestamp
T_1	Android application database to save images ...	2014-01-31 17:30:33
T_2	How to use simpleadapter with activity Android...	2014-02-02 10:31:25
T_3	How to find mobile phones on hotspot networks in iPhone?	2014-02-03 21:40:01

Response ID	Task ID	Responses	Timestamp
$R_{1,1}$	T_1	Android SQLite database with multiple...	2014-01-31 18:23:01
$R_{1,2}$	T_1	Save the image to your sdcard. ...	2014-02-01 15:01:53
$R_{1,3}$	T_1	Storing images in your database will…	2014-02-01 16:38:17
$R_{3,1}$	T_3	iOS 7 system of Apple devices provide...	2014-02-03 22:14:27

- $(T_1, R_{1,1})$ is a task-response pair.
- In this example, $CS=\{(T_1, R_{1,1}), (T_1, R_{1,2}), (T_1, R_{1,3}), (T_3, R_{3,1})\}$.
- (iPhone, iOS7) is a word-pair in $(T_3, R_{3,1})$.

Problem Definitions

- Topic
  - A semantically coherent topic ϕ is a multinomial distribution of words $\{p(w|\phi)\}_{w \in W}$ with the constraint $\sum_{w \in W} p(w|\phi) = 1$.

- Topic Discovery in Crowd-oriented Service Data
  - Given the input of a crowd-oriented service data *CS*, we are required to infer the latent topics ϕ over in *CS*.

Generative Process of TCS Model

- Each task and response is viewed as a document, respectively.
- TCS shares ingredients with Latent Dirichlet Allocation (LDA):
  - Each topic has a distribution over words;
  - Each document has a distribution over topics;

Algorithm 1: Generative process of TCS

for *each topic* $k \in \{1, \cdots, K\}$ **do**
    draw a word distribution $\phi_k \sim \text{Dirichlet}(\beta)$;
for *each document* $d \in \{1, \cdots, D\}$ **do**
    draw topic distribution $\theta_d \sim \text{Dirichlet}(\alpha)$;
    if *d is a task* **then**
        sample a response d' with regard to the number of sketch pairs between d and d';
    if *d is a response* **then**
        select the corresponding task d';
    generate new document topic distribution θ' by combining θ_d and $\theta_{d'}$;
    for *each sentence* $s \in d$ **do**
        choose a topic $z \sim \text{Multinomial}(\theta')$;
        generate words $w \sim \text{Multinomial}(\phi_z)$;

- If a document d is a task, sample a response from the set of task-response pairs;
- Otherwise, d is a response and select its corresponding task;
- Combine the task and response as a new document and generate the new distribution over topics;
- Each sentence is the basic unit for topic assignment.

201

Challenges of TCS Model

- It is infeasible to count and store frequencies of all word pairs due to the excessively high cost.
  – Our Solution: Only storing significant (frequent) word-pairs and removing extremely infrequent word-pairs.

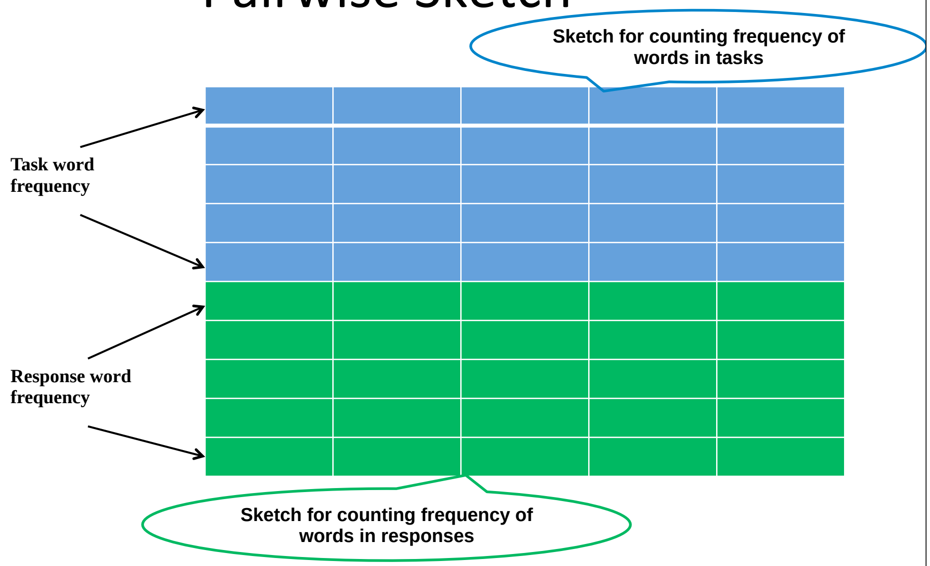
- How to training the TCS model efficiently when the correlation of task-response pair is considered?
  – Our Solution: Speeding up the training and belief updating process according to significant word-pairs.

202

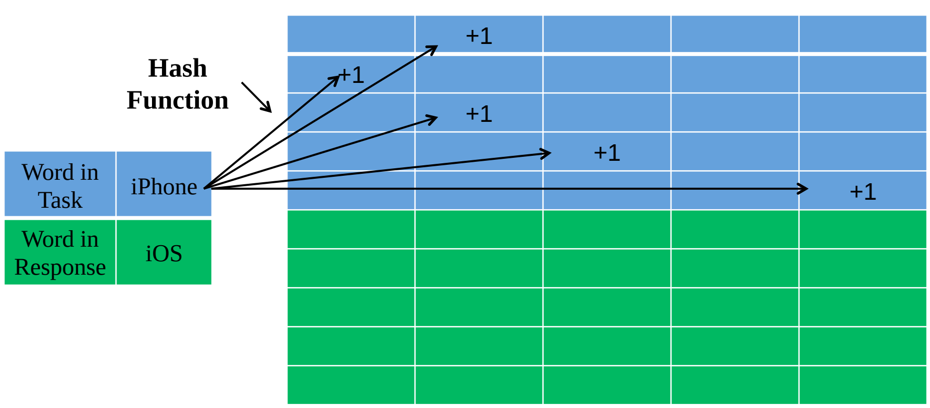
Key ideas of Pairwise Sketch

- Main ideas
  – A sketch-based method
  – Approximate the frequency of word-pairs in tasks and responses with bounded error within a probability.
- Only frequent word-pairs are significant for topic modeling
  – Extremely infrequent word-pairs in tasks and responses are removed.
- Effective Space Complexity
  – O($\frac{1+\delta}{\varepsilon\delta}|T|$)
  – Refer to our paper for the proof if you are interested.

Pairwise Sketch

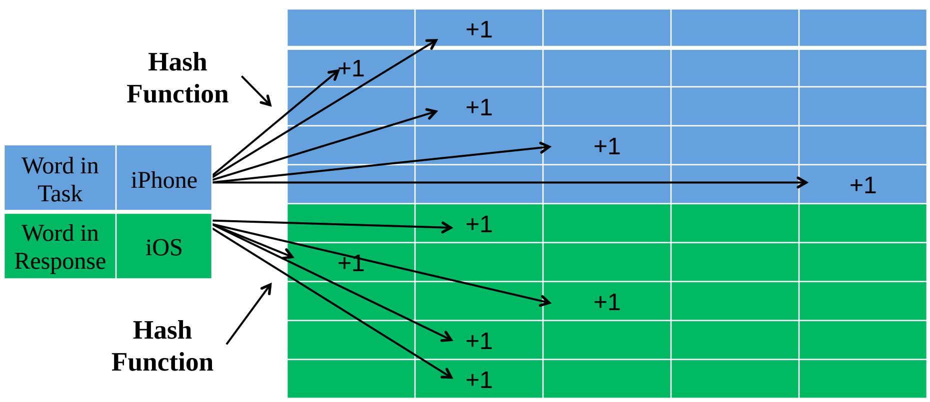


Pairwise Sketch



205

Pairwise Sketch



206

Belief Residual of Sentences

- The belief that a sentence *s* of a specific document *d* is generated by topic *k* is denoted by $\mu^k_{d,s}$
  - $\mu^k_{d,s}$ is calcul...

The estimation of the topic distribution inthe document level.

The estimation of the word distribution in the topic level.

$$\mu^k_{d,s} \propto \frac{(1-\zeta)\mu^k_{d,-s} + \zeta\mu^k_{l_d} + \alpha_k}{\sum_{k'}\left((1-\zeta)\mu^{k'}_{d,-s} + \zeta\mu^{k'}_{l_d} + \alpha_{k'}\right)}$$

$$\frac{\Gamma\left(\sum_{r'}(n_{\cdot,-s,w'}\mu^k_{\cdot,-s,w'} + \beta_{w'})\right)}{\Gamma\left(\sum_{w'}(n_{\cdot,-s,w'}\mu^k_{\cdot,-s,w'} + \beta_{w'} + n_{d,s,w'})\right)}$$

$$\prod_{w\in s}\left(\frac{\Gamma(n_{\cdot,-s,w}\mu^k_{\cdot,-s,w} + \beta_w + n_{d,s,w})}{\Gamma(n_{\cdot,-s,w}\mu^k_{\cdot,-s,w} + \beta_w)}\right)$$

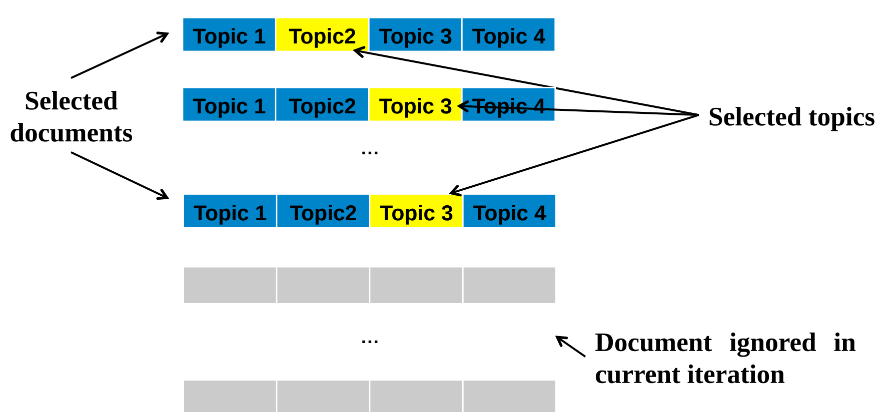
- The belief residual r^k_{ds} between two successive iterations t and $r^k_{d,s} = \left|\mu^k_{d,s}(t) - \mu^k_{d,s}(t-1)\right|$ as follows,

Belief Update Algorithm

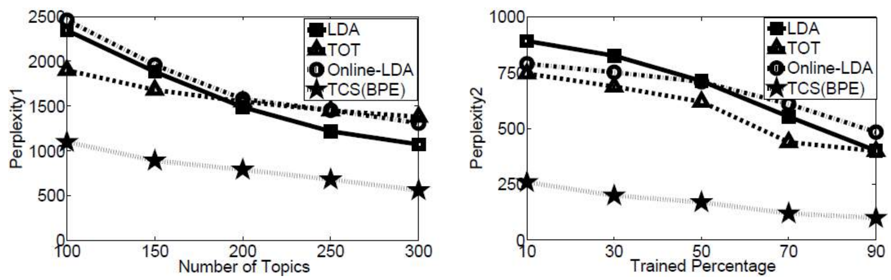
- After each iteration
  - Sort $\boldsymbol{r}_d$ in a descending order for all documents;
  - Select several documents with the largest residuals;
  - For each selected document
    - Sort $\boldsymbol{r}^k_d$ in descending order;
    - Select several topic with the largest residual;
  - Update the corresponding μ^k_{ds};
  - Normalize the corresponding μ^k_{ds};

Belief Update Algorithm

- A Running Example



Experimental Studies: Effectiveness



- TCS demonstrates good performance in terms of perplexity.
- Perplexity1 describes the held-out perplexity on the learned model .
- Perplexity2 is used to evaluate the effectiveness of prediction of the model.

Research in Crowdsourcing

- Crowdsourced Science
  – Traditional Science that enhanced by crowdsourcing
- Science of Crowdsourcing
  – The characteristics of Human Computation as new hardware

210

Crowdsourced Science

- Discover new tasks suitable for crowds
  - Information Retrieval
    - New methods for experiments
  - Machine Learning
    - New and cheap resource of labeled data
- Quality Control
  - How to determine the discovered new galaxy in GalaxyZoo
- Gamification
  - How to make it fun in Fold.it
    - The crowds fold a branch to help enumerate structure of a protein.

Science of Crowdsourcing

- The study of HPU as new hardware
  - What is the **clock-rate**?
  - What is the basic **operation** on HPU?
  - What is the **reliability** of HPU?
- Algorithms Design based on HPU
  - **Complexity** of human algorithms?
  - Is there **NP-hard** theory based on HPU?

212

Acknowledgement

- Dr. Caleb Chen Cao
- Dr. Yongxin Tong
- Mr. Jason Chen Zhang
- Prof. H. V. Jagadish
- Mr. Leihao Xia
- Mr. Zhao Chen
- Mr. Rui Fu
- Mr. Ziyuan Zhao

213