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

CONTRIBUTED ARTICLES
Searching for Jim Gray: A Technological Odyssey

- Locally computed features quickly at software only locations as to the ease
  of location, extracting data can be much faster than ever before
- The M.S. Grant was designed to
  - Be able to handle location and
  - Assist in finding
- New computer tools and
  - Could help with group
  - Computer vision
  - New forms of
  - Analysis of
  - Satellite
  - Imagery
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
Marketplaces

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

- **Marketplaces**
  - Provide crowds with tasks

- **Crowds**
  - Workers perform tasks

For definitions, analysis, free book chapters, and other crowdsourcing resources go to:

http://www.resultsfromcrowds.com/
Notable Marketplaces

• Mechanical Turk
• CrowdFlower
• CloudCrowd
• Clickworker
• SamaSource

Eg, oDesk: Micro- vs. Macro-task

Game Jobs

Flash Game Interface Design

Narrow results by:

CATEGORY
Software Development

Translation task
Eg, Amazon Mechanical Turk (AMT)

Make Money by working on HITs
HTTs – Human Intelligence Tasks - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:
- Can work from home
- Choose your own work hours
- Get paid for doing good work

Find an interesting task → Work → Earn money

Get Results from Mechanical Turk Workers
Ask workers to complete HITs – Human Intelligence Tasks - and get results using Mechanical Turk. Register Now.

As a Mechanical Turk Requester you:
- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you’re satisfied with the results

Find your account → Load your tasks → Get results

AMT HIT

- Tasks
  - Called HIT (Human Intelligence Task)
  - Micro-task

- Eg
  - Data cleaning
  - Tagging / labeling
  - Sentiment analysis
  - Categorization
  - Surveying
  - Photo moderation
  - Transcription

Translation task

Translated from English to Russian:

Translate a text between the markers below from English to Russian. Human translation only! Machine translations will be rejected.

Translation:

Any notes? Advice? Emotions? (Optional)

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**
- 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 sought for a HIT

$r = 1$

$r = 3$
Round (= Step)

- Algorithms are executed in rounds
- # of rounds \(\approx\) 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

Which is better?

\[
\begin{align*}
  & s_1 = 2 \\
  & r_1 = 3
\end{align*}
\]

Which is the best?

\[
\begin{align*}
  & s_2 = 3 \\
  & r_2 = 2
\end{align*}
\]

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]

![Tournament Graph]

- Varying $r_i$, $s_i$
- Varying $r_i$
- Constant $r_i$, $s_i$

Budget (B)

Pr[max item]

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

  Round 1
  Round 2
  Round 3

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

Round 1

Round 2

Round 3

Round 4

Round 5

Total, 6 questions
With 5 rounds
Top-K Operation

- This is a top-k list algorithm
- Analysis

<table>
<thead>
<tr>
<th></th>
<th>$k = 1$</th>
<th>$k \geq 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td># of questions</td>
<td>$O(n)$</td>
<td>$O(n + k \lceil \log_2 n \rceil)$</td>
</tr>
<tr>
<td># of rounds</td>
<td>$O(\lceil \log_2 n \rceil)$</td>
<td>$O(k \lceil \log_2 n \rceil)$</td>
</tr>
</tbody>
</table>

- 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]**

\[ N = 10, \ s = 4, \ k = 2 \]

\[ s-rank() \]

**Top-2 items**

\[ s-rank() \]

\[ s-rank() \]

\[ s-rank() \]

\[ s-rank() \]

**Top-K [Polychronopoulous-WebDB13]**

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

\[ s-rank() \]

\[ W1 \]

\[ 4 \]

\[ 1 \]

\[ 2 \]

\[ 3 \]

\[ W2 \]

\[ 4 \]

\[ 2 \]

\[ 1 \]

\[ 3 \]

\[ W3 \]

\[ 3 \]

\[ 2 \]

\[ 3 \]

\[ 4 \]

**Median Ranks**

\[ 4 \]

\[ 2 \]

\[ 2 \]

\[ 3 \]

\[ Top-2 \]

\[ ICDE 2015 Tutorial \]
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 \( \left\lfloor \frac{S}{2} \right\rfloor \) 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

A > B > C > D > E

\[ W1 \quad W2 \quad W3 \quad W4 \]

Error  Submit
Sort [Marcus-VLDB11]

N=5, S=3

A → B → E → D → C

Topological Sort

Sorted Result

A → E → D → B → C

ICDE 2015 Tutorial

#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)

<table>
<thead>
<tr>
<th>Worker</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>4</td>
</tr>
<tr>
<td>W2</td>
<td>3</td>
</tr>
<tr>
<td>W3</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Worker</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>1</td>
</tr>
<tr>
<td>W2</td>
<td>2</td>
</tr>
<tr>
<td>W3</td>
<td>1</td>
</tr>
</tbody>
</table>

Mean rating

1.3

3.6

8.2
Sort [Marcus-VLDB11]

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

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

```
  _______  
 /         
|         |  
|  1 2 3 4 5 6 7 |
|         |  
 
  _______  
 /         
|         |  
|  1 2 3 4 5 6 7 |
|         |  
 
```

S may not be accurately sorted

- How to select the size of S
  - Random
  - Confidence-based
  - Sliding window

• #3: Hybrid Sort
  - First, do rating-based sort $\rightarrow$ sorted list $L$
  - Second, do comparison-based sort on $S$ $(S \subseteq L)$
    • $S$ may not be accurately sorted
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

Finds that in general comparison-sort > rating-sort
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 $P$

• $\approx$ Filter, Find, Screen, Search
Select [Yan-MobiSys10]

- Improving mobile image search using crowdsourcing

ICDE 2015 Tutorial

Select [Yan-MobiSys10]

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

ICDE 2015 Tutorial
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

![Diagram showing early prediction in CrowdSearch]

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

![Diagram showing time line for prediction]
Select [Yan-MobiSys10]

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

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

---

Count [Marcus-VLDB13]

- Label Count (via sampling)

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

What is the gender of this person?
- male  
- female

What is the gender of this person?
- male  
- female
Count [Marcus-VLDB13]

• Batch Count

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

About how many of the 10 people are male? 4

About how many of the 10 people are female?

• 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**
  - \( \approx \) similarity join, entity resolution (ER), record linkage, de-duplication, …
  - Beyond the exact matching
- [Chaudhuri-ICDE06] similarity join
  - \( R \ JOIN_p S \), where \( p = \text{sim}(R.A, S.A) > t \)
  - \( \text{sim}(\cdot) \) 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

[Yes] [No]

#2 Naïve Batching Join

Batch factor \( b = 2 \)
Join [Marcus-VLDB11]

Find pairs of images with the same celebrity
- To select pairs, click on an image on the right.
- To magnify a picture, hover your pointer over it.
- To submit a selected pair, click on the green button.
- If none of the celebrities match, click on the red button.
- There may be multiple candidates per pair.

### #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, $10,000 \times 10,000 / (20 \times 20) = 250,000$ HITs \(\rightarrow\) 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)

### Main Issue: HIT Generation Problem

- Pair-based HIT Generation
  - Na""ive 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 \( \rightarrow \) approximation solution

---

Cluster-based HIT Generation

<table>
<thead>
<tr>
<th>ID</th>
<th>Product Name</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>iPad Two 16GB WIFI White</td>
<td>$490</td>
</tr>
<tr>
<td>r2</td>
<td>iPad 2nd generation 16GB WIFI White</td>
<td>$499</td>
</tr>
<tr>
<td>r3</td>
<td>iPhone 4th generation White 16GB</td>
<td>$495</td>
</tr>
<tr>
<td>r4</td>
<td>Apple iPhone 4 16GB White</td>
<td>$520</td>
</tr>
<tr>
<td>r5</td>
<td>Apple iPhone 3rd generation Black 16GB</td>
<td>$575</td>
</tr>
<tr>
<td>r6</td>
<td>iPhone 4S 16GB White</td>
<td>$599</td>
</tr>
<tr>
<td>r7</td>
<td>Apple iPad2 16GB WIFI White</td>
<td>$499</td>
</tr>
<tr>
<td>r8</td>
<td>Apple iPod shuffle 2GB Blue</td>
<td>$49</td>
</tr>
<tr>
<td>r9</td>
<td>Apple iPod shuffle USB Cable</td>
<td>$10</td>
</tr>
</tbody>
</table>

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]
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]

---

A simple example – crowd data sourcing (Qurk)

<table>
<thead>
<tr>
<th>name</th>
<th>Picture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lucy</td>
<td><img src="Lucy.png" alt="Picture" /></td>
</tr>
<tr>
<td>Don</td>
<td><img src="Don.png" alt="Picture" /></td>
</tr>
<tr>
<td>Ken</td>
<td><img src="Ken.png" alt="Picture" /></td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

**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”
A simple example – crowd data sourcing

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

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Treated a sore throat with garlic and oregano leaves...</td>
</tr>
<tr>
<td>Treated a sore throat and low fever with garlic and ginger...</td>
</tr>
<tr>
<td>Treated a heartburn with water, baking soda and lemon...</td>
</tr>
<tr>
<td>Treated nausea with ginger, the patient experienced sleepiness...</td>
</tr>
</tbody>
</table>

- 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:

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

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

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.

$$\text{supp}_u(\text{headache} \rightarrow \text{coffee}) = \frac{1}{7} \cdot \frac{9}{10} \quad \text{conf}_u(\text{headache} \rightarrow \text{coffee}) = \frac{9}{10}$$
**Significant rules**

- **Significant rules**: Rules were the mean user support and confidence are above some specified thresholds $\Theta_{sr}, \Theta_{sc}$.

- **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 \( f_r \):
  - \( \mu \) – the sample average
  - \( \Sigma \) – 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 \( r \)

\[
M_r = \int_{\theta_x^-}^{\theta_x^+} \int_{\theta_c^-}^{\theta_c^+} f_r(s, c) \, dc \, ds
\]

- \( r \) 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

\[
frequent(\{\text{headache, espresso}\}) \Rightarrow frequent(\{\text{headache, coffee}\})
\]

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

<table>
<thead>
<tr>
<th>Complexity boundaries</th>
</tr>
</thead>
</table>

• 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)

<table>
<thead>
<tr>
<th></th>
<th>W.r.t. the Input</th>
<th>W.r.t. the Output</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Crowd</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower</td>
<td>$\Theta(\log</td>
<td>S(\Psi)</td>
</tr>
<tr>
<td>Upper</td>
<td>$\Omega(</td>
<td>\log</td>
</tr>
<tr>
<td><strong>Comp.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower</td>
<td>$\Omega(</td>
<td>\log</td>
</tr>
<tr>
<td>Upper</td>
<td>$\Omega\left(</td>
<td>I(\Psi)</td>
</tr>
</tbody>
</table>

$|I(\Psi)| = 2^{O(|\Psi|)}$, $|S(\Psi)| = 2^{O(|\Psi|)}$
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 of a 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 x.
 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 x.
 MORE} 
WITH SUPPORT = 0.03
```

$\text{x} = \text{Central\_Park},$
$\text{y} = \text{Basketball}$

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

“Every week.” (support = $1/2$)
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

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
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

Original query

new query 1 → Answer to query 1

new query 2 → Answer to query 2

... → ...

Answer to original query

What is the most efficient way to dismantle?

Can we do it all fully automated?

How can we “glue” the answers together?

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

```
Bmi
  / \                      / \
Height  Weight           Tall  Age
       / \                  / \                 / \
Gender  Fat   Attractiveness  Safe  Loud  View  Near Bus
       / \                  / \               / \     
Good Neighborhood  Size  Good Location  #Rooms
```

Crowd Mining
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 $l$)

$Q = \text{Select name, BMI from pictures}$

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

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 $B_{pre}$

**Output**
- Find $b, l$
  - $b:A \rightarrow$
  - $l:A \rightarrow$
- That minimize
  - $E_x = \min_{b,l} \left[ \sum_{a \in A} l(a) b(a) - Q(o) \right]$
- Subject to
  - $\sum_{a \in A} b(a) = B_{obj}$
  - $\sum_{\text{pre-processing tasks}} \text{Cost}(\text{task}) < B_{pre}$
### Solution components

<table>
<thead>
<tr>
<th>Choosing Dismantling Questions</th>
<th>Estimating Statistics</th>
<th>Calculating $b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Based on probability of new answer (attribute) and expected answer’s correlation</td>
<td>• Inductive solution</td>
<td>• [Sabato, Kalai ICML’13]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Adaptations to match our scenario</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Calculating $l$</th>
<th>Deciding When to Stop</th>
</tr>
</thead>
<tbody>
<tr>
<td>• A well studied problem</td>
<td>• Minimal learning budget as a function of the number of attributes</td>
</tr>
<tr>
<td>• Collecting dataset based on calculated heuristics</td>
<td></td>
</tr>
</tbody>
</table>

---

### 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
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?

<table>
<thead>
<tr>
<th></th>
<th>Social Media Network</th>
<th>General Purpose Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility</td>
<td>Highly convenient, on all kinds of mobile devices</td>
<td>Specific online platform</td>
</tr>
<tr>
<td>Incentive</td>
<td>Altruistic or payment</td>
<td>Mostly monetary incentive</td>
</tr>
<tr>
<td>Supported tasks</td>
<td>Simple tasks as decision making</td>
<td>Various types of tasks</td>
</tr>
<tr>
<td>Communication Infrastructure</td>
<td>‘Tweet’ and ‘Reply’ are enough</td>
<td>Complex workflow control mechanism</td>
</tr>
<tr>
<td>Worker Selection</td>
<td>Active, Enabled by ‘@’</td>
<td>Passively, No exact selection</td>
</tr>
</tbody>
</table>

Whom to Ask?

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

Andy Lau  Cecilia Cheung  Nicholas Tse  Jackie Chan

“HK Coliseum”  “?”  “Hong Kong Cultural Centre”  “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?"

- $\epsilon$: 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)
  - \( \text{JER}(A,B,C) = 0.1 \times 0.2 \times 0.2 + (1 - 0.1) \times 0.2 \times 0.2 + 0.1 \times (1 - 0.2) \times 0.2 + 0.1 \times 0.2 \times (1 - 0.2) = 0.072 \)
  - Better than A(0.1), B(0.2) or C(0.2) individually
What if we enroll more
- \( JER(A,B,C,D,E) = 0.0704 < JER(A,B,C) \)
- The more the better?

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

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 $J_n = \{j_1, j_2, \ldots, j_n\} \subseteq S$ is a set of jurors with size $n$ that can form a voting.

  ![Jury Example](image)

  A Jury $J_n = \{j_1, j_2, j_3\}$ with 3 jurors

  **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.

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

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

  ![Voting Scheme Example](image)

  A Voting $V_n = \{1,0,1\}$ from $J_n$
Problem Definition

- **Individual Error-rate**

  **Definition 4** (Individual Error Rate - \( \epsilon_i \)). The individual error rate \( \epsilon_i \) is the probability that a juror conducts a wrong voting. Specifically:

  \[
  \epsilon_i = \Pr(\text{vote otherwise}|\text{a task with ground truth } A)
  \]

  \[
  \begin{array}{c}
  \epsilon_i (0.1) \\
  \epsilon_i (0.3) \\
  \epsilon_i (0.2) \\
  r_i (0.3) \\
  r_i (0.4) \\
  r_i (0.2)
  \end{array}
  \]

  A Voting \( V_n = \{1,0,1\} \) from \( f_n \)

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

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:

\[
\text{JER}(J_n) = \sum_{k=2}^{n} \sum_{S \subseteq F_k} \prod_{j \in S} \epsilon_j \prod_{j \notin S} (1 - \epsilon_j)
\]

\[
= \Pr(C \geq \frac{n+1}{2} | J_n)
\]

where \( F_k \) is all the subsets of \( S \) with size \( k \) and \( \epsilon_j \) is the individual error rate of juror \( j_i \).

\[
\text{JER}(J_2) = 0.1^*0.3^*0.2 + (1-0.1)^*0.3^*0.2 + 0.1^*(1-0.3)^*0.2 + 0.1^*0.3^*(1-0.2)
\]

\[
\text{JER}(J_3) = 0.1^*0.3^*0.2 + (1-0.1)^*0.3^*0.2 + 0.1^*(1-0.3)^*0.2 + 0.1^*0.3^*(1-0.2)
\]

\[
= 0.029
\]
Problem Definition

- Crowdsourcing Models (model of candidate microblog users)

  **Definition 7** (Altruism Jurors Model - AMT). 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 $\sum_{i \in J} r_i \leq B$.

Problem Definition

- Juror 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 (AMT 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
  – Otiode 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
Based on Majority Voting, to achieve an overall confidence of $\theta$, how should we build a most economical market?

- “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 ?
How about \{B, C, D\}?

The expected cost is lower (2.36), but the market confidence disagrees with the threshold \(\theta_2 = 0.82\).
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 $i$, the Investor Confidence $c_i$ is the probability that $i$ chooses the same option as the ground truth. Respectively, given a ground truth $G$, the confidence

$$c_i = Pr\{i_i\text{chooses correctly}\}$$

$$= Pr\{G = 0\} \cdot Pr\{v_i = 0|G = 0\} + Pr\{G = 1\} \cdot Pr\{v_i = 1|G = 1\}$$

$$= Pr\{v_i = G|G\}$$

- $v_i$ 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 \( WM_n = \{i_1, i_2, \ldots, i_n\} \subseteq I \) with size \( n \), where each \( i_t \) is associated with an individual confidence \( c_t \) and actual voting \( v_t \).

- **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:

  \[
  OP(WM_n) = \begin{cases} 
  1 & \text{if } \sum v_i \geq \left\lceil \frac{n}{2} \right\rceil \\
  0 & \text{if } \sum v_i \leq \left\lceil \frac{n}{2} \right\rceil 
  \end{cases}
  \]

  - 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 \( G \):

  \[
  MC(WM_n) = \Pr(OP(WM_n) = G | G) = \Pr(\lvert C \rvert \geq \left\lceil \frac{n}{2} \right\rceil) = \Pr(\lvert C \rvert \geq \frac{n+1}{2})
  \]

  \[
  = \sum_{k=\left\lceil \frac{n}{2} \right\rceil}^{n} \prod_{A \in F_k} c_i \prod_{j \in A^c} (1 - c_j)
  \]

  - \( F_k = \{A | \lvert A \rvert = k, A \subseteq WM_n \} \) 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:
  
  $$Cost(WM_n) = |W| = \left| \{ t_i | t_i \in WM_n \text{ s.t. } v_i = OP(WM_n) \} \right|$$

- **Expected Market Cost**
  
  $$E[Cost(WM_n)] = \sum_{k=1}^{n} k \cdot \Pr(|W| = k)$$
  
  $$= \sum_{k=1}^{n} k \left( \sum_{A \subseteq F_k} \prod_{a \in A} (1 - c_a) \prod_{j \in A'} c_j \right) + \sum_{A \subseteq F_k} \prod_{a \in A} (1 - c_a) \prod_{j \in A'} c_j$$

- 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 $I = \{ t_1, \ldots, t_N \}$ with size $N$, a Market Confidence threshold $\theta$, the Effective Market Problem (EMP) is to find a subset of all investors $WM_n \subseteq I$, so that:
  
  minimize $E[Cost(WM_n)]$
  
  subject to $MC(WM_n) \geq \theta$

  A market **BUILDER** for tasks holders
Experimental Studies

• Calculating MC and Cost - Effectiveness

  ![Graphs showing effectiveness of calculating MC and Cost](image)

  • 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
  – 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
    • $d = \{d_1, d_2, ..., d_m\}$
  – Reported by the worker
    • $r = \{r_1, r_2, ..., r_m\}$
  – 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 – 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

\[ M_i = C_i \cdot \frac{\text{Odd}}{D_i + 1} = C_i \cdot \frac{\text{Odd}}{D(\hat{r}_i||\bar{p}) + 1} \]  

\( C_i \) is the invested capital of \( T_i \), and Odd is the preset parameter such that at most a trader could earn \( \text{Odd} \times C_i \) 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

\[ 0 < M_i \leq \text{Odd} \times C_i \]  

The maximum equality is observed when \( \hat{r}_i = \bar{p} \).

• Goodness of a report
  – Expected logarithm form
COPE – The Design

• Pre-market Building
  – Generate Seed Capital
    • Promised Salaries as initial funds
  – Tendency Evaluation
    • Optimistic
    • Pessimistic
    • Group Mean
      For bins adjustment during
      $\mu_n = \frac{\sum \hat{\mu}_n^2}{|n|}$ an update
      $\hat{\mu}_p = \frac{\sum \hat{\mu}_p^2}{|p|}$

COPE – The Design

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

\[ p^* = \frac{Pr(p|\bar{r}) \times \frac{Pr(p)L(r|\bar{p})}{Pr(\bar{r})}}{Pr(\bar{r})} \]

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

- task: estimate man’s age according to a photo
- \( dir \) means Direct Pay, \( mkt \) means market-based

• 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.
Crowd-Oriented Service Data

• A snippet of crowd-oriented service from Stack Overflow

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Tasks</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Android application database to save images ...</td>
<td>2014-01-31 17:30:33</td>
</tr>
<tr>
<td>T2</td>
<td>How to use simpleadapter with activity Android...</td>
<td>2014-02-02 10:31:25</td>
</tr>
<tr>
<td>T3</td>
<td>How to find mobile phones on hotspot networks in iPhone?</td>
<td>2014-02-03 21:40:01</td>
</tr>
</tbody>
</table>
Crowd-Oriented Service Data

- A snippet of crowd-oriented service from Stack Overflow

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Tasks</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Android application database to save images ...</td>
<td>2014-01-31 17:30:33</td>
</tr>
<tr>
<td>T2</td>
<td>How to use simpleadapter with activity Android...</td>
<td>2014-02-02 10:31:25</td>
</tr>
<tr>
<td>T3</td>
<td>How to find mobile phones on hotspot networks in iPhone?</td>
<td>2014-02-03 21:40:01</td>
</tr>
</tbody>
</table>
## Crowd-Oriented Service Data

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

<table>
<thead>
<tr>
<th>Response ID</th>
<th>Task ID</th>
<th>Responses</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>T1</td>
<td>Android SQLite database with multiple...</td>
<td>2014-01-31 18:23:01</td>
</tr>
<tr>
<td>R2</td>
<td>T1</td>
<td>Save the image to your sdcard. ...</td>
<td>2014-02-01 15:01:53</td>
</tr>
<tr>
<td>R3</td>
<td>T1</td>
<td>Storing images in your database will...</td>
<td>2014-02-01 16:38:17</td>
</tr>
</tbody>
</table>
Crowd-Oriented Service Data

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

<table>
<thead>
<tr>
<th>Response ID</th>
<th>Task ID</th>
<th>Responses</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>T1</td>
<td>Android SQLite database with multiple...</td>
<td>2014-01-31 18:23:01</td>
</tr>
<tr>
<td>R2</td>
<td>T1</td>
<td>Save the image to your sdcard. ...</td>
<td>2014-02-01 15:01:53</td>
</tr>
<tr>
<td>R3</td>
<td>T1</td>
<td>Storing images in your database will...</td>
<td>2014-02-01 16:38:17</td>
</tr>
</tbody>
</table>
Characteristic of Crowd-Oriented Service Data-I

- Task-Response Pairs
  - Task-Response Correlation

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

<table>
<thead>
<tr>
<th>Response ID</th>
<th>Task ID</th>
<th>Responses</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>T1</td>
<td>Android SQLite database with multiple...</td>
<td>2014-01-31 18:23:01</td>
</tr>
<tr>
<td>R2</td>
<td>T1</td>
<td>Save the image to your sdcard. ...</td>
<td>2014-02-01 15:01:53</td>
</tr>
<tr>
<td>R3</td>
<td>T1</td>
<td>Storing images in your database will...</td>
<td>2014-02-01 16:38:17</td>
</tr>
</tbody>
</table>

Characteristic of Crowd-Oriented Service Data-II

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

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Tasks</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Android application database to save images</td>
<td>2014-01-31 17:30:33</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response ID</th>
<th>Task ID</th>
<th>Responses</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>T1</td>
<td>Android SQLite database with multiple...</td>
<td>2014-01-31 18:23:01</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>R100</td>
<td>T1</td>
<td>Storing images in your database will...</td>
<td>2014-02-04 11:36:02</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### Characteristic of Crowd-Oriented Service Data-III

#### Dynamic Evolution with Time

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Tasks</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>Android application database to save images ...</td>
<td>2014-01-31 17:30:33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response ID</th>
<th>Task ID</th>
<th>Responses</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>T1</td>
<td>Android SQLite database with multiple...</td>
<td>2014-01-31 17:30:33</td>
</tr>
<tr>
<td>R2</td>
<td>T1</td>
<td>Save the image to your sdcard. ...</td>
<td>2014-02-01 10:31:25</td>
</tr>
<tr>
<td>R3</td>
<td>T1</td>
<td>Storing images in your database will...</td>
<td>2014-02-01 21:40:01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Tasks</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>How to use simpleadapter with activity Android...</td>
<td>2014-02-02 10:31:25</td>
</tr>
</tbody>
</table>

#### Timeline

- **1st Bucket**
- **2nd Bucket**
- **3rd Bucket**
Challenges

• How to model big crowd-oriented service data
  
  **Task-Response Correlation**

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

• High training efficiency is important for the big data

• Topics over big crowd-oriented service data are evolving

Problem Definitions

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

<table>
<thead>
<tr>
<th>Response ID</th>
<th>Task ID</th>
<th>Responses</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>R_{1,1}</td>
<td>T_{1,0}</td>
<td>Android SQLite database with multiple...</td>
<td>2014-01-31 18:23:01</td>
</tr>
<tr>
<td>R_{1,2}</td>
<td>T_{1}</td>
<td>Save the image to your sdcard. ...</td>
<td>2014-02-01 15:01:53</td>
</tr>
<tr>
<td>R_{1,3}</td>
<td>T_{1}</td>
<td>Storing images in your database will…</td>
<td>2014-02-01 16:38:17</td>
</tr>
<tr>
<td>R_{3,1}</td>
<td>T_{3}</td>
<td>iOS 7 system of Apple devices provide...</td>
<td>2014-02-03 22:14:27</td>
</tr>
</tbody>
</table>

• (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 $\phi$ 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 $\phi$ 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.

- If a document $d$ is a task, sample a response $d'$ with regard to the number of sketch pairs between $d$ and $d'$;

- Otherwise, $d$ is a response and select the 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.
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.

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(\) \text{ Refer to our paper for the proof if you are interested.}$$
Pairwise Sketch

Sketch for counting frequency of words in tasks

Task word frequency

Response word frequency

Sketch for counting frequency of words in responses

Hash Function

Word in Task

iPhone

Word in Response

iOS
Belief Residual of Sentences

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

- The belief residual $r_{d,s}^k$ between two successive iterations $t$ and $(t-1)$ is calculated as follows:

$$r_{d,s}^k = |\mu_{d,s}^k(t) - \mu_{d,s}^k(t-1)|$$

- The estimation of the topic distribution in the document level.

- The estimation of the word distribution in the topic level.
Belief Update Algorithm

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

---

Belief Update Algorithm

- A Running Example

```
Selected topics

Selected documents

Topic 1  Topic 2  Topic 3  Topic 4

...  ...  

Topic 1  Topic 2  Topic 3  Topic 4

...  ...  

Document ignored in current iteration
```
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
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 \textit{clock-rate}\textsuperscript{a}?\footnote{The clock-rate is the frequency at which the hardware operates.}
  - What is the basic \textit{operation}\textsuperscript{b} on HPU?\footnote{The operation is the basic function that the hardware performs.}
  - What is the \textit{reliability}\textsuperscript{c} of HPU?\footnote{The reliability is the measure of how consistently the hardware operates.}
- Algorithms Design based on HPU
  - \textit{Complexity}\textsuperscript{d} of human algorithms?\footnote{The complexity of an algorithm measures how difficult it is to understand and implement.}
  - Is there \textit{NP-hard}\textsuperscript{e} theory based on HPU?\footnote{NP-hard problems are those that are computationally intensive and difficult to solve.}
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