Title: On Optimality of Jury Selection Problem in Crowdsourcing

Yudian, Reynold, Silviu, Luyi

EDBT 2015
Outline

- Introduction (Crowdsourcing)
- Problem Definition (Jury Selection Problem)
- Our Solution (Optimality)
- Conclusion
Why do we need crowd?

- **Problems**
  - Which picture visualizes better "Golden Gate Bridge"?
    - ![Golden Gate Bridge pictures]
    - Submit

- **Possible Solutions**
  - ![Bill Gates CEO of Microsoft?](yes-no.png)

---

Crowdsourcing Definition

Definition

Coordinating a **crowd** to do **micro-tasks** that solve **problems**.

Example

- **problems:** entity resolution
- **An example micro-task:**

  Are they the same?
  
  iPad 2 = iPad Two
  
  [ ] YES  [ ] NO

  [SUBMIT]
Amazon Mechanical Turk

- **Requesters**
  - Get Results from Mechanical Turk Workers
  - 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
  - Fund your account, Load your tasks, Get results

- **Micro-Tasks**
  - Are they the same? iPad 2 = iPad Two
  - YES ☐ NO ☐

- **Workers**
  - Is Bill Gates now the CEO of Microsoft?
  - YES ☐ NO ☐

- **Official Amazon Mechanical Blog (August, 2012)**

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

  http://mechanicalturk.typepad.com/blog/2012/08/mechanical-turk-featured-on-aws-report.html
Outline

- Introduction (Crowdsourcing)
- Problem Definition (Jury Selection Problem)
- Our Solution (Optimality)
- Conclusion
Problem Intuition (Worker Selection) - VLDB 12

Given (1) a Task
(2) a fixed Budget B
(3) a set of workers

Worker Selection Problem:
Choose a subset of workers, such that the task can be completed successfully (i.e., with high quality), in the most economical manner?

Next: Task and Worker

Task: Decision Making Task

- Answers are “yes” and “no”
- One (unknown) ground truth

**Decision Making Task**

<table>
<thead>
<tr>
<th>Is Bill Gates now the CEO of Microsoft?</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES 🟢</td>
</tr>
</tbody>
</table>

- Simplicity
- (Extensions) Multiple Choice Tasks

Worker - (quality, cost)

- Each Worker: (quality, cost)  
  Ex: A (0.77, $9)

- Jury: a subset of workers (Ex: \{A,B,D\})


Jury Selection Problem

**Task:**

- **Budget:** $20
- **Workers:**

For each Jury:

1. **Jury Cost:** $5 + $7 + $6 = $18
2. **Jury Quality:** $JQ$ (0.7, 0.65, 0.2),
   
   $Pr$ (correctly deriving a result based on workers’ answers)

*Select a Jury (subset of workers) such that the Jury Quality is maximized in all Jury whose cost does not exceed the Budget.*
Jury Quality Computation (MV) – VLDB12

- **Jury Quality for Majority Voting Strategy**

  - **MV** : return the answer which receives the highest votes

  - **Cost**({$5, $7, $6}) = 18 ≤ 20

  - **JQ**({0.7, 0.65, 0.2}, MV) = 54.3%

  

  \[
  \text{JQ}({0.7, 0.65, 0.2}, \text{MV}) = 0.7 \times 0.65 \times 0.8 + 0.7 \times 0.35 \times 0.2 + 0.3 \times 0.65 \times 0.2 + 0.7 \times 0.65 \times 0.2 = 54.3\%
  \]
Enumerating all Jury set satisfying budget constraint

**optimal jury set**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0.77, $9)</td>
<td>(0.7, $5)</td>
<td>(0.6, $2)</td>
</tr>
</tbody>
</table>

- Cost({$9, $5, $2}) = 16 ≤ 20
- JQ({0.77, 0.7, 0.6}, MV) = 77.42%

**Question: Is it optimal?**

Is it possible to provide a better solution for JSP, by replacing MV with another strategy?
Outline

- Introduction (Crowdsourcing)
- Problem Definition (Jury Selection Problem)
- Our Solution (Optimality)
- Conclusion
Classification of Voting Strategies

Based on whether the result is returned with degree of randomness, we can classify the voting strategies into two categories:

deterministic voting strategy (left part in the graph) and
randomized voting strategy (right part in the graph).

Example:
{0,1,1} 0.7,0.6,0.2

Majority Voting (Deterministic):
return 1

Randomized Majority Voting (Randomized):
return 0 with probability 1/3
return 1 with probability 2/3
Existence of Optimal Voting Strategy

Given a Jury set \( J \) and a strategy \( S \), the corresponding Jury Quality \( JQ(J,S) \) can be computed. An important question is:

Does there exists an optimal strategy \( S^* \), such that given a Jury set \( J \), the JQ for this strategy is not lower than the JQ for any strategy (including all deterministic and randomized strategies)?

\[
JQ(J, S^*) \geq JQ(J, S) \text{ for any } S
\]

We formally prove that the \textit{Bayesian Voting Strategy (BV)} is the optimal strategy, i.e., \( S^* = BV \).
*Proof of Optimality*

To answer this question, let us reconsider Definition 3. Let $h(V) = \mathbb{E}[\mathbb{1}_{S(V) = 0}]$. We have (i) $h(V) \in [0, 1]$; and (ii) $\mathbb{E}[\mathbb{1}_{S(V) = 1}] = 1 - h(V)$. Also, let $P_0(V) = \Pr(V = V, t = 0)$, and $P_1(V) = \Pr(V = V, t = 1)$. Hence, $JQ(J, S, \alpha)$ can be rewritten as

$$\sum_{V \in \Omega} \left[ P_0(V) \cdot h(V) + P_1(V) \cdot (1 - h(V)) \right]$$

$$= \sum_{V \in \Omega} \left[ h(V) \cdot (P_0(V) - P_1(V)) + P_1(V) \right]$$

This gives us a hint to maximize $JQ(J, S, \alpha)$ and find the optimal voting strategy $S^*$. Let $h^*(V) = \mathbb{E}[\mathbb{1}_{S^*(V) = 0}]$. It is observed that $P_1(V)$ is constant for a given $V$ and $h(V) \in [0, 1]$ for all $S$’s (no matter it is a deterministic one or a randomized one). Thus, to optimize $JQ(J, S, \alpha)$, it is required that

1. if $P_0(V) - P_1(V) < 0$, $h^*(V) = 0$, and so, $S^*(V) = 1$;
2. if $P_0(V) - P_1(V) \geq 0$, $h^*(V) = 1$, and so, $S^*(V) = 0$. 
Bayesian Voting Strategy

Example:
{0,1,1} 0.7,0.6,0.2

Majority Voting Strategy:
give 1 vote for the supported answer

0: 1 (by worker 1)
1: 1 (by worker 2) + 1 (by worker 3) = 2

Bayesian Voting Strategy (Deterministic Strategy):
give \( \log\frac{p}{1-p} \) vote for the supported answer

0: \( \log\frac{0.7}{0.3} = 0.8473 \)
1: \( \log\frac{0.6}{0.4} + \log\frac{0.2}{0.8} = -0.981 \)

JQ({0.77,0.7,0.6}, MV) = 77.42%
JSP solution: {A, B, E}

JQ({0.77,0.6,0.25,0.2}, BV) = 86.95%
JSP solution: {A, E, F, G}
1. Given Jury J, JQ computation for BV, or JQ(J,BV)

Recall that the JQ computation requires enumerating exponential number (w.r.t $|J|$) of states, i.e.,

$$|\{0,1\}|^*|\{0,1\}|^{|J|} = 2^{|J|+1}$$

2. The number of Jury set satisfying Budget Constraint is

Exponential w.r.t. $N$, in the worst case $2^N$
Complexity 1 of JSP

1. Given Jury $J$, JQ computation for BV, or $\text{JQ}(J,BV)$

Recall that the JQ computation requires enumerating exponential number (w.r.t $|J|$) of states, i.e.,

\[ |\{0,1\}| \times |\{0,1\}| \times |J| = 2^{|J|+1} \]

- **NP-hardness of JQ computation**
- **Polynomial Approximation Algorithm** (with Pruning Technique)
- **Bounded by 1% Error**
*Q1: Computing JQ for BV is NP-hard

In order to prove the NP-hardness of computing JQ for BV, we can reduce the partition problem, a well-known NP-Complete Problem (also a decision problem) to the problem of computing JQ for BV.

**Partition Problem (NP-Complete Problem)**

Input: \( W = \{ w_1, w_2, \ldots, w_n \} \), \( w_i \) is integer (\( 1 \leq i \leq n \))

Output: yes/no

Decide whether \( W \) can be partitioned into two disjoint multi-sets \( W_1 \) and \( W_2 \), such that the sum of elements in \( W_1 \) is equal to the sum of elements in \( W_2 \).

**Reduction**

Input: \( W = \{ w_1, w_2, \ldots, w_n \} \), \( w_i \) is integer (\( 1 \leq i \leq n \))

Construct \( J = \{ j_1, j_2, \ldots, j_n \} \) and \( J' = \{ j_1, j_2, \ldots, j_{n+1} \} \) based on \( W \), then

1. If \( JQ(J', BV) > JQ(J, BV) \), then the output for partition problem of \( W \) is "yes";
2. If \( JQ(J', BV) \leq JQ(J, BV) \), then the output for partition problem of \( W \) is "no";

Since computing JQ for BV is not in NP (it is not a decision problem), then it is a NP-hard problem.
**Q1: Bucket-Based Approx. Alg. (Pruning)**

**Settings:**

\[
\sigma(q_1) = \sigma(q_2) = 1.2 \quad \sigma(q_i) = \log \frac{q_i}{1-q_i}
\]

**Compute JQ(J,BV):**

\[
\begin{align*}
\sigma(q_i) & \quad \rightarrow \quad (0,1) \\
\neg \sigma(q_i) & \quad \rightarrow \quad (-1.2,1-q_i) \\
\end{align*}
\]

\[
\begin{align*}
\sigma(q_2) & \quad \rightarrow \quad (1.2,q_1) \\
\neg \sigma(q_2) & \quad \rightarrow \quad (-1.2,1-q_1) \\
\end{align*}
\]

\[
\begin{align*}
\neg \sigma(q_2) & \quad \rightarrow \quad (-2.4,(1-q_1)(1-q_2)) \\
\end{align*}
\]

**Real Computed JQ(J,BV):**

\[
q_1q_2 + \left[q_1(1-q_2) + (1-q_1)q_2 \right]/2
\]

**Approximations**

\[
\log \frac{0.99}{1-0.99} < 4.6
\]

Aggregated bucket number

Represent it as a bucket number

\[
\begin{align*}
A & = \log \frac{q_1}{1-q_1} \\
B & = \log \frac{q_2}{1-q_2}
\end{align*}
\]

**numBuckets**
**Q1: Approximation Error Bound**

Notations:
Let \( \hat{JQ}(J, BV) \) denote the estimated JQ of the approximation algorithm, and \( JQ(J, BV) \) denote the real JQ.

We can prove:

1. \( \hat{JQ}(J, BV) \leq JQ(J, BV) \)
2. \( JQ(J, BV) - \hat{JQ}(J, BV) < e^{\frac{5}{4d}} - 1 \)

The time complexity of approximation algorithm is \( O(dn^3) \) and if \( d \geq 200 \), the approximation error is bounded within 1%.

\[ d = \frac{\text{numBuckets}}{n} \]

The polynomial algorithm will give within 1% approximation error bound.

**Real:** 80%
**Estimated:** 79-80%
Complexity 2 of JSP

2. The number of Jury set satisfying Budget Constraint is

Exponential w.r.t. N, in the worst case $2^N$

- NP-hardness of JSP
- Simulated Annealing Heuristic for general JSP
Combinatorial Optimization Problem

- Similar to Knapsack Problem, with the difference in the Objective Function

*NP-hard, intuitively as computing the JQ (Objective Function) is NP-hard

*Even though regarding it as an oracle, deriving the optimal solution is also NP-hard

=> N-th order knapsack problem
Simulated Annealing Heuristic

- Heuristic solving combinatorial optimization problem
- Avoid local minimum, probability of accepting a worse place

Minimize the cost function: \( c(x) \)

Starting point: \( x_0 \)

Global minimum

Local minimum
*Simulated: Different Voting Strategies

MV: Majority Voting
BV: Bayesian Voting
RB: Random Ballot Voting (Randomly returns 0 or 1)
RMV: Randomized Majority Voting

Randomly generate 10 workers with quality \( \mathcal{N}(\mu, 0.1^2) \)

(a) Varying \( \mu \)
(b) Varying \( n (\mu = 0.3) \)
(c) Varying \( n (\mu = 0.7) \)
*Simulated : Proposed Approx. Algorithm*

Observe the effect of our proposed approximation algorithms

(a) effect with the change of mean and variance  
(b) vary the bucket number  
(c) approximation error bound  
(d) pruning techniques
Real: End-to-End System Comparison

Collect Data from AMT:
600 questions, each question answered by 20 workers

Known Ground Truth -> workers’ qualities

Outline

- Introduction (Crowdsourcing)
- Problem Definition (Jury Selection Problem)
- Our Solution (Optimality)
- Conclusion
System (Optimal Jury Selection System)

**Decision Making Task**

<table>
<thead>
<tr>
<th>Is Bill Gates now the CEO of Microsoft?</th>
</tr>
</thead>
<tbody>
<tr>
<td>YES</td>
</tr>
</tbody>
</table>

**Budget-Quality Table**

<table>
<thead>
<tr>
<th>Budget</th>
<th>Optimal Jury Set</th>
<th>Quality</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>{ E, F }</td>
<td>75%</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>{ F, G }</td>
<td>80%</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>{ B, F, G }</td>
<td>84.5%</td>
<td>14</td>
</tr>
<tr>
<td>20</td>
<td>{ A, E, F, G }</td>
<td>86.95%</td>
<td>20</td>
</tr>
</tbody>
</table>

**Trade-Off**

<table>
<thead>
<tr>
<th>Budget 14</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
</tr>
<tr>
<td>(0.7, $5)</td>
</tr>
</tbody>
</table>

**All candidate Jurors Set (quality, cost)**

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.77, $9)</td>
<td>(0.7, $5)</td>
<td>(0.65, $7)</td>
<td>(0.6, $5)</td>
<td>(0.6, $2)</td>
<td>(0.25, $3)</td>
<td>(0.2, $6)</td>
</tr>
</tbody>
</table>
Thank you!

Contact Info:
Yudian Zheng
DataBase Group
Computer Science Department
The University of Hong Kong