Information Management via CrowdSourcing

Hector Garcia-Molina

Stanford University
Crowdsourcing
Not to be confused with:

- Wisdom of the Crowd
- Cloud Computing :-)


Does figure show > 45 dots?

Question A
Does figure show > 45 dots?
Does figure show > 45 dots?

Report Results for Question A
Does figure show > 45 dots?

Question B
Does figure show > 45 dots?
Does figure show > 45 dots?

Report Results for Question B
Many Crowdsourcing Marketplaces!
Real World Examples

- Categorizing Images
- Search Relevance
- Data Gathering
- Image Matching Translation
Many Research Projects!
The Many Faces of Crowdsourcing
The Many Faces of Crowdsourcing

Human-Computer Interaction

Software Systems

Human Issues

Machine Learning

Information Management
Crowd Information Management

• Two Aspects:
  – Crowd as Information Source
  – Crowd as Data Processor
Fundamental Tradeoffs

- Latency
- Cost
- Uncertainty
Efficiency: Fundamental Tradeoffs

- How long can I wait?
- How much $$ can I spend?
- What is the desired quality?
- Uncertainty

Latency
Efficiency: Fundamental Tradeoffs

- How much $$ can I spend?
- How long can I wait?
- What is the desired quality?
- Which questions do I ask humans?
- Do I ask in sequence or in parallel?
- How much redundancy in questions?
- How do I combine the answers?
- When do I stop?
Example: CrowdScreen

Dataset of Items → Predicate → Filtered Dataset

Item X satisfies predicate?

Y Y N
Strategy

YESs

NOs
Strategy

YESs

NOs

continue
decide PASS
decide FAIL
Strategy

decision point

YESs

NOs
More Examples

YESs

NOs
More Examples
More Examples

YESs

NOs
More Examples

![Graph showing YESs vs. NOs]

- YESs
- NOs
Some Optimizations
What is “best” strategy?
What is “best” strategy?

$p(x,y) = \text{probability of miss-classification at } x,y$

$\text{end}(x,y) = \text{probability of terminating at } x,y$

Expected error:

$\sum p(x,y) \cdot \text{end}(x,y)$

Expected cost:

$\sum (x+y) \cdot \text{end}(x,y)$
One (of many) optimization problems:

Find strategy that minimizes expected cost (# questions), such that expected error is less than threshold (and number of questions never exceeds m).
Example of Results

Average performance on Varying $s$

Runtime on Varying $m$
Beyond Single Filter

• Probabilistic Strategies
• Multiple Filters
• Categorizer (output more than 2 types)
Beyond Filtering

• Finding Max
• Sorting
• Clustering
• Entity Resolution
• Adding terms to a taxonomy
• Building a Folksonomy
• ...
Beyond Simple Models

- Worker Error Models
- Task Design
- Tracking Worker Abilities
- Payments
- Response Time Issues
- ...

34
Crowd As Information Source

Declarative queries

DBMS like thing

Web
The Deco Data Model

- Schema designer
  - relations and other stuff
  - Conceptual schema
  - Actual schema
  - automatic (system)
  - RDBMS
- End user
  - relations

RDBMS
### Small Example

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>French</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>California</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.8</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
### Small Example

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>French</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>California</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.8</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

![User view](image)

### Anchor

<table>
<thead>
<tr>
<th>restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
</tr>
<tr>
<td>Bytes</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

### Dependent

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.8</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>5.0</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.6</td>
</tr>
<tr>
<td>Bytes</td>
<td>4.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Dependent

<table>
<thead>
<tr>
<th>restaurant</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>French</td>
</tr>
<tr>
<td>Bytes</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

### Dependent

<table>
<thead>
<tr>
<th>restaurant</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

38
Small Example

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>French</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>California</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.8</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- User view

- Anchor
- fetch rule
- Dependent
- fetch rule
- fetch rule
### Small Example

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>French</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>California</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.8</td>
<td>California</td>
</tr>
</tbody>
</table>

**User view**

![User view diagram]

**Anchor**

**fetch rule**

**Dependent**

**fetch rule**

**fetch rule**

**fetch rule**
### Small Example

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>French</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>California</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.8</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **User view**
- **Anchor**
- **Dependent**

- **resolution rule**
- **Chez Panisse**
- **Bytes**
- **Chez Panisse**
- **Bytes**
- **Chez Panisse**
- **Bytes**
- **Dependent**
- **Dependent**
# Small Example

## User View

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>French</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
<td>California</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.8</td>
<td>California</td>
</tr>
</tbody>
</table>

...  

## 1. Fetch

## 2. Resolve

## 3. Join

### Anchor

<table>
<thead>
<tr>
<th>restaurant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
</tr>
<tr>
<td>Bytes</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

### Dependent

<table>
<thead>
<tr>
<th>restaurant</th>
<th>rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>4.8</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>5.0</td>
</tr>
<tr>
<td>Chez Panisse</td>
<td>4.9</td>
</tr>
<tr>
<td>Bytes</td>
<td>3.6</td>
</tr>
<tr>
<td>Bytes</td>
<td>4.0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>restaurant</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chez Panisse</td>
<td>French</td>
</tr>
<tr>
<td>Bytes</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>restaurant</th>
<th>cuisine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes</td>
<td>California</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
SELECT n,l,c
FROM country
WHERE l = ‘Spanish’
ATLEAST 8
Deco Prototype V1.0
Experimental Setup

SELECT n,l,c FROM country WHERE l = 'Spanish' ATLEAST 8

- Experimental Goals:
  - Different fetch configurations
    - Basic: $\varnothing \Rightarrow n + n \Rightarrow l + n \Rightarrow c$
    - Reverse: $l \Rightarrow n + n \Rightarrow l + n \Rightarrow c$
    - Hybrid: $l \Rightarrow n,c + n \Rightarrow l,c$
  - Different filter locations
    - After vs. between joins
- Experimental Setup:
  - 5 cents/task on MTurk
  - Empty tables initially
  - Default: reverse + between
Experiment 1: Different Fetch Rules

# Non-NULL Result Tuples

Time [minutes]
Experiment 1: Different Fetch Rules

$12.00 and 2 hrs
$2.20 and 15 min
$1.30 and 11 min
$12.00 and 2 hrs
Conclusion

• Crowdsourcing is exciting area!
• Many challenges!