CROWD-ENABLED DATABASE

LIN, WING-HO (JOE) & LEE, LIK-HANG (PAUL)
WHAT IS CROWDSOURCING?
CROWDSOURCING IN EVERYDAY LIFE

Web Security

CROWDSOURCING IN EVERYDAY LIVE(2)

Medical knowledge exchange

**figure 1**

Share your findings with the medical community

“We developed Figure 1 so members of the healthcare community could share images, knowledge, and clinical insight with each other, while keeping patient privacy safe.”

Joshua Landy, MD, Co-Founder

Available on the iPhone

App Store
EXISTING CROWDSOURCING PLATFORM

Amazon Mechanical Turk (AMT)

Make Money by working on HITs

HITs - 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

or learn more about being a Worker

Get Results from Mechanical Turk Workers

Ask workers to complete HITs - Human Intelligence Tasks - and get results using Mechanical Turk. Get Started.

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
WHY CROWDSOURCING IN DATABASE?

Some queries cannot be answered by machines only, because
1. require human input
2. too computationally difficult
1. CROWDDB

ISSUES OF RELATIONAL DATABASES

Issue 1: Subjective comparison

Example:

```sql
SELECT image FROM picture
WHERE label="HKUST"
ORDER BY relevance;
```

ISSUES OF RELATIONAL DATABASES(2)

Issue 2: Missing information

Example:
SELECT URL FROM Department
WHERE name = "MATH";

ISSUES OF RELATIONAL DATABASES(3)

Issue 3: Entity resolution

Example:
SELECT salary FROM Staff
WHERE name="Tim Cook";

NEW OPERATION: CROWDORDER

Solution for issue 1

SELECT image FROM picture
WHERE label="HKUST"
ORDER BY CROWDORDER(image, “Which picture best represent HKUST?”);
NEW KEYWORD: CROWD

Solution for issue 2
CREATE TABLE Department (university STRING, name STRING, URL CROWD STRING);

SELECT URL FROM Department WHERE name = "EECS";

NEW OPERATION: CROWDEQUAL

Solution for issue 3

SELECT salary FROM staff
WHERE name CROWDEQUAL(“Tim Cook”);

CROWDDB DESIGN CONSIDERATIONS

1. Worker quality

1. Malicious behavior of workers

CROWDDB DESIGN CONSIDERATIONS (2)

3. Clarify of tasks
   EG: Enter the phone number of the receipt

4. Reputation of task requesters
   EG: Google VS lesser-known startups

2. QURK

Handle atypical database workload

Determine # of Turker / cost per HIT

Generate a set of tasks for humans to perform

An example of task: JOIN operator

Avoid unnecessary re-running steps

The Qurk System Overview

2. QURK

• Operators addressed in Qurk
  1. Filters
  2. Crowd-provided Data == Missing Value
  3. Table-valued Ranking Functions
  4. Table-valued Join Operator
2. QURK

1. Filters

```
SELECT * FROM images where isFlower(img)
```

**TASK** isFlower(Image img) RETURN Bool: TaskType: Question Text: “Does this image: `<img src='%s'>` contain a flower?”, URLify(img) Response: Choice(“YES”, “NO”)
2. QURK

II. Crowd-provided Data

SELECT companyName, findCEO(companyName).CEO, findCEO(companyName).Phone FROM companies

TASK findCEO(String companyName)

RETURNS (String CEO,String Phone):

TaskType: Question Text: “Find the CEO and the CEO’s phone number for the company %s”, companyName

Response: Form(('CEO',String), ('Phone',String))
2. QURK

III. Table-valued Ranking Functions (SORT)

TASK `rankProducts(String[] prodNames)`

RETURNS String[]: TaskType:

Rank Text: “Sort the following list of products by their user reviews on Amazon.com” List: `prodNames`
IV. Table-valued Join Operator (JOIN)

**TASK** `imgContains(Image[] survivors, Image[] missing)`

**RETURNS** `Bool: TaskType: JoinPredicate`

**Text:** ``Drag a picture of any `<b>Survivors</b>` in the left column to their matching picture in the `<b>Missing People</b>` column to the right.``

**Response:** `DragColumns("Survivors", survivors, "Missing People", missing)`

2. QURK

- General Ideas on Optimization from Qurk
  1. Runtime Pricing
  2. **Batch Predicates**
  3. **Operator Implementation**
  4. **Join Heuristics**
  5. Task Result Cache
  6. Model Training
3. DECLARATIVE APPROACH USING HUMANS, ALGORITHMS, DATABASE

Yelp(I): = rJpeg(I),  haCopyright(I),  aDark(I),  hFood(I)

r = DB table  ha = human/algorithm  a = algorithm  h = human

Find Images:
Not dark
Not copyrighted
Only food

Get Image from Database

Check copyright

Filter out the dark images

3. DECLARATIVE APPROACH USING HUMANS, ALGORITHMS, DATABASE

• Uncertainty in answers due to...
  – Human errors
  – Subjective views
  – Spam

3. DECLARATIVE APPROACH USING HUMANS, ALGORITHMS, DATABASE

• Majority Voting
  – Choose the answer if the majority of crowd selects the answer
• Probabilistic Approach (Similar to maximum likelihood)
  – Set the threshold and compute the statistical confidence
  – Return true if the value exceeds threshold, Otherwise, false
• Item-Response Theory (used in exam e.g. GRE)
  – Review the answer to see the trustworthiness of a worker
  – But it is less practical in crowdsourcing as there will be very few responses

COMPARISON AMONG 3 MOST CITED PAPERS

• QURK vs CrowdDB
  – CrowdDB: propose a high-level framework for querying crowd workers and outline considerations for various operators
  – Declarative Approach: propose a declarative language for querying crowd workers and outline considerations for an uncertainty model over the collected results.
  – QURK: focus on how to implement crowd-powered operators and design a cost-optimized (Time/Cost reduction) system for executing HIT-based queries with these operators
    • Another Research Paper on Optimization of operators, i.e. Filters, Join, Sorting
4. QURK’S FURTHER STUDY ON Sorts AND JoINS

• Practical aspects in Qurk
  – Filters + Join
  – Sort

QUARK FILTER: INAPPROPRIATE CONTENT

photos(id PRIMARY KEY, picture IMAGE)

Query = SELECT * FROM photos WHERE isSmiling(photos.picture);

User Definition function

MATCHING CELEBRITIES

SIMPLE JOIN $O(NM)$

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

Yes  No
NAÏVE BATCHING JOIN O(NM/B)

SMART JOIN $O(NM/B^2)$

4-10x reduction in cost

Utilize the transitivity property
The following is the intuition: if
\( C \) joins with \( X \)
\( C \) does not join with \( Z \)

And
\( D \) joins with \( X \)
do we need to compare \( D \) and \( Z \)?
Possibly Reducing the number of comparisons!
Is the cat above smiling?

Yes  No

Is the cat above smiling?

Yes  No
JOIN HEURISTIC

- Avoid the comparison between male & female
  - E.g. 5 male and 5 female → Original 10x10 = 100 comparisons
  - Separate the photo by Gender → 5x5 + 5x5 = 50 comparisons

gender
hair color
skin color

50-66% reduction in cost

Sorting relies on Human judgement

INTERFACE (NO BATCHING)

Comparison

more dangerous?

Rating

how dangerous?

Batching

ISSUES WITH BATCHING

• Downside
  – Not welcomed by Crowd worker
  – Human Fatigue

• Merit
  – Saved Budget
  – Save Time, i.e. achieve learning curve by doing similar time

**MEASURING QUALITY OF SORTING**

Kendall’s Tau rank correlation Range: [-1, 1]

- Completely Comparison-Based
  - $\tau = 1$ (completely accurate)
  - $O(\text{# items}^2)$
- Completely Rating-Based
  - $\tau \approx 0.8$ (accurate)
  - $O(\text{# items})$
HYBRID APPROACH

• First, collect the score of items based on rating scores
• Take the average rating score on each item
• Then, use comparisons between item, in 1 out of 3 below options:
  – Random: pick S items, compare
  – Confidence-based: pick most confusing “window”, compare that first, repeat
  – Sliding-window: for all windows, compare

5. MAIN IDEA OF CROWD TOP-K/GROUP-BY WITH CROWDSOURCING

- Group the photos of individual players
  - Group-By Queries
  - Use “Player-Name” as attribute

- Find their most recent photos
  - Max/Top-k Queries
  - Use the “Photo Taken Date” as attribute

Susan B. Davidson, Sanjeev Khanna, Tova Milo, and Sudeepa Roy. 2013. Using the crowd for top-k and group-by queries. In *Proceedings of the 16th International Conference on Database Theory (ICDT ’13)*
6. MAIN IDEA OF SKYLINE QUERIES WITH CROWDSOURCING

- Missing value leads to incomplete dataset that impacts the answer quality from computation of skyline sets
- Missing value can be outsourced to human workers but the response times and execution costs are the major issues
  - Crowdsourcing effort will focus on the missing values of High impact tuples that influence the answer quality strongly.
  - Missing values of Low impact tuples will be resolved by Value Prediction, namely **Imputation algorithm**.

REFERENCE


