Data-Driven Crowdsourcing

Tova Milo
Data Everywhere

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

I will focus today on human knowledge

Think of humanity and its collective mind expanding...
Background - Crowd (Data) sourcing

The engagement of crowds of Web users for data procurement
Can we trust the crowd?
We need to be careful ...

- What questions to ask?
  [SIGMOD13, VLDB13, ICDT14, SIGMOD14 VLDB15, SIGMOD15, VLDB16, SIGMOD18]

- How to define & determine correctness of answers?
  [WWW12, EDBT15, ICDE17]

- Who to ask? how many people? how to best use the resources?
  [VLDB13, ICDT13, ICDE13, SIGMOD15 PODS’15, ICDT16, VLDB16, WEDBD18]
Crowd Mining: Crowdsourcing in an open world

- Human knowledge forms an open world
- Goal: extract interesting and important patterns
  - Health care: Folk medicine, people’s habits, doctor’s intuition...
  - Finance: People’s habits & preferences, consultant’s intuition...
  - ...
- What questions to ask?
Back to classic databases...

- Significant data patterns are identified using **data mining**
- A useful type of pattern: **association rules**

*The classical supermarket example...*

- 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>Treated a sore throat with garlic and oregano leaves...</th>
</tr>
</thead>
<tbody>
<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, then 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 you have both nausea and fever, how often do you use a ginger and honey infusion?

Use the two types interleavingly.
More Examples

Ann, a vacationer, is interested in finding child-friendly activities at an attraction in NYC and a good restaurant nearby (plus relevant advice).

“You can play baseball in Central Park and eat at Maoz Vegetarian. **Tips:** Apply for a ballfield permit online”

“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”

A dietician may wish to study the culinary preferences in some population, focusing on food dishes that are rich in fiber.
Ann, a vacationer, is interested in finding child-friendly activities at an attraction in NYC and a good restaurant nearby (plus relevant advice).

“*You can play baseball in Central Park and eat at Maoz Vegetarian.*

**Tips:** Apply for a ballfield permit online

“*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

---

**General knowledge:**
- General truth, objective data, not associated with an individual
- *E.g.*, geographical locations
- Can be found in a knowledge base or an ontology

**Individual knowledge:**
- Related to the habits and opinions of an individual
- *E.g.*, travel recommendations
- We can ask people about it

When missing in the knowledge base, we can ask the crowd!

Crowd answers can be recoded in a knowledge base

---

*Data-Driven Crowdsourcing*
Given an ontology of general knowledge and a mining task

- Incrementally explore relevant patterns

\{Ball\_Game playAt Central\_Park\}

- Generate (closed and open) questions to the crowd about them

How often do you **play ball games** at Central Park?

Which **ball games** do you **play at Central Park**?

What else do you do at Central Park?

- Evaluate the significance of the patterns and discover related ones

Pattern score = 0.6

\{Baseball playAt Central\_Park. Permit getAt "www.permits.org"\}

- Produce a concise output that summarizes the findings

*Data-Driven Crowdsourcing*
Architecture Sketch

User Interface

- User
- User/worker Profile
- Knowledge Base
- Inferred knowledge updates

Query Engine
- Query
- Budget, preferences
- Request refinement
- NL request

NL Parser / Generator
- NL task
- NL answer
- Next Crowd worker

Crowd Task Manager
- Answer aggregation
- Significance function
- Overall Utility
- Task, preferences

Crowd results
- Summarized general
- Summarized individual
- Inference and summarization

Significant results
- Results summary
- Task, preferences

Crowd Selection
- Reward

Next Crowd worker

Inferred
- Individual
- General

User data
- User/worker Profile
- Raw crowd results
- Summarized crowd results

Data-Driven Crowdsourcing

EU-FP7 ERC

MoDaS
Mob Data Sourcing
Knowledge Repository

Different types of knowledge:

• A general knowledge base is **input** to the system

• Knowledge **inferred** in previous query evaluation

  – **General knowledge** – completes the knowledge base
    May be annotated with trust/error probability
  
  – **Individual knowledge** – more volatile
    may be annotated with user properties
Formal Model Based on RDF

Ontology of general facts

DB of personal history per crowd member

<table>
<thead>
<tr>
<th>T1</th>
<th>I visited the Bronx Zoo and ate pasta at Pine on April 5th</th>
<th>[Visit doAt Bronx_Zoo]. [Pasta eatAt Pine]</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td>I played basketball in Central Park on April 13th</td>
<td>[Basketball playAt Central_Park]</td>
</tr>
<tr>
<td>T3</td>
<td>I played baseball in Central Park and ate falafel at Maoz Veg. on April 27th</td>
<td>[Baseball playAt Central_Park]. [Falafel eatAt Maoz_Veg]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Formal Model Based on RDF

Different types of knowledge:
• A general knowledge base is input to the system
• Knowledge inferred in previous query evaluation can be recorded
  – General knowledge: completes the knowledge base, may be annotated with trust/error probability
  – Individual knowledge: more volatile, may be annotated with user properties

Input general
Inferred
general
Inferred
individual
Shake
Shack
Grimaldi's
People
Frequently eat at

Data-Driven Crowdsourcing
The user query may be formulated in a formal language.

E.g., OASSIS-QL is a SPARQL-based query language for crowd mining.

[SIGMOD’14, SIGMOD’15]

Find popular combinations of an activity in a child-friendly attraction at NYC and a restaurant nearby (plus relevant advice)

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

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

Crowd task:
isSignificant({\text{Biking doAt Central Park}},...)
Budget: $0.5
User preferences: ...

$\text{x} = \text{Central Park},$
$\text{y} = \text{Biking},$
$\text{z} = \text{Maoz Veg.}$

“How often do you go biking at Central Park and eat at Maoz Vegetarian?”

“Once a month.”
(support = 12/365)
What is a good algorithm?

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
Efficient Query Evaluation Algorithm

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

```
Biking doAt Park
```

```
Biking doAt Central_Park
```

```
Biking doAt Central_Park.
Basketball playAt Central_Park
```
The algorithm:

- Lazily construct the semantic subsumption partial order
- Traverse it in a top-down manner
- Prune insignificant parts

- See complexity analysis in the paper
Sometimes theory fails...

Practical (Crowd) Aspects of the Algorithm

• Asking a sequence of questions “in context”
• Quick pruning of irrelevant items by crowd members

• Open questions – letting crowd members specify patterns

“What else do you do when you play basketball in Central Park?”
Crowd Task Manager

- Distributes tasks to crowd members
- Aggregates and analyzes the answers
- Dynamically decides what to ask next

**Crowd task:**

\[
\text{isSignificant}\{\text{Baseball doAt Central_Park}\} \\
\text{Budget: $0.5} \\
\text{User preferences: ...}
\]

"How often do you play baseball at Central Park?"

**Answer 1:** never (score=0)

**Answer 2:** once a week (score=1/7)

**Aggregation:** estimated mean \( M \)

**Significance:** \( \Pr(M \geq \Theta) \geq 0.5 \)

**Overall utility:** next question expected to reduce error probability by 0.1
Crowd Task Manager

Aggregation, significance and utility choices depend on the type of data collected from the crowd.

For **individual** data, the aggregated answer should account for diverse opinions

- e.g., statistical modeling

For **general** data the aggregated answer should reflect the truth

  e.g., weighing by expertise, outlier filtering

---

Data-Driven Crowdsourcing
Theoretical research focuses on computing some DB operator with the crowd:
Sort, Top-K, Group-by, skyline, join...

- Some underlying ground truth is often assumed
- Typically Boolean questions ("is a>b ?")
- Simple error model (user error < 0.5, given overall error bound)
- Mostly lower and upper bounds on the number of required questions
Sometimes theory fails (2)

Theoretical research focuses on computing some DB operator with the crowd:
Sort, Top-K, Group-by, skyline, join...

- Some underlying **ground truth** is often assumed
- Typically **Boolean questions** ("is a>b ?")
- Simple **error model** (user error < 0.5, given overall error bound)
- Mostly lower and upper bounds on the **number of required questions**

When high accuracy is required and the crowd error is high...

[Salable filtering algorithms. Groz, M. ICDT16]
Crowd Selection

- **By explicit user requirements:**
  crowd members should be NYC residents

- **By expertise:**
  based on past answers regarding general knowledge

- **By similarity to the user:** based on profiles, past questions and answers regarding individual data
  - Reward crowd members accordingly
Declarative User Selection

Ann plans to go to Australia in January, looks for a recommended Surfing beach near Sydney

```
1 SELECT VARIABLES $x
2 WHERE
3   {$x instanceOf Place.
4     $x near Sydney}
5 SATISFYING
6   {Surfing doAt $x.
7     _ in January}
8 ORDER BY DESC(SUPPORT)
9 LIMIT 5
```
## User Profiles

<table>
<thead>
<tr>
<th>profile(Ann):</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF livesIn</td>
<td>Paris</td>
</tr>
<tr>
<td>SELF hasGender</td>
<td>Female</td>
</tr>
<tr>
<td>SELF hasHobby</td>
<td>Photography</td>
</tr>
<tr>
<td>SELF hasHobby</td>
<td>Bird_Watching</td>
</tr>
<tr>
<td>SELF graduatedFrom</td>
<td>NYU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>profile(Bill):</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF livesIn</td>
</tr>
<tr>
<td>SELF hasGender</td>
</tr>
<tr>
<td>SELF hasHobby</td>
</tr>
<tr>
<td>SELF graduatedFrom</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>profile(Carol):</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELF livesIn</td>
</tr>
<tr>
<td>SELF hasGender</td>
</tr>
<tr>
<td>SELF hasHobby</td>
</tr>
</tbody>
</table>
## Extended profile (User Answers)

**answers(Ann):**

<table>
<thead>
<tr>
<th>Fact-set</th>
<th>Support</th>
<th>Frequency</th>
<th>IC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sport doAt Place</td>
<td>0.8</td>
<td>0.8</td>
<td>0.08</td>
</tr>
<tr>
<td>Surfing doAt Place</td>
<td>0.1</td>
<td>0.01</td>
<td>0.96</td>
</tr>
<tr>
<td>Surfing doAt Santa_Cruise</td>
<td>0.005</td>
<td>0.001</td>
<td>0.99</td>
</tr>
<tr>
<td>Volleyball doAt Place</td>
<td>0.001</td>
<td>0.02</td>
<td>0.93</td>
</tr>
</tbody>
</table>

**answers(Bill):**

<table>
<thead>
<tr>
<th>Fact-set</th>
<th>Support</th>
<th>Frequency</th>
<th>IC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sport doAt Place</td>
<td>0.85</td>
<td>0.8</td>
<td>0.08</td>
</tr>
<tr>
<td>Surfing doAt Place</td>
<td>0.08</td>
<td>0.01</td>
<td>0.96</td>
</tr>
<tr>
<td>Surfing doAt Wanda_Beach. _ in January</td>
<td>0.012</td>
<td>0.0001</td>
<td>0.99</td>
</tr>
<tr>
<td>Volleyball doAt Place</td>
<td>0</td>
<td>0.02</td>
<td>0.93</td>
</tr>
</tbody>
</table>

**answers(Carol):**

<table>
<thead>
<tr>
<th>Fact-set</th>
<th>Support</th>
<th>Frequency</th>
<th>IC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sport doAt Place</td>
<td>0.2</td>
<td>0.8</td>
<td>0.08</td>
</tr>
<tr>
<td>Surfing doAt Place</td>
<td>0.026</td>
<td>0.01</td>
<td>0.96</td>
</tr>
<tr>
<td>Surfing doAt Long_Reef. _ in January</td>
<td>0.013</td>
<td>0.00009</td>
<td>0.99</td>
</tr>
<tr>
<td>Drawing doAt Place</td>
<td>0.15</td>
<td>0.05</td>
<td>0.84</td>
</tr>
</tbody>
</table>
Declarative User Selection

```sql
ASSIGN BY Ann TO $u
FROM ontology WHERE
  {$x instanceOf Place.
   $x near Sydney}
FROM profile($u) WHERE
  {SELF livesIn Sydney}
FROM tran($u) WHERE
  {{Surfing doAt []} WITH SUPPORT > 0.02
SIMILAR profile($u) TO profile(Ann)
  WITH SIMILARITY >= 0.75
SIMILAR tran($u) TO tran(Ann)
  WITH SIMILARITY >= 0.75
SIMILAR tran($u) TO
  {Surfing doAt $x.
   _ in January} AS surfhabit
  WITH SIMILARITY AS surfSim >= 0
ORDER BY surfSim LIMIT 1
```

Crowd members who live in Sydney and surf frequently

With profile and past answers similar to Ann’s

Surfing near Sydney in January (or similar)
Task Manager (revisited)

- Distributes tasks to selected crowd members
- Aggregates and analyzes the answers
- Dynamically decides what to ask next

Theoretical research here typical focuses on optimal task distribution

- Some crowd expertise levels are assumed/dynamically inferred
- Some tasks difficulty levels are assumed/dynamically inferred
- Some bound on the possible worker load is assumed
- Maximization of result quality while minimizing time/cost

Data-Driven Crowdsourcing
Still, sometimes queries are too difficult...

Solution: **Dismantle** into easier ones, then **Reassemble**

Can we do it all automatically with the crowd?
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 queries

**Input:** Query ("Select BMI from Pictures") and Budget

**Using:** Value, Dismantling, and Example crowd questions

**Output:**

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

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

Natural language interface

Data-Driven Crowdsourcing
The Case for Natural Language (NL)

• General and individual knowledge needs can be mixed in an intricate manner in NL

  “What are the most interesting places near Forest Hotel, Buffalo, we should visit in the fall?”

• Our goal:
  – identifying the knowledge needs of each type
  – and translating them into formal queries
Challenges

- **Distinguishing** general from individual expressions in the question

  - Opinion Mining tools can detect some individual expressions (opinions, preferences) but not all (habits and practices)

  - Matching to a knowledge base is insufficient to detect general expressions, since knowledge bases are dirty and incomplete

- **Translating** each expression accordingly

  - Existing NL-to-query translation tools rely on a knowledge base and are thus irrelevant for individual expressions

- **Integrating** the results into a well-formed query
New modules & contributions are painted black [SIGMOD’15, VLDB’15]

Input: question parsed using an NL Parser

Detect individual expressions (IXs)

Translate individual expressions

Translate general expressions

Knowledge base(s)

FREyA

Translate general expressions

IX detection patterns

Vocabularies

Translation patterns

Compose into a formal query

output: well-formed query, to be evaluated by crowd mining tools

Data-Driven Crowdsourcing
The IX detection Module

- The parsed NL sentence can be represented as a directed, labeled graph.
- We use SPARQL-like selection patterns and vocabularies to detect Ixs within the graph, and then other patterns to select the full Ixs.

A (simplified) example: a verb (VB) whose subject (nsubj) is individual.

```
filter($y in V_participant &&
(POS($x) = "VBP" ||
POS($x) = "VB")
```
Detection & Translation Patterns

• IX detection
  – Lexical Individuality (e.g. “interesting place” vs. “northern place”)
  – Participant individuality (e.g. “we visit” vs. “Tim Duncan visits”)
  – Syntactic individuality (e.g. “Tim Duncan should play for the NY Knicks”)

• Query creation
  – NL to OASSIS-QL mapping

\[
{x \text{ amod } y} \rightarrow \{\text{tran($x$)} \text{ hasLabel “tran($y$)”}\}
\]

| 6 | hasLabel | "interesting" |
Before We Conclude

Other crowdsourcing systems can be put in terms of the architecture for comparing and identifying possible extensions.

- NL to query translators
- Majority vote, costume function
- # questions is fixed or bounded
- Declarative crowdsourcing platforms
- Crowdsourced entity resolution
- Task to worker assignment
Summary

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:
• Almost no theory (and when exists, too “clean”)
• (very) interactive computation
• A huge amount of data
• Varying quality and trust
Thanks

Antoine Amerilli, Yael Amsterdamer, Moria Ben-Ner, Rubi Boim, Susan Davidson, Ohad Greenshpan, Benoit Gross, Yael Grossman, Ana Itin, Ezra Levin, Ilia Lotosh, Slava Novgordov, Neoklis Polyzotis, Sudeepa Roy, Pierre Senellart, Amit Somech, Wang-Chiew Tan, Brit Youngmann, ...

EU-FP7 ERC
MoDaS
Mob Data Sourcing