MoDaS

Mob Data Sourcing

Asking the Right Questions in Crowd Data Sourcing

Tova Milo

TEL AVIV UNIVERSITY אוניברסיטת תל-אביב

Outline

- Introduction to crowd (data) sourcing
- Databases and crowds
- Declarative is good
- How to best use resources
- Conclusion

Ack: Some slides are borrowed (with permission) from the VLDB’11 tutorial [DFKK11].

Disclaimer: - Very high level
- More questions than answers
- Some nudity ☺

Crowd Sourcing 101

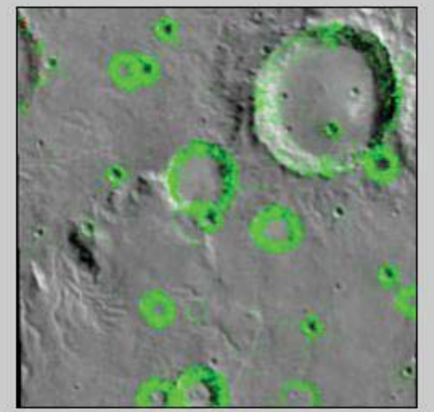
Billions of devices

Crowd Sourcing 101

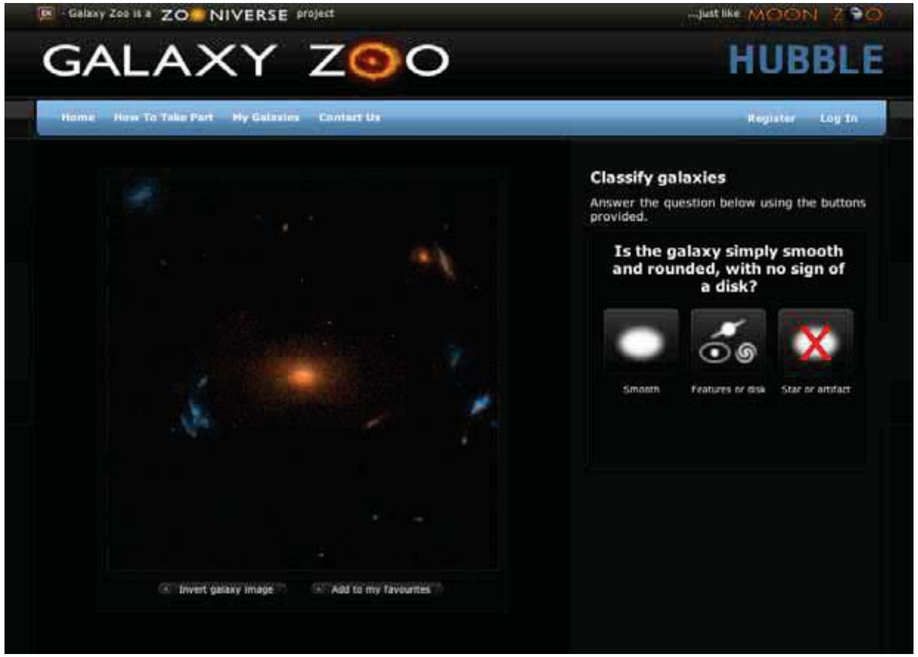
Ubiquitous connectivity

Examples

Citizen science



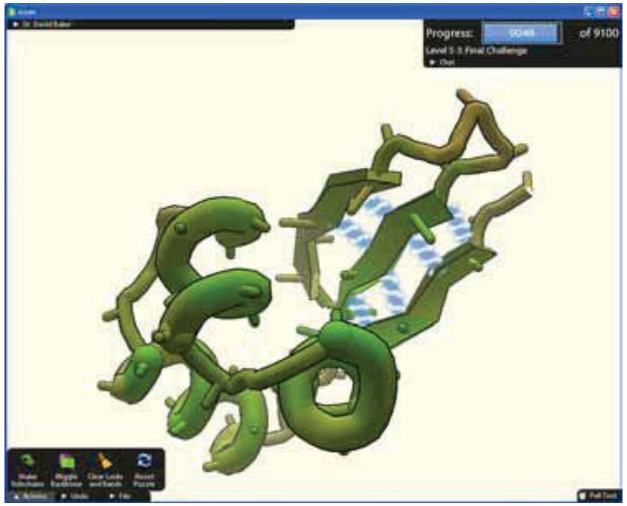
Pixels indicate Clickworker's identified craters



Classify galaxies

Answer the question below using the buttons provided.

Is the galaxy simply smooth and rounded, with no sign of a disk?

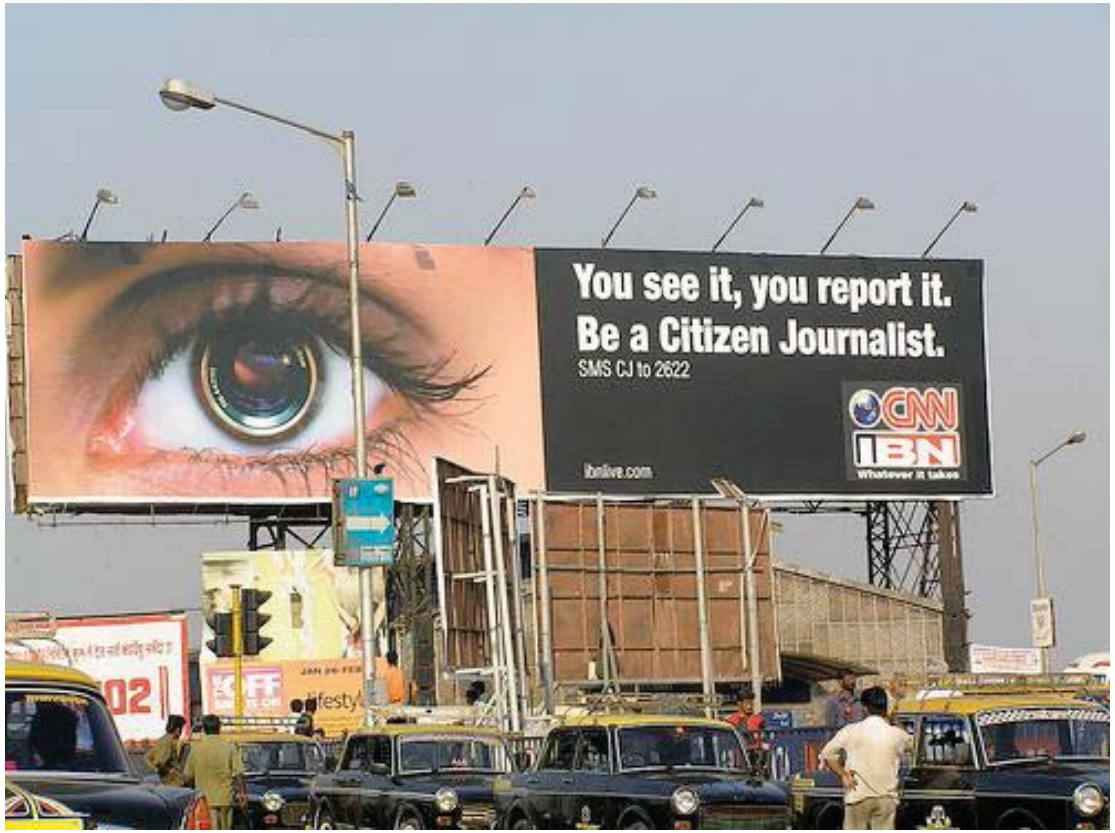


foldit BETA

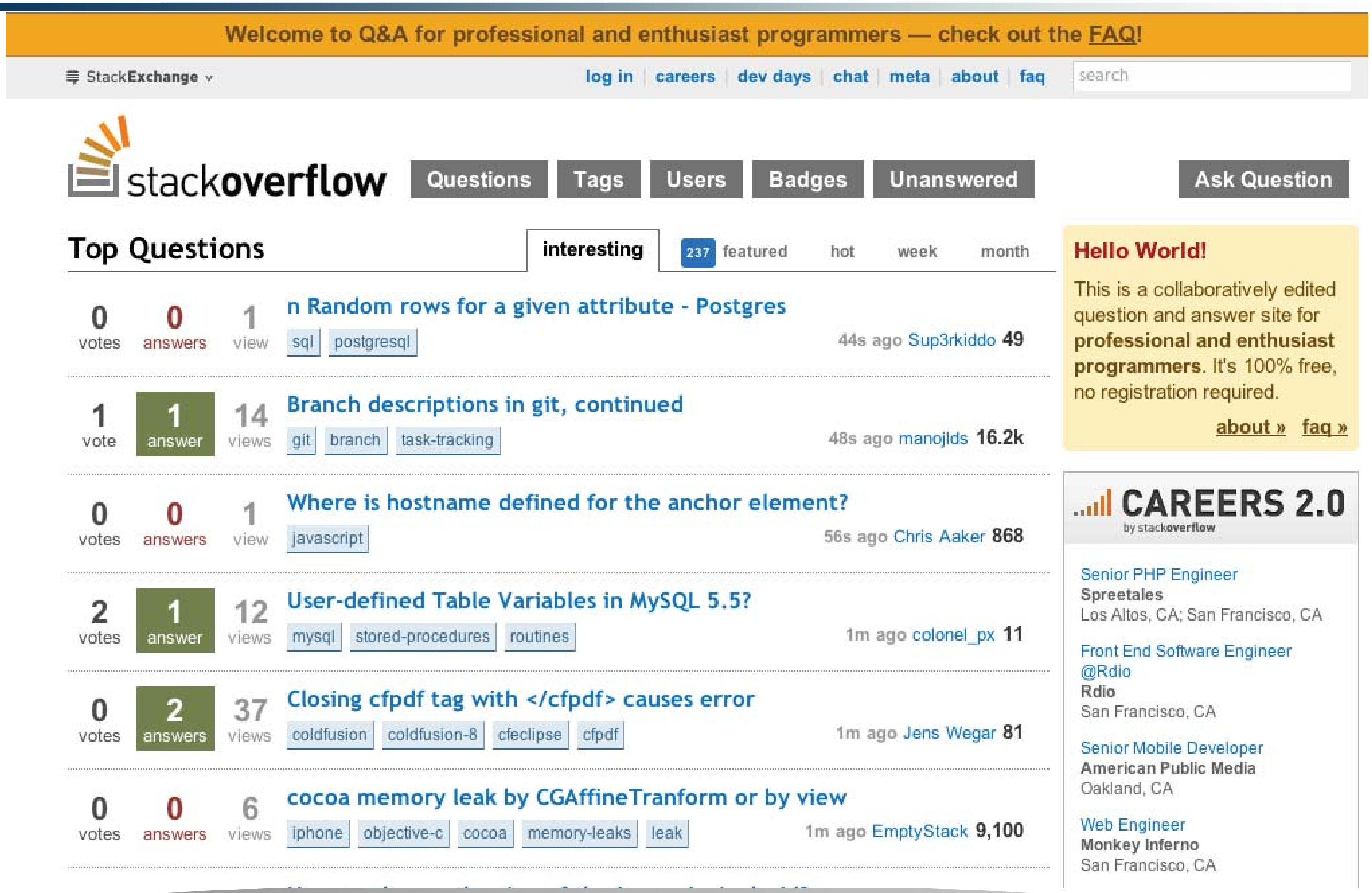
Solve Puzzles for Science

Examples

Citizen journalism and sensing



Examples



Welcome to Q&A for professional and enthusiast programmers — check out the FAQ!

Top Questions

0 votes 0 answers 1 view — n Random rows for a given attribute - Postgres — sql postgresql — 44s ago Sup3rkiddo 49

1 vote 1 answer 14 views — Branch descriptions in git, continued — git branch task-tracking — 48s ago manojlds 16.2k

0 votes 0 answers 1 view — Where is hostname defined for the anchor element? — javascript — 56s ago Chris Aaker 868

2 votes 1 answer 12 views — User-defined Table Variables in MySQL 5.5? — mysql stored-procedures routines — 1m ago colonel_px 11

0 votes 2 answers 37 views — Closing cfpdf tag with </cfpdf> causes error — coldfusion coldfusion-8 cfeclipse cfpdf — 1m ago Jens Wegar 81

0 votes 0 answers 6 views — cocoa memory leak by CGAffineTranform or by view — iphone objective-c cocoa memory-leaks leak — 1m ago EmptyStack 9,100

Hello World!

This is a collaboratively edited question and answer site for **professional and enthusiast programmers**. It's 100% free, no registration required.

about » faq »

CAREERS 2.0
by stackoverflow

Senior PHP Engineer
Spreetales
Los Altos, CA; San Francisco, CA

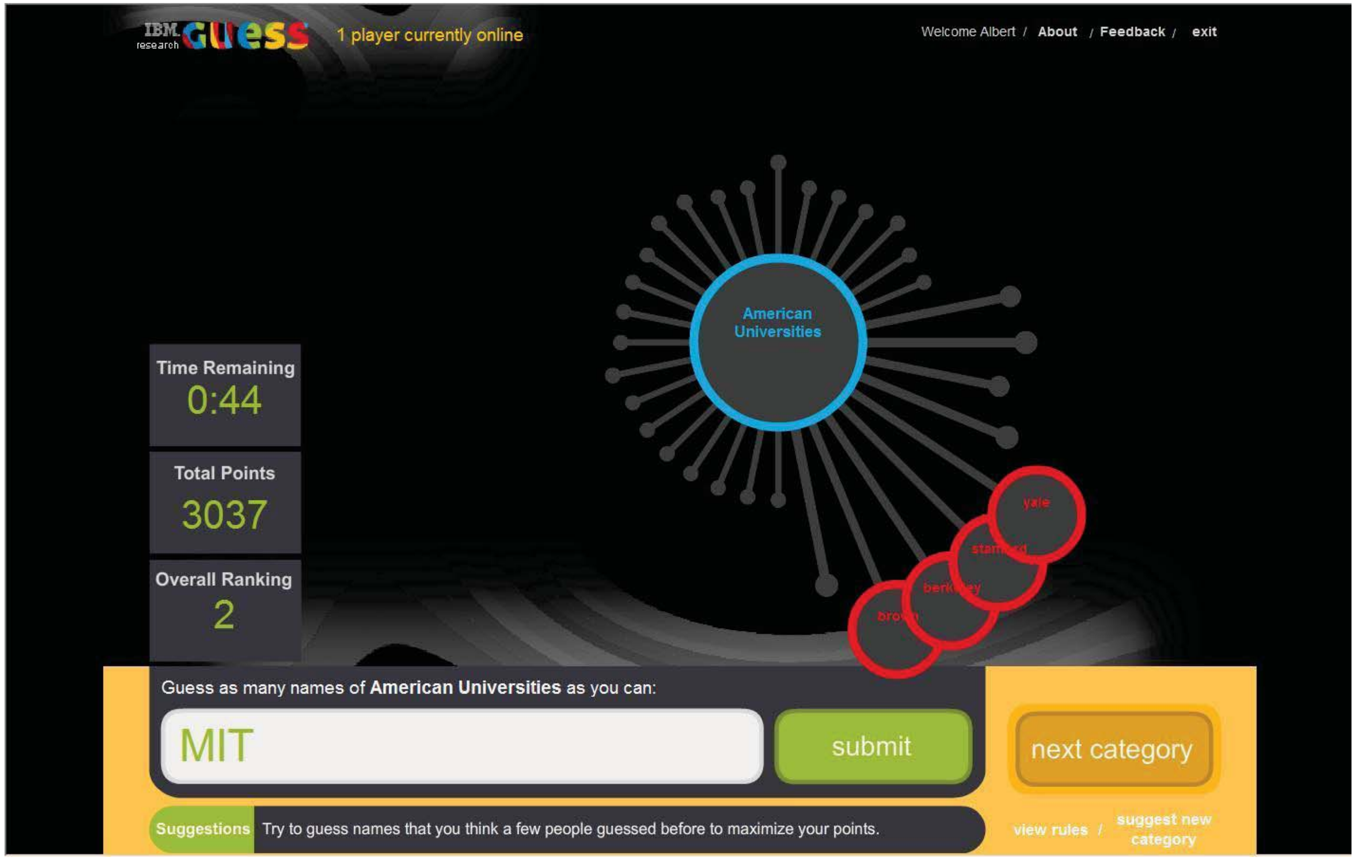
Front End Software Engineer @Rdio
Rdio
San Francisco, CA

Senior Mobile Developer
American Public Media
Oakland, CA

Web Engineer
Monkey Inferno
San Francisco, CA

Examples

Games are fun!


IBM research Guess
1 player currently online
Welcome Albert / About / Feedback / exit
American Universities
yale
stanford
berkeley
brown
submit
next category
view rules / suggest new category

Time Remaining 0:44

Total Points 3037

Overall Ranking 2

Guess as many names of **American Universities** as you can:

MIT

Suggestions Try to guess names that you think a few people guessed before to maximize your points.

So what is it all about?

- Bederson & Quinn (Human Computation) CHI’11
  - Motivation (Pay, altruism, enjoyment,…)
  - Quality control (we’ll talk more about that)
  - Aggregation (We’ll also talk more about that)
  - Human skills (Visual recognition, language, …)
  - …

Outline

- Introduction to crowd data sourcing
- Databases and crowds
- Declarative is good
- How to best use resources
- Conclusion

Databases and Crowds

- How can crowds help databases?
  - Fix broken data: entity resolution, inconsistencies
  - Add missing data
  - Subjective comparisons

- How can databases help crowd apps
  - Lazy data acquisition (only get the data that is needed)
  - Manage the data sourced from the crowd
  - Semi automatically create user interfaces

Database platforms for Crowd-based Data Sourcing

- Data models, query languages
  (query processing, optimization,…)
  - Qurk (MIT)
  - CrowdDB (Berkley, ETH)
  - sCOOP (Stanford, UCSC)
  - FusionCOMP (TsuKuba)
  - MoDaS (Tel Aviv University)
  - …

- Data quality

- Asking (the crowd) the right questions

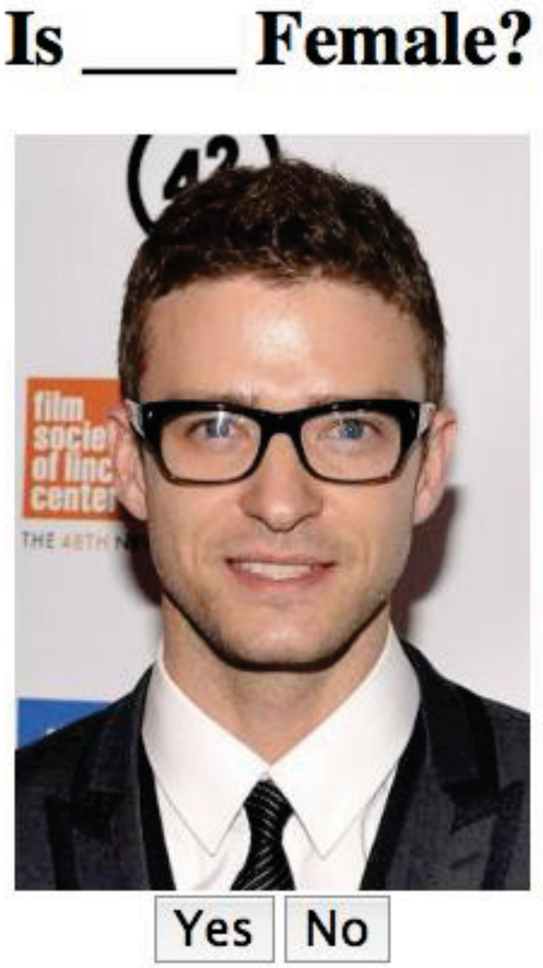
Qurk (MIT)

- Goal: crowd-source comparisons, missing data
- Basis: SQL3 + UDF
  - UDF encapsulates crowd input
  - Special template language for crowd UDFs
  - Specify UI, quality control, possibly opt. hints

- References:

  [Marcus et al, CIDR’11, SIGMOD’11]

Qurk example



Is ____ Female?

ɪnen in a “people” database

ple(

(256),

e(p);

```
TASK isFemale(tuple) TYPE:Filter
    Question: “is %s Female”,
                    Tuple[“photo”]
    YesText: “Yes”
    NoText: “No”
```

The magic is in the templates

- Templates generate UIs for different kinds of crowd-sourcing tasks
  - Filters: Yes/No questions
  - Joins: comparisons between two tuples (equality)
  - Order by: comparisons between two tuples (>=)
  - Generative: crowdsource attribute value

- Templates also specify quality control; e.g.

  COMBINER: MajorityVote

But, can you trust the Crowd?

Spencer Tunick

Many questions

- How to determine correctness ?
- How to clean the data?
- What questions to ask?
- Who to ask? (How many? When to stop?)
- How to best use resources?

Outline

- Introduction to crowd data sourcing
- Databases and crowds
- Declarative is good (but we need more…)
- How to best use resources
- Conclusion

Example: Conflicts resolution

- Average value? Majority vote? Probabilistically?
- But some people know nothing about a given topic
- So maybe a “biased (probabilistic) vote”?
- But how to bias?
- A “chicken or the egg” problem:

  To know what is true we need to know who to believe.
  But to know this we need to know who is usually right
  (and in particular, what is true..)

Example: So what can we do?

- Start with some estimation on the trust in users
- Gain confidence in facts based on the opinion of users that supported them
  - Give bigger weight to user that we trust
- Then update the trust level in users, based on how many of the facts which they submitted, we believe
- Iterate until convergence

  Trusted users give us confidence in facts,

  and users that supported these facts gain our trust…

[Galland et al, WSDM 2010]

- And there is also a probabilistic version…

But what do we want?

- Not yet another data cleaning algorithm
- We want to have easy control on the employed policy (for data cleaning, query selection, user game scores,…)
- We really don’t want to (re)write Java code (for each tiny change!)
- We want (seamless) optimization, update propagation,…

Database approach:

Define a **declarative language** for specifying policies

[Deutch, Greenshpan, Kostenko, M. ICDE’11 ,WWW’12]
[Deutch, Koch, M. PODS’10]

Proposed language

- Add to SQL (relational algebra) a REPAIR-KEY construct

  REPAIR-KEY “repairs” key violations in the database, choosing one possible option, probabilistically, according to the support

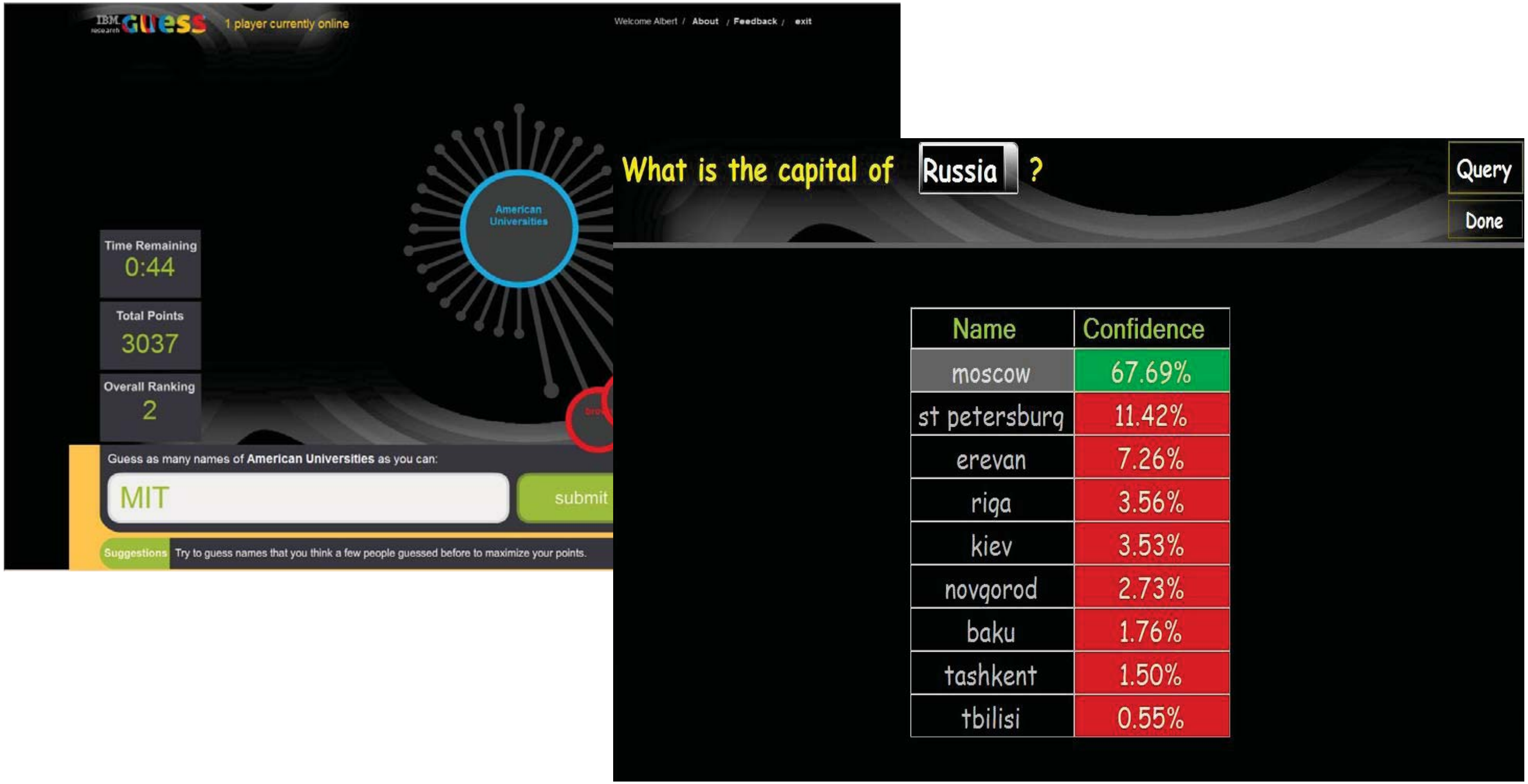
- And a WHILE construct

- Semantics: Markov chain of DB instances. Probability of a fact to hold in a given instance.

Name	Cuisine	support
Anton’s	French	0.8
Anton’s	Continental	0.2
McDonald	FastFood	1.0
…	…	

- Expresses nicely common policies for cleaning, selection of questions, scoring answers

TriviaMaster (ICDE 2011 demo)



IBM research Guess
1 player currently online
Welcome Albert / About / Feedback / exit
American Universities
submit
Query
Done

Time Remaining 0:44

Total Points 3037

Overall Ranking 2

Guess as many names of **American Universities** as you can:

MIT

Suggestions: Try to guess names that you think a few people guessed before to maximize your points.

What is the capital of Russia ?

Name	Confidence
moscow	67.69%
st petersburg	11.42%
erevan	7.26%
riga	3.56%
kiev	3.53%
novgorod	2.73%
baku	1.76%
tashkent	1.50%
tbilisi	0.55%

Some complexity results

Formal problem: Given a Markov Chain of database instances and an SQL query on the database ("what is Anton's cuisine ?"), compute the probabilities of the different answers.

- Theorem: Exact computation is #P-hard
- Theorem: If Markov Chain is **ergodic**, computable in EXPTIME
  - Compute the stochastic matrix of transitions
  - Compute its fixpoint
  - For ergodic Markov Chain it corresponds to correct probabilities
  - Sum up probabilities of states where the query event holds
- Theorem: In general, 2-EXPTIME
  - Apply the above to each connected component of the Markov Chain
  - Factor by probability of being in each component

Some complexity (cont.)

Approximations:

- Absolute approximation: approximates correct probability ±ε
- Relative approximation: approximates correct probability up to a factor in-between (1- ε), (1+ ε).

[Relative is harder to achieve]

Language	Exact computation	Relative approx	Absolute approx
(Linear) datalog	#P-hard In PSPACE	NP-hard	In PTIME
Inflationary fixpoint	#P-hard In PSPACE	NP-hard	In PTIME
Non-inflationary fixpoint	#P-hard In (2)EXP-TIME	NP-hard	NP-hard; PTIME in input size and mixing time

Still lots of open questions

- How (and when) can we evaluate things fast enough?
- How to store the vast amount of data?
  - Distributed Databases? Map-reduce?
- The data keeps changing. How to handle updates?
- …

Outline

- Introduction to crowd data sourcing
- Databases and crowds
- Declarative is good
- How to best use resources
- Conclusion

Partial knowledge

	q1	q2	q3	q4	q5	q6	…		
u1	a	5		b					
u2	a		3						
u3		5	3	b					
u4	b	2	3						
u5	c		3	a					
…									

- Goal: Compute an aggregate function f for each query, e.g.
  - Some metric of the distribution (e.g. entropy)
  - Most frequent answer
  - Aggregated value (e.g. average)

Increasing knowledge

- Limited overall resources
- Limited user availability
- Bounded resources per question

Which cells to resolve?

[Boim, Greenshpan, M., Novgorodov, Polyzotis, Tan. ICDE’12,…]

Quantifying uncertainty

- Assume t answers suffice for computing f for q
- Comp(q): all possible completions of q’s column
- Dist(r – r’): distance between two results of f
- Uncertainty(q): max{ Dist(f(X) - f(Y)) | X,Y in Comp(q) }

  i.e. the largest distance between possibly completions

Quantifying uncertainty (cont.)

- Uncertainty measures for a Users-Answers matrix M
  - **Max-uncertainty(M)**
  - **Sum-uncertainty(M)**

- **Problem statement (X-uncertainty Reduction)**

  Given a matrix M, a choice x ϵ {max,sum}, and a set of constraints, identify a set C of empty cells that satisfy the constraints and where

  Max $_{M' \in M_C}$ **X-uncertainty(M')** is minimized.

  Where M_C contains all possible matrices that we can derive from M by resolving solely the cells in C.

Example

- Target function
  - – Entropy, average, most frequent,…
- Constraints
  - – A: bound k on the over number of cells
  - – B: also a bound k’ on questions per users
  - – C: here k’ is a bound on users per question

Some complexity results

- **max-Uncertainty Reduction**

  in PTIME for all constraints classes
  - Greedy algo for constraints class A (and C)
  - Using Max-flow for constraints class B

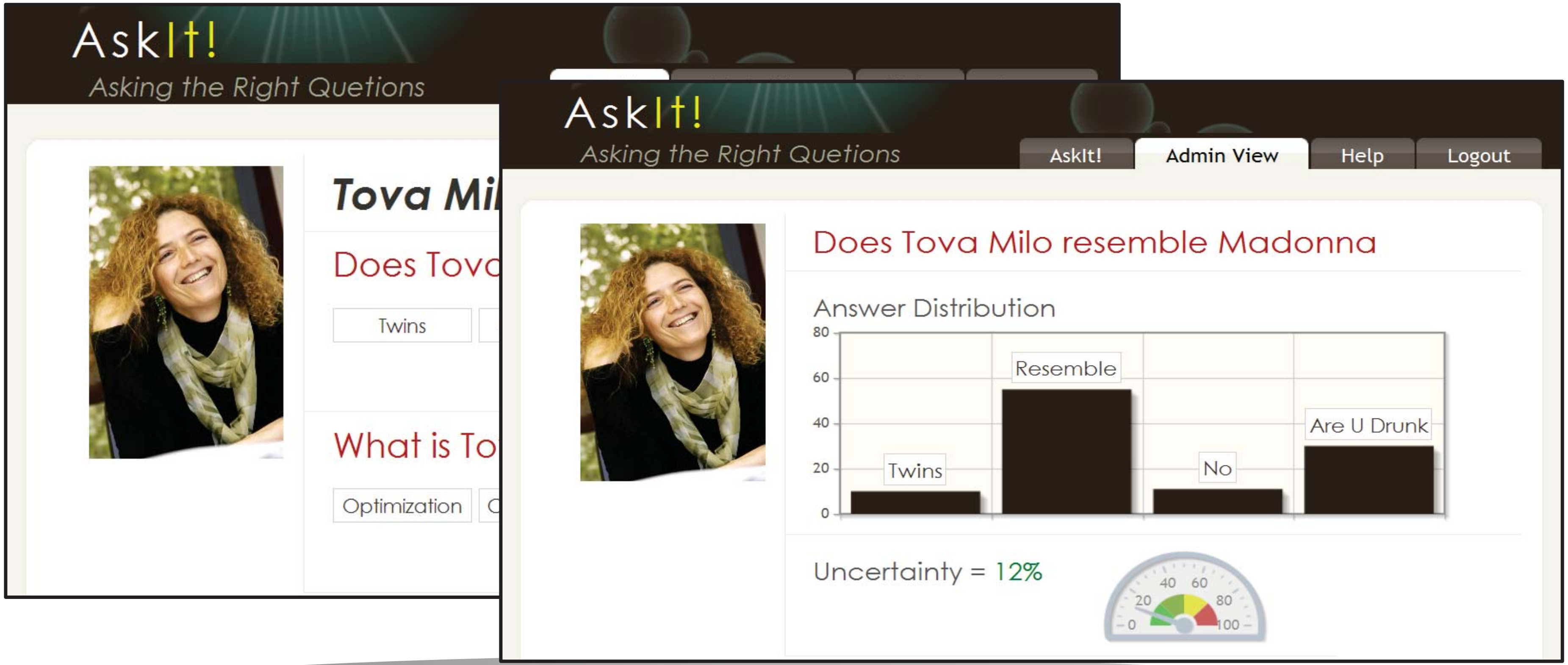
- **sum-Uncertainty Reduction**

  in PTIME for constraint classes A and C
  - Dynamic programming

  NP-COMPLETE for constraints class B
  - Reduction for perfect 3 set cover

AskIt (ICDE’12 demo)

- Gather information (scientific as well as fun)
  on ICDE’12 authors, participants, papers, presentations,…



Tova Mi

Does Tov

What is To

Does Tova Milo resemble Madonna

Uncertainty = 12%

Lots of open questions

- Use prior knowledge about users/answers
    - Predict answers
    - Predict who can/will answer what

  [Collaborative Filtering-style analysis is useful here]

- Worse-case analysis vs. expected error

- Treat other goal functions

- Optimization

- Incremental computation

…

Outline

- Introduction to crowd data sourcing
- Databases and crowds
- Declarative is good
- How to best use resources
- Conclusion

Conclusion

- All classical issues:

  Data models, query languages, query processing, optimization, HCI
- BUT
  - (Very) interactive computation
  - (Very) large scale data
  - (Very) little control on quality/reliability
  - Closed vs. open world assumption

!תודה

Thanks!

Merci!