CompSci 590.6
Understanding Data: Theory and Applications

Lecture 20
Crowd Sourcing: Max Operator

Instructor: Sudeepa Roy
Email: sudeepa@cs.duke.edu

Fall 2015
Today’s Reading

1. So Who Won? Dynamic Max Discovery with the Crowd
   Guo-Parameswaran- Garcia-Molina
   SIGMOD 2012
   (slides available online)

2. Top-k and Clustering with Noisy Comparisons
   Davidson-Khanna-Milo-Roy
   ICDT 2013/TODS 2014
   (following slides)
Humans intelligence in performing database tasks

Crowdsourcing

Data collection
Data curation
Integration
Join
Search
Top K/Max
Clustering (Group-By)

.....
Example

Q. Group the photos of individual players

Q. Find their most recent photos
How a DBMS Thinks

Q. Group the photos of individual players

Group-By Queries
Use “Name” attribute

Q. Find their most recent photos

Max/Top-k Queries
Use “Date” attribute

What if name/date is missing
Image processing? Photo forensics?
Ask the “Crowd”!
How the Crowd Thinks

Q. Group the photos of individual players
How the Crowd Thinks

Q. Find their most recent photos
Crowd Sourcing

• Using human intelligence to do tasks which are harder to automate

• Many crowdsourcing platforms

• Many recent crowd-powered databases
  - Qurk, CrowdDB, Deco
This lecture: Max

A possible query:

```sql
SELECT TOP 1 R.picture
FROM SoccerPlayerPhotoTable AS R
GROUP BY R.player
ORDER BY R.date DESC
```

Fixed but unknown attributes: R.player, R.date
Framework at a Glance

**Internal Computation (Top-K/Group-By)**

- Value(a) > Value(b) → Yes/No
- Value(b) > Value(c) → Yes/No

Crowd-sourced DBMS

**Comparison Error**
Crowd’s answer may be wrong with some prob.

**Crowd**

**Cost Model**
- Asking questions costs money
  - Additive
  - Count #comparisons

We still want the correct answer w.h.p.
Our Goal

Minimize the total #comparisons
while outputting the correct max
w.p. ≥ 1 − δ (given constant δ > 0)
Elements: Type and Value

- #Elements = n
  - (n = 16)
  - Two attributes: Type and Value

- Type
  - e.g. Name = “Maradona”
  - used in Group-By
  - #Types = J (J clusters, J = 4)

- Value
  - e.g. Date when photo was taken
  - used in Top-K

- Unknown, but “ground-truth” exists for Types and Values
Type and Value Comparisons

• DB Queries vs. Comparisons (questions to the crowd)

• Type Comparisons:
  – Type(x) = Type(y)?

• Value Comparisons:
  – Value(x) > Value(y)?
  – Assumes same type

• Answer is Boolean
  – Cannot ask “What is Type(x)/Value(x)?”

First, assume no comparison error
• How many comparisons for n elements (at least and at most)?
• What are the pros and cons for the above two trees?
Type and Value Comparisons

• DB Queries vs. Comparisons (questions to the crowd)

• Type Comparisons:
  – Type(x) = Type(y)?

• Value Comparisons:
  – Value(x) > Value(y)?
  – Assumes same type

• Answer is Boolean
  – Cannot ask “What is Type(x)/Value(x)??”

But, answers are not always correct

• Crowd makes mistakes
• Next, error model
Constant Error Model

Type comparisons: Type(x) = Type(y)?
Value comparisons: Value(x) > Value(y)?
(for MAX – value comparisons only)

Constant error model:

- Standard model
- Probability of wrong answer
  \[ \leq \frac{1}{2} - \varepsilon, \quad \varepsilon > 0 \]

Is the first photo older?
Wrong: w.p. \[ \leq \frac{1}{2} - \varepsilon \]
Correct: w.p. \[ \geq \frac{1}{2} + \varepsilon \]

Same person?
Wrong: w.p. \[ \leq \frac{1}{2} - \varepsilon \]
Correct: w.p. \[ \geq \frac{1}{2} + \varepsilon \]
Algorithm for Constant Error Model

• Exact comparisons: Binary tree structure is not necessary
• Noisy comparisons: Repeat comparison + majority vote
• Constant error model: $\theta(n)$ algorithm (Feige et. al. ’94)
• Our goal: Total no. of comparisons = $n + o(n)$ cannot repeat even twice in most of the internal nodes
Analysis of algorithm for constant error model on board
Variable Error Model

For value comparisons only: Value(x) > Value(y)?

• Error probability < 1/ \( f(\Delta) \), \( f \) = a strictly growing function

\[ \Delta = \text{Distance of } x, y \text{ in sorted order} \]

\[ \Delta = 2 \]

For “any” strictly monotone function \( f \), \( n + o(n) \) comparisons suffice to find \( \max \) under variable error model

- \( n + O(1) \) for \( f(\Delta) = e^\Delta \)

\[ \leq 1/e \]

\[ \leq 1/e^2 \]

e.g. Error probability when \( f(\Delta) = e^\Delta \)

Is the first photo older? - Harder

Is the first photo older? – Easier
Main Steps for Max

Goal: $n + o(n)$

- **Upper levels**
  - Use Feige's algorithm
  - $Y$ nodes $\Rightarrow \Theta(Y)$ cost

- **Lower levels**
  - Just 1 comparison at each internal node
  - No majority vote

**Key idea:**

Start with a random permutation of elements at the leaves

- **Max does not lose in the lower levels w.h.p.**
- **Intuition:** Max does not meet 2\textsuperscript{nd} Max in the lower levels w.h.p.
- **Total no. of comparisons** = $n + \Theta(Y) = n + o(n)$

3/20/2013  ICDT 2013
Clustering with Correlated Types and Values

Full correlation:
- Elements of same type form contiguous blocks in the sorted order on values

\[ O(n \log J) \] value and type comparisons suffice to find the clusters

For no correlation and no error: lower bound: \( \Omega(nJ) \), upper bound \( O(nJ) \)
• Linear cost in total in each inner iteration
• #Inner iterations = $O(\log J)$
• Cost: $C(n) \leq C(n/2) + O(n \log J)$
• $C(n) = O(n \log J)$

4J blocks $\Rightarrow$ each block has $\leq s/4J$ elements, $s/4J \geq 1$
$\Rightarrow$ at most J blocks contain $\geq 2$ types (active)
#remaining elements $\Rightarrow \leq s/4 + s/4 \leq s/2$
$O(\log J)$ iterations
Partial correlation:
- Elements of same type form almost contiguous blocks

\[ O(n \log (\alpha J) + \alpha J) \] value and type comparisons suffice
• Next paper:

• “So who won...”
  – using slides available online

• Constant error model & pair-wise comparisons
  – like before