Human-in-the-loop Data Integration

Guoliang Li
Department of Computer Science, Tsinghua University, China
http://dbgroup.cs.tsinghua.edu.cn/ligl
Acknowledgement

Jianhua Feng @Tsinghua
Lizhu Zhou @Tsinghua
Beng Chin Ooi @NUS
Chen Li @UCI
Acknowledgement

Thank everyone who support me!
✓ Collaborators
✓ Students
✓ Friends
✓ …..
Data Integration (DI)

Combine data in different sources and provide users with a unified view

<table>
<thead>
<tr>
<th>Brand</th>
<th>Product</th>
<th>Region</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>iPhone6S</td>
<td>Beijing</td>
<td>4000</td>
</tr>
<tr>
<td>Apple</td>
<td>iPhone6SP</td>
<td>Beijing</td>
<td>5000</td>
</tr>
<tr>
<td>Samsung</td>
<td>Galaxy S7</td>
<td>Beijing</td>
<td>3500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Loc</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>6S 4.7’</td>
<td>Bei Jing</td>
<td>40K</td>
</tr>
<tr>
<td>6S 5.5’</td>
<td>Bei Jing</td>
<td>30K</td>
</tr>
<tr>
<td>S7</td>
<td>Bei Jing</td>
<td>35K</td>
</tr>
</tbody>
</table>

Is there any correlation between location and revenue?

Data Science Pipeline: Data Integration → Data Analysis
Data Integration (DI)

- Data Integration is important and challenging
  - New York Times
    - 80% of a data science project is to clean and integrate the data, while 20% is actual data analysis
  - Mark Schreiber of Merck
    - data scientists spend 98% of on “grunt work” and
    - only one hour per week on “useful work”

- In many communities
  - DB, AI, KDD, Web
Entity Matching in DI

Date Integration
- data acquisition, extraction, cleaning, schema matching, entity matching, etc.

Entity Matching (EM): A core problem
- Find pairs of records referring to the same entity

<table>
<thead>
<tr>
<th>Brand</th>
<th>Product</th>
<th>Region</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>iPhone6S</td>
<td>Beijing</td>
<td>4000</td>
</tr>
<tr>
<td>Apple</td>
<td>iPhone6SP</td>
<td>Beijing</td>
<td>5000</td>
</tr>
<tr>
<td>Samsung</td>
<td>Galaxy S7</td>
<td>Beijing</td>
<td>3500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Loc</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple 6S 4.7’</td>
<td>Bei Jing</td>
<td>40K</td>
</tr>
<tr>
<td>Apple 6S 5.5’</td>
<td>Bei Jing</td>
<td>30K</td>
</tr>
<tr>
<td>Samsung S7</td>
<td>Bei Jing</td>
<td>35K</td>
</tr>
</tbody>
</table>

Data are full of errors and inconsistencies.
**Hybrid Human-Machine EM**

### DIMA System
- Similarity-based processing system
- Entity Matching
- Ease to use

### CDB System
- Crowd-powered SQL
- New optimization Model
- Entity Matching
- Cost, Latency, Quality

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Model</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>6S</td>
<td>iPhone</td>
<td>5.7</td>
</tr>
<tr>
<td>S7</td>
<td>Samsung</td>
<td>5.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Type</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone6</td>
<td>6th</td>
<td>red</td>
</tr>
<tr>
<td>Samsung</td>
<td>7gen</td>
<td>white</td>
</tr>
</tbody>
</table>
Hybrid Human-Machine EM

Two tables of records

Machine-Based Algorithms

Pruning dissimilar pairs

Candidate record pairs

Crowd-Based Algorithms

Asking crowd to label some pairs

Matching record pairs

<table>
<thead>
<tr>
<th>iPhone6S</th>
<th>iPhone 6S 4.7’</th>
<th>0.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone6SP</td>
<td>iPhone 6S 5.5’</td>
<td>0.75</td>
</tr>
<tr>
<td>Galaxy S7</td>
<td>Samsung S7</td>
<td>0.5</td>
</tr>
<tr>
<td>iPhone 6S</td>
<td>iPhone 6S 4.7’</td>
<td>0.72</td>
</tr>
<tr>
<td>iPhone 6S</td>
<td>iPhone 6S 5.5’</td>
<td>0.72</td>
</tr>
<tr>
<td>iPhone 6SP</td>
<td>iPhone 6S 4.7’</td>
<td>0.72</td>
</tr>
<tr>
<td>Galaxy S7</td>
<td>Samsung S7</td>
<td>0.5</td>
</tr>
<tr>
<td>iPhone6S</td>
<td>Samsung S7</td>
<td>0.1</td>
</tr>
<tr>
<td>iPhone6S</td>
<td>Samsung S7</td>
<td>0.1</td>
</tr>
<tr>
<td>Galaxy S7</td>
<td>iPhone 6S 4.7’</td>
<td>0.1</td>
</tr>
<tr>
<td>Galaxy S7</td>
<td>iPhone 6S 5.5’</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Pruning 4 dissimilar pairs

Jaccard
Edit distance
Semantics

Removing 2 non-matched pairs
Dima: Distributed In-memory Similarity-based System

Query interface
- Extended SQL
- Easy to use

Distributed In-memory Processing Engine
- Indexing
- Similarity Operations
- Optimizer

Support similarity-based query processing

Ji Sun, Zeyuan Shang, Guoliang Li, Dong Deng, Zhifeng Bao. Dima: A Distributed In-Memory Similarity-Based Query Processing System. VLDB 2017
CDB: A Crowd-powered Database

Fine-grained Tuple-level Graph model

Multi-goal optimization

EaseCrowd

ChinaCrowd

Guoliang Li, Chengliang Chai, Ju Fan, Jian Li, Yudian Zheng. CDB: A Crowd-Powered Database. SIGMOD 2017.
**DIMA: Rule-Based Matching**

<table>
<thead>
<tr>
<th>Name</th>
<th>Storage</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple 6S 4.7’</td>
<td>64GB</td>
<td>40K</td>
</tr>
<tr>
<td>Apple 6S 5.5’</td>
<td>128GB</td>
<td>30K</td>
</tr>
<tr>
<td>Samsung S7</td>
<td>64GB</td>
<td>35K</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Brand</th>
<th>Capacity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple iPhone6S</td>
<td>64</td>
<td>4000</td>
</tr>
<tr>
<td>Apple iPhone6SP</td>
<td>128G</td>
<td>5000</td>
</tr>
<tr>
<td>Samsung Galaxy S7</td>
<td>64G</td>
<td>3500</td>
</tr>
</tbody>
</table>

Jaccard(Name, Brand) ≥ 0.8 ∧ ED(Storage, Capacity) < 2

DIMA: Rule-Based Matching

□ Challenges

– How to obtain the rules?
  • High-quality rules
  • Explainable, Programmable

– How to apply the rules?
  • Avoid Cartesian product
  • Fast and Scalable
Quantifying Rules

$M$: positive examples  \hspace{1cm}  $N$: negative examples

$M\varphi$: record pairs that satisfy a rule $\varphi$

$M_\Psi$: record pairs that satisfy a rule set $\Psi=\{\varphi\}$

Objective function: $|M_\Psi \cap M| - |M_\Psi \cap N|$

Goal: Find a rule set $\Psi$ to maximize $|M_\Psi \cap M| - |M_\Psi \cap N|$
Given two entities, we map them to tree nodes while ignoring the semantics, which is important to the entity categorization problem. To this purpose, we propose to provide four similarity functions:

1. **Set-based**. The similarity between two values on attribute 
2. **Character-based**. They have rather small string similarity. However, they are similar.
3. **Ontology-based**. We map the entities to the ontology for venues of publications provided by Google Scholar Metric.
4. **Ontology Similarity**. Consider the tree structure of venues in Figure 4, and two nodes of the ontology for venues of publications provided by Google Scholar Metric. Given two entities, we map them to tree nodes while ignoring the semantics, which is important to the entity categorization problem. To this purpose, we propose to provide four similarity functions:

   - **Set-based**. The similarity between two values on attribute 
   - **Character-based**. They have rather small string similarity. However, they are similar.
   - **Ontology-based**. We map the entities to the ontology for venues of publications provided by Google Scholar Metric.
   - **Ontology Similarity**. Consider the tree structure of venues in Figure 4, and two nodes of the ontology for venues of publications provided by Google Scholar Metric.

The above similarity functions have a common limitation. Thus the overall complexity is

\[
O(|p| + |q|) \text{ for each } (p, q) \text{ such that } \delta(p, q) \geq \delta
\]

where \(\delta\) is the threshold.

They are effective.

\[
\text{Infinitene} \rightarrow \text{Finite}
\]

\[
\checkmark \quad F(a, b) \text{ on attribute pairs}
\]

**NP-hard Effective Algorithm**

\[
\delta: [0, 1]
\]
Rule-Based Matching

**Challenges**

- How to obtain the rules?
  - High-quality rules
  - Explainable, Programmable

- How to apply the rules?
  - Avoid Cartesian product
  - Fast and Scalable
Applying Rules

A Rule: Name is similar to Brand

It is expensive to enumerate every record pair!
✓ 10million*10million

Signature-based Method
✓ If two records do not share a common signature, they cannot be matching

Dima - Signature

Signature-based Method

✓ If two records do not share a common signature, they cannot be matching

Balance-Aware Signature

✓ The signatures are selectable
✓ Balance the workload

Dong Deng, Guoliang Li, He Wen, Jianhua Feng. An Efficient Partition Based Method for Exact Set Similarity Joins. VLDB, 2016
Dima: Load Balance

Challenges

- How to **generate signatures?**
  - Partition-based

- How to **select the signatures?**
  - Dynamic programming

- How to **balance the workload?**
  - NP-hard
  - Greedy algorithms

\[
\begin{align*}
\mathcal{W}_j &= \sum_{i=1}^{\eta_i} \left( b_i \sum_{g \in \text{pSig}_+^{+i,j}, \mathcal{P}(g) = j} \mathcal{F}^-[g] \right. \\
&\left. + c_i \sum_{g \in \text{pSig}_-^{+i,j}, \mathcal{P}(g) = j} \left( \mathcal{F}^+[g] + \sum_{g \in \text{pSig}_+^{+i,j}, \mathcal{P}(g) = j} \mathcal{F}^-[g] + \mathcal{F}^+[g] \right) \right) \\
b_i &= \begin{cases} 
1 & \text{Z}[i] = 1 \\
0 & \text{Z}[i] \neq 1
\end{cases} \\
c_i &= \begin{cases} 
1 & \text{Z}[i] = 2 \\
0 & \text{Z}[i] \neq 2
\end{cases} \\
s.t. \sum_{i=1}^{\eta_i} \text{Z}[i] \geq \theta_{|s|,i}.
\end{align*}
\]
Dima: Indexing

Signature-based partition
✓ Local join
✓ Avoid join on different nodes

Global Indexing
✓ Signature → nodes

Local Indexing
✓ Signature → records
Dima: Query Processing

**EM Operation**
- ✓ Selection
- ✓ Join
- ✓ Topk

**EM Query Processing**
- ✓ Global
  - ✓ ZipPartition; Balance-aware
- ✓ Local
  - ✓ Avoid Duplicates

---

IndexRDD

<table>
<thead>
<tr>
<th>Partition₀</th>
<th>ZipPartition</th>
<th>ProbeRDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>L⁺xJ⁺;L⁻xJ⁻</td>
<td>L⁺xJ⁺;L⁻xJ⁻</td>
<td>verify</td>
</tr>
<tr>
<td>L⁺xJ⁺;L⁻xJ⁻</td>
<td>L⁺xJ⁺;L⁻xJ⁻</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Partitionₙ</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FrequencyTable

- F⁺(sig), F⁻(sig)

Global Mapping

- P(sig)

DataRDD

- P₀: record
- P₁: record
- Pₙ: record
Dima: Query Optimization

\[ \varphi = \land \lambda \text{ where } \lambda: F(a,b) \geq \delta \]

EM Optimizer for multiple attributes
- Join order
- Selection Order
- Cost Estimation
- Size Estimation
- #Partitions

Guoliang Li, Jian He, Deng Dong, Jian Li, Jiahua Feng. Efficient Similarity Search and Join on Multi-Attribute Data. SIGMOD 2015.
Hybrid Human-Machine EM

DIMA System
- Similarity-based processing system
- Entity Matching
- Ease to use

CDB System
- Crowd-powered SQL
- New optimization Model
- Entity Matching
- Cost, Latency, Quality

<table>
<thead>
<tr>
<th>Name</th>
<th>Model</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>6S</td>
<td>iPhone</td>
<td>5.7</td>
</tr>
<tr>
<td>S7</td>
<td>Samsung</td>
<td>5.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Type</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>iPhone6</td>
<td>6th</td>
<td>red</td>
</tr>
<tr>
<td>Samsung</td>
<td>7gen</td>
<td>white</td>
</tr>
</tbody>
</table>
CDB: Selection-Inference-Refine

Next round

Challenge 1: How to select? Cost/Latency
Question Selection

Challenge 2: How to infer? Cost
Result Inference

Challenge 3: How to tolerate errors? Quality
Answer Refinement

Candidate questions
Selected questions
Crowd answers
Deduced answers
Refined answers

Next round
Challenges

- Labeling order
  - A\(=\)B, B\(\neq\)C \(\rightarrow\) A\(\neq\)C
  - B\(\neq\)C, A\(\neq\)C \(\rightarrow\) A\(?\)B

- Cost
  - Optimal Order

- Latency
  - Parallel crowdsourcing

- Quality
  - Transitivity Errors
  - If workers give B\(=\)C, then deduce A\(=\)C
Inference - Partial Order

- Candidate pairs $p_{ij}$
- Partial order
  
  $- p_{ij} > p_{ij'}$ if $s_{ij}^k \geq s_{ij'}^k$

Selection-Inference-Refine Framework

– Selection: minimize the number of questions
– Inference: infer the answers of no-asked questions
– Refine: tolerate errors of partial order and crowd
Question Selection

Serial Algorithms: Ask one question in each iteration

- Comparable Vertices
  - $O(\log |P|)$, $|P|$ is length of path

- Incomparable Vertices
  - $O(B \log |V|)$, $B$ is path number
  - $|V|$ is vertex number

Diagram of graph with vertices labeled $g_1, g_2, g_3, \ldots, g_9$ and edges connecting them.
Question Selection

Parallel Algorithm
Select multiple vertices and ask them together in each iteration

Multi-Path Algorithm

Topology-Sorting-Based Algorithm

Select multiple vertices and ask them together in each iteration.
Refinement

Overall weighted similarities of pairs

<table>
<thead>
<tr>
<th>$p_{ij}$</th>
<th>$\hat{s}_{ij}$</th>
<th>$p_{ij}$</th>
<th>$\hat{s}_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{12}$</td>
<td>0.72</td>
<td>$p_{37}$</td>
<td>0.21</td>
</tr>
<tr>
<td>$p_{13}$</td>
<td>0.68</td>
<td>$p_{45}$</td>
<td>0.97</td>
</tr>
<tr>
<td>$p_{23}$</td>
<td>0.60</td>
<td>$p_{46}$</td>
<td>0.43</td>
</tr>
<tr>
<td>$p_{24}$</td>
<td>0.28</td>
<td>$p_{47}$</td>
<td>0.42</td>
</tr>
<tr>
<td>$p_{25}$</td>
<td>0.29</td>
<td>$p_{56}$</td>
<td>0.41</td>
</tr>
<tr>
<td>$p_{26}$</td>
<td>0.40</td>
<td>$p_{57}$</td>
<td>0.44</td>
</tr>
<tr>
<td>$p_{27}$</td>
<td>0.41</td>
<td>$p_{67}$</td>
<td>0.98</td>
</tr>
<tr>
<td>$p_{34}$</td>
<td>0.39</td>
<td>$p_{89}$</td>
<td>0.37</td>
</tr>
<tr>
<td>$p_{35}$</td>
<td>0.39</td>
<td>$p_{10,11}$</td>
<td>0.44</td>
</tr>
</tbody>
</table>

Equi-depth histograms

Equation:

$$\omega_k = \frac{\sum_{p_{ij} \in P g} s_{ij}^k}{\sum_{p_{ij} \in P g} \sum_{1 \leq t \leq m} s_{ij}^t} \cdot \hat{s}_{ij} = \sum_{k \in [1, m]} \omega_k \cdot s_{ij}^k$$
Results

Cost $100 \times$

Latency $10 \times$

Quality $5\%$
Crowd-based Method - CDB

A crowd-powered database system
- Users require to write code to utilize crowdsourcing platforms
  - CDB encapsulates the complexities of interacting with the crowd

Limitations of existing systems
- Coarse-grained optimization - Tree model – Table Level
- Single-goal optimization - Cost

Highlights of CDB
- Fine-grained optimization - Graph Model – Tuple Level
- Multi-goal optimization – Cost, Latency, Quality
(1) Fine-Grained Query Model.

We develop a crowd-powered database system that enables the multi-goal optimization. Our system supports four types of tasks: (i) Collection task: it asks the crowd to collect new information, e.g., the affiliation of a professor. (ii) Single-choice task: it asks the crowd to select a single answer from multiple choices, e.g., selecting the country of a university from 100 given countries. (iii) Multiple-choice task: it asks the crowd to select multiple answers from multiple choices, e.g., selecting the country of a university from 100 given countries. (iv) Truth task: it asks the crowd to verify the truth of a statement, e.g., verifying the affiliation of a professor. We utilize relational tables to maintain the metadata. (1) Task. We utilize relational tables to maintain the metadata. (2) Query Optimization.

We propose a graph-based query model that supports the tuple-level optimization. We formally define the graph model and enable the online task selection. Based on the join/selection predicates in the query, we construct a graph model to find connected paths with 3 solid edges. This graph model connects the query and the data (e.g., Figure 1). This graph model enables the multi-goal optimization.

To address these limitations, we have developed a crowd-powered system to interact with crowdsourcing platforms. We have deployed our system on top of AMT. Another goal is to automatically publish the tasks to crowdsourcing platforms. We have deployed our system on AMT.

(2) Multi-Goal Optimization in One Framework.

We propose a holistic framework for task assignment and define a declarative query language that supports the multi-goal optimization. (i) For cost control, our goal is to minimize the number of tasks to find all the answers. For example, our method targets at selecting the most "beneficial" tasks which can be used to prune other tasks. (ii) For latency control, we adopt the round-lem is NP-hard and propose an expectation-based method to select tasks. (iii) For quality control, we aim to reduce the number of rounds for truth inference. Thus it calls for a new crowd-powered system to enable the multi-goal optimization. (4) We have implemented and deployed our system on AMT.

For example, the three tasks, and ask such tasks in parallel to reduce the latency.

To summarize, we make the following contributions.

1. We introduce a unified multi-goal optimization framework that supports the cost, latency and quality (Section 5).

2. We propose a graph-based query model that supports the tuple-level optimization. We formally define the graph model and enable the online task selection.

3. We introduce a declarative query language that supports the multi-goal optimization at the same time.

4. We have implemented and deployed our system on AMT.

The results demonstrate the performance superiority of both simulated and real experiments, and the experimental results of Section 6.
Tree Model vs Graph Model

- Find connected paths with 3 solid edges

Optimal Tree Model: 9+5+1=15 tasks
Graph Model: 3 tasks
Tree Model vs Graph Model

- Find connected paths with 3 solid edges

Optimal Tree Model: 9+5+1=15 tasks
Graph Model: 3 tasks
Table 1: Four Relational Tables (The Attribute Pairs That Can Be Joined Are Highlighted).

SELECT * FROM Paper, Researcher, Citation, University
WHERE Paper.Author CROWDEQUAL Researcher.Name AND
Paper.Title CROWDEQUAL Citation.Title AND
Researcher.Affiliation CROWDEQUAL University.Name

C DB: Graph Model

- Vertices
  - Tuples
- Edges
  - Join predicate
- Weight
  - Similarity

CDB: Cost Control

- Minimize Cost
  - Find all the results with the minimal cost

- Budget (pay as you go)
  - Find the most results with a given budget (B tasks)

- Expectation-based Method
  - Pruning ability to cut the graph

\[ E(t, t') = \frac{\prod_{i=1}^{x}(1 - \omega(t, t_i))}{x} \alpha + \frac{\prod_{i=1}^{y}(1 - \omega(t_i, t'))}{y} \beta \]
CDB: Latency Control

- Which tasks can be asked in parallel
  - Tasks have correlations
  - Tradeoff: cost and latency
  - Minimize the number of rounds without increasing the cost

- Task batching
  - Connected components
  - Edges containing tuples from the same table
CDB: Quality Control

Truth Inference
- A unified inference model
  • Single choice
  • Multiple choice
  • Fill/Collection

Online Task Assignment
- Worker Model
- Task Model
- Worker answer prediction

\[ p_i = \frac{\prod_{(w,a) \in V_t} (q_w)^{\mathbb{1}_{i=a}} \cdot \left(\frac{1-q_w}{\ell-1}\right)^{\mathbb{1}_{i \neq a}}}{\sum_{j=1}^{\ell} \prod_{(w,a) \in V_t} (q_w)^{\mathbb{1}_{j=a}} \cdot \left(\frac{1-q_w}{\ell-1}\right)^{\mathbb{1}_{j \neq a}}} \]

\[ C(t) = \sum_{\{(w,a) \in V_t\} \land \{(w',a') \in V_t\} \land \{w \neq w'\}} \frac{\text{sim}(a,a')}{(|V_t| \choose 2)} \]

\[ I(t) = \mathcal{H}(\tilde{p}) - \sum_{i=1}^{\ell} \left[ p_i \cdot q_w + (1 - p_i) \cdot \frac{1 - q_w}{\ell - 1} \right] \cdot \mathcal{H}(\tilde{p}) \]
Data Integration As A Service

User Interface
- Declarative Language
- API
- Workflow Management

Data Extraction
- Data Cleaning
- Entity Matching
- Schema Matching

Hybrid Machine-Human Computing
- Rule-based Reasoning
- Machine Learning

Crowd As A Service
- Crowd Modeling
- Crowd Scheduling
- Crowd Control

Data
Lessons Learned

- Human is important in data integration
- Machine step
  - Rules are important
  - Require high-quality examples
- Crowd step
  - Crowd is double-edged sword
    - high quality for easy tasks
    - low quality for hard tasks
  - Inference is important
    - Can reduce the cost significantly
    - But may sacrifice quality
All the codes are open-sourced at https://github.com/TsinghuaDatabaseGroup/

Thank You!