Group Linkage

Byung-Won On, Penn State University, USA
Nick Koudas, University of Toronto, Canada
Dongwon Lee, Penn State University, USA
Divesh Srivastava, AT&T Labs, USA

ICDE 2007

Motivation

- Data quality problem is increasing in DB applications
  - Dedicated venues: IQIS, CleanDB, IQ
- Reasons
  - Transcription errors
  - Lack of standards for recording fields
  - Errors due to poor design: eg, update anomalies, missing key constraint
Record Linkage

- Determining if two (record) entities are similar
- Eg
  - Address in CRM
    - #1: Dongwon Lee, 110 E. Foster Ave. #410, State College, PA, 16802
    - #2: LEE Dong, 110 East Foster Avenue Apartment 410, University Park, PA 16802-2343
  - Citation in Digital Library
    - #2: [SM83] G. Salton et al. 1983

Landscape

- Abundant research in many disciplines
- A.K.A.
  - DB: approximate join, merge/purge, record linkage
  - DL: citation matching, author name disambiguation
  - AI: identity matching
  - NLP: word sense disambiguation
  - IR: web query results clustering
  - LIS: name authority control
Group Linkage

- Often, “entity” is represented as a group of relational records (sharing a group ID).
- Eg,
  - An author with a group of publication records
  - A household in a census survey with a group of family members
  - An image with a group of sub-images in a grid

**Group Linkage Problem**: to determine if two entities represented as groups are approximately the same or not

---

Group Linkage Example

Collateral, 04
The Last Samurai, 03
Minority Report, 02
Vanilla Sky, 02

T. Cruise

V. F. Xu

Sofa-Jumping
Vanilla Sky
The Last Samurai
Mission Impossible
Mission Impossible 2

TX204
PPQ03
Popular Group Similarity

- Jaccard
  - Intuitive, cheap to run
  - Error-prone
  \[
sim(g_1, g_2) = \frac{|g_1 \cap g_2|}{|g_1 \cup g_2|}
\]

Q: Can we combine Jaccard and Bipartite Matching for Group Linkage?

- Bipartite Matching
  - Cardinality
  - Weighted
  - Rich
  - Expensive to run

Intuition for Better Similarity

- Two groups are similar if:
  - A large fraction of elements in the two groups form matching element pairs
  - There is high enough similarity between matching pairs of individual elements that constitute the two groups
Group Similarity

- Two groups of elements:
  - $g_1 = \{r_{11}, r_{12}, \ldots, r_{1m_1}\}$, $g_2 = \{r_{21}, r_{22}, \ldots, r_{2m_2}\}$
  - The group measure $BM$ is the normalized weight of the maximum bipartite matching $M$ in the bipartite graph $(N = g_1 \cup g_2, E = g_1 \times g_2)$

$$BM_{\text{sim}, \rho}(g_1, g_2) = \frac{\sum_{(r_{i1}, r_{i2}) \in M} (\text{sim}(r_{i1}, r_{i2}))}{m_1 + m_2 - |M|}$$

such that $\text{sim}(r_{i1}, r_{i2}) \geq \rho$
- $BM(g_1, g_2) \geq \theta$

User-set Parameters

Example ($\rho = 0.3, \Theta = 0.9$)

$BM_{\text{sim}, \rho}(g_1, g_2) = \frac{0.9 + 0.7}{3 + 2 - 2} = \frac{1.6}{3} = 0.53 < \Theta$

Therefore, $g_1 \not\sim g_2$!
**Challenge**

- **Each BM group measure uses the maximum weight bipartite matching**
  - Bellman-Ford: $O(V^2E)$
  - Hungarian: $O(V^3)$
- **Large number of groups to match**
  - $O(NM)$

**Solution: Greedy matching**

- Bipartite matching computation is expensive because of the requirement
  - No node in the bipartite graph can have more than one edge incident on it
- Let’s relax this constraint:
  - For each element $e_i$ in $g_1$, find an element $e_j$ in $g_2$ with the highest element-level similarity $\equiv S_1$
  - For each element $e_j$ in $g_2$, find an element $e_i$ in $g_1$ with the highest element-level similarity $\equiv S_2$
Upper/Lower Bounds

\[ BM_{\text{sim}, \rho} (g_1, g_2) = \frac{\sum_{(r_i, r_j) \in M} \left( sim(r_i, r_j) \right)}{m_1 + m_2 - |M|} \]

\[ UB_{\text{sim}, \rho} (g_1, g_2) = \frac{\sum_{(r_i, r_j) \in S_1 \cup S_2} \left( sim(r_i, r_j) \right)}{m_1 + m_2 - |S_1 \cup S_2|} \]

\[ LB_{\text{sim}, \rho} (g_1, g_2) = \frac{\sum_{(r_i, r_j) \in S_1 \cap S_2} \left( sim(r_i, r_j) \right)}{m_1 + m_2 - |S_1 \cap S_2|} \]

- Properties:
  - Numerator of UB is at least as large as that of BM
  - Denominator of UB is no larger than that of BM
  - \[ \Rightarrow \] UB is the upper-bound of BM
Theorem & Algorithm

\[ BM_{\text{sim}, \rho}(g^1, g^2) \leq UB_{\text{sim}, \rho}(g^1, g^2) \]

Theorem 1

- IF \( UB(g^1, g^2) < \theta \rightarrow BM(g^1, g^2) < \theta \rightarrow g^1 \neq g^2 \)

\[ LB_{\text{sim}, \rho}(g^1, g^2) \leq BM_{\text{sim}, \rho}(g^1, g^2) \]

Theorem 2

- ELSE IF \( LB(g^1, g^2) \geq \theta \rightarrow BM(g^1, g^2) \geq \theta \rightarrow g^1 \approx g^2 \)
- ELSE, compute \( BM(g^1, g^2) \)

Goal:

\[ BM(g^1, g^2) \geq \theta \]

MAX Heuristics

\[ MAX_{\text{sim}, \rho}(g^1, g^2) = \max_{(r_{i}, r_{j}) \in g^1 \times g^2} sim(r_{i}, r_{j}) \]

- Two groups with high BM will share at least one pair of very similar elements
  - Use MAX to quickly identify those
  - No guarantee of avoiding false identification
- We proposed 4 group similarity measures:
  - BM, UB, LB, and MAX
Evaluation

- Evaluated Search version (vs. Join version)
- Use bibliography data set from ACM and DBLP digital libraries
  - Authors with his/her publication lists
- Various cases
  - Real vs. Synthetic
  - Uniform vs. Skewed
  - Jaccard vs. 4 proposals (BM, UB, LB, and MAX)
  - Hybrid as blocking method
- Main evaluation metric: AVG recall

BM vs. Jaccard

S1: Left: 300 DBLP groups
Right: 700,000 ACM groups + 1/3 or 3 dummy groups

Jaccard gets confused easily
**BM vs. Jaccard**

**S2:**
- Left: 100 ACM groups
- Right: Left + 100 erroneous groups (30%, 45%, 60%)

![Bar chart showing average recall comparison between Jaccard and BM methods with different error rates.]

**MAX vs. UB**

**R2Net:**
- Left: 100 DBLP groups on AI topics
- Right: 700,000 ACM groups

![Bar chart showing average recall comparison between MAX and UB methods with different values.]

ICDE 2007 / Group Linkage
ACM Dataset

**R2Net**: Left: 100 DBLP groups on AI topics  
Right: 700,000 ACM groups

![Graph](image)

UB(10)|BM(k)

Conclusion

- When entities have a group of elements in them, group linkage is useful and efficient
- Directions
  - More efficient implementation => Approximate Group Linkage
  - Hierarchical Group Linkage: OLAP
  - Group => Tree, Graph
  - Application to Image Retrieval