

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

#1: G. Salton and M. McGill, "Introduction to Modern Information Retrieval," McGraw-Hill, 1983

#2: [SM83] G. Salton et al. 1983

Landscape

- Abundant research in many disciplines

- A.K.A.

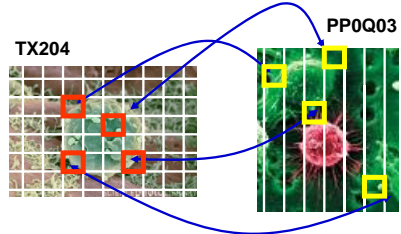
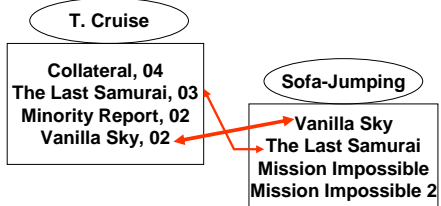
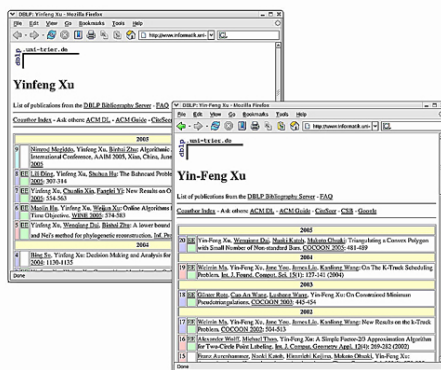
- DB: approximate join, merge/purge, record linkage
- DL: citation matching, author name disambiguation
- AI: identity matching
- NLP: word sense disambiguation
- IR: web query results clustering
- LIS: name authority control

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



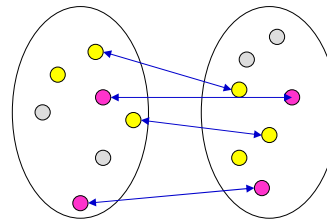
Popular Group Similarity

- Jaccard
- Intuitive, cheap to run
- Error-prone

$$\text{sim}(g_1, g_2) = \frac{|g_1 \cap g_2|}{|g_1 \cup g_2|}$$

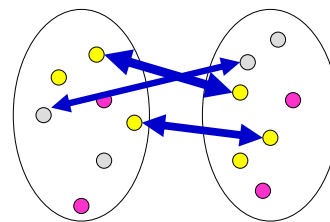
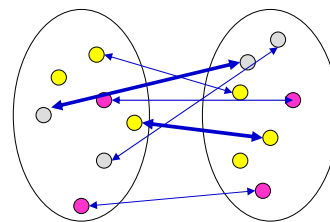
Q: Can we combine Jaccard and Bipartite Matching for Group Linkage?

- 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

$$sim(g_1, g_2) = \frac{|g_1 \cap g_2|}{|g_1 \cup g_2|}$$

- Two groups of elements:
 - $g_1 = \{r_{11}, r_{12}, \dots, r_{1m1}\}$, $g_2 = \{r_{21}, r_{22}, \dots, r_{2m2}\}$
 - The group measure **BM** is the normalized weight of the maximum bipartite matching M in the bipartite graph ($N = g_1 \cup g_2$, $E = g_1 \times g_2$)

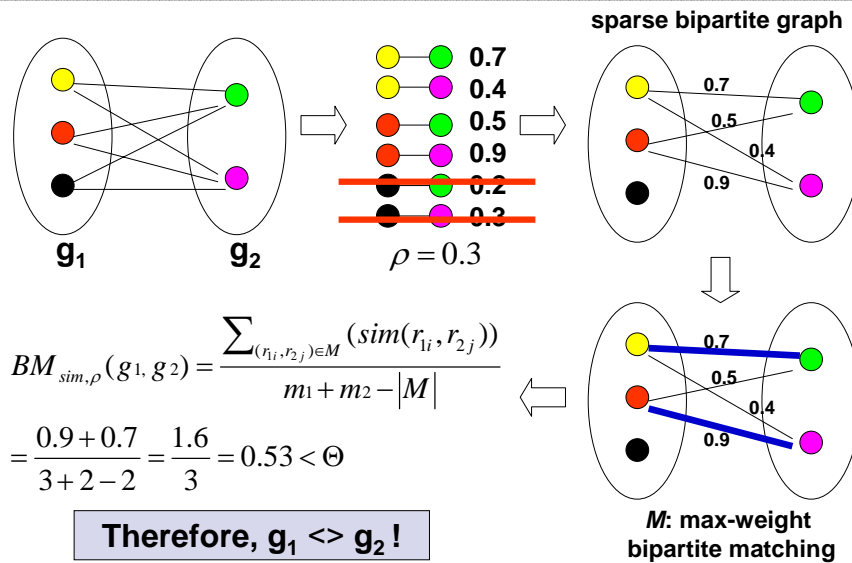
$$BM_{sim,\rho}(g_1, g_2) = \frac{\sum_{(r_{1i}, r_{2j}) \in M} (sim(r_{1i}, r_{2j}))}{m_1 + m_2 - |M|}$$

such that $sim(r_{1i}, r_{2j}) \geq \rho$

- $BM(g_1, g_2) \geq \theta$

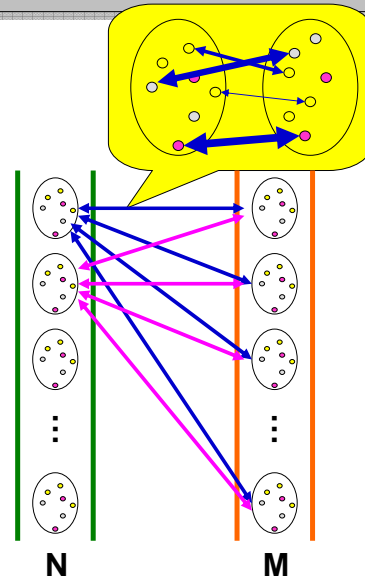
User-set Parameters

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



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
 - $O(NM)$



ICDE 2007 / Group Linkage

11

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 $\Leftrightarrow S_1$
 - For each element e_j in g_2 , find an element e_i in g_1 with the **highest** element-level similarity $\Leftrightarrow S_2$

ICDE 2007 / Group Linkage

12

Upper/Lower Bounds

$$BM_{sim,\rho}(g_1, g_2) = \frac{\sum_{(r_i, r_j) \in M} (sim(r_i, r_j))}{m_1 + m_2 - |M|}$$

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

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

Upper/Lower Bounds

$$BM_{sim,\rho}(g_1, g_2) = \frac{\sum_{(r_i, r_j) \in M} (sim(r_i, r_j))}{m_1 + m_2 - |M|}$$

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

- Properties:
 - Numerator of UB is **at least as large as** that of BM
 - Denominator of UB is **no larger than** that of BM
- => UB is the upper-bound of BM

Theorem & Algorithm

$$BM_{sim,\rho}(g_1, g_2) \leq UB_{sim,\rho}(g_1, g_2) \quad \text{Theorem 1}$$

- **IF** $UB(g_1, g_2) < \theta \rightarrow BM(g_1, g_2) < \theta \rightarrow g_1 \neq g_2$

$$LB_{sim,\rho}(g_1, g_2) \leq BM_{sim,\rho}(g_1, g_2) \quad \text{Theorem 2}$$

- **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

$$MAX_{sim,\rho}(g_1, g_2) = \max_{(r_{1i}, r_{2j}) \in g_1 \times g_2} sim(r_{1i}, r_{2j})$$

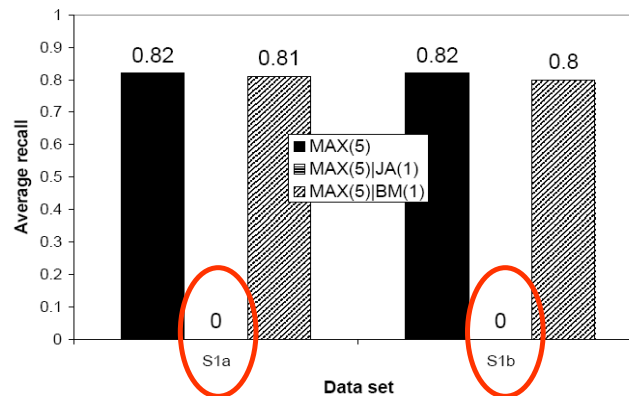
- Two groups with high BM will share at least one pair of very similar elements
 - Use MAX to quickly identify those
 - No guarantee of avoiding false identification
- We proposed 4 group similarity measures:
 - **BM**, **UB**, **LB**, and **MAX**

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

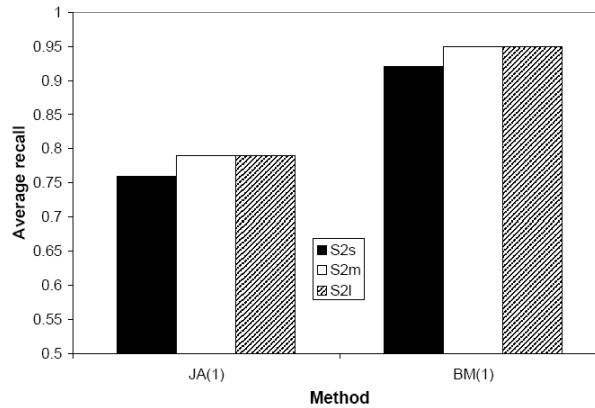
S1: Left: 300 DBLP groups
Right: 700,000 ACM groups + 1/3 or 3 dummy groups



Jaccard gets confused easily

BM vs. Jaccard

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

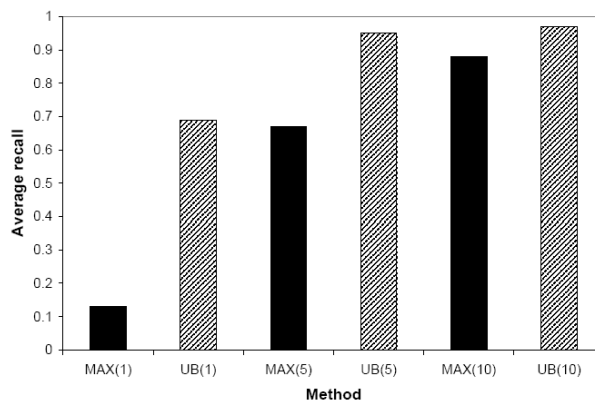


ICDE 2007 / Group Linkage

19

MAX vs. UB

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

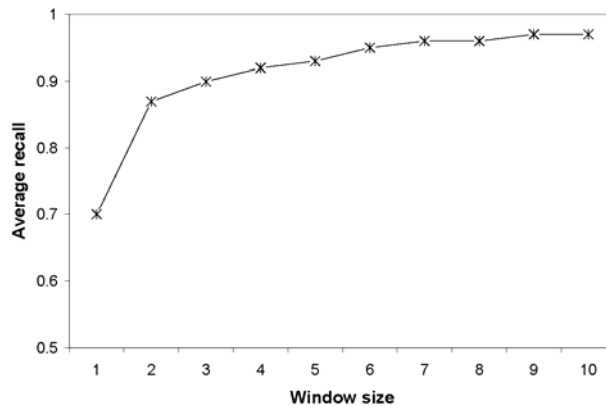


ICDE 2007 / Group Linkage

20

ACM Dataset

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



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