Discovering Complex Matchings across Web Query Interfaces: A Correlation Mining Approach

Bin He, Kevin Chen-Chuan Chang, Jiawei Han

Presented by: Maryam Karimzadehgan
mkarimzadehgan@cs.uwaterloo.ca

Outline

- Motivation of integrating the deep web
- Mining Algorithm
- H-Measure
- Data Preparation Step
- Experiments
- Conclusion and Discussion

2

Motivation: Matching → Mining

- Group attributes

  Larger concept e.g. { adults, seniors, children, infants} denotes the number of passengers.
- Synonym relationship

  Different sources may use different attributes for the same concept. e.g. {from} = {depart}, {to} = {destination}

Group attribute + Synonym attribute → Complex Matching

m:n → m:n matching

e.g. {adults, seniors, children, infants} = {passengers} 4:1 matching

3

Correlation Mining

- Match more than two attribute groups

{adults, seniors, children, infants} = { passengers} = { number of tickets}
4:1:1

n-ary complex matching → aggregation of several binary m:n mathcings

- DCM (Dual Correlation Mining) framework for mining *n-ary* complex mathchings

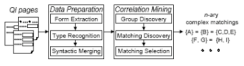
4

Outline

- Introduction
- Motivation of integrating the deep web
- Mining Algorithm
- H-Measure
- Data Preparation Step
- Experiments
- Discussion

5

DCM Framework (Dual Correlation Mining)



Data Preparation:

schema transaction →mining by extracting the attribute entities from query interfaces

Correlation Mining:

finds complex matchings considering both positive and negative correlations

6

Mining Algorithm

- **Group discovery (individual attribute)**
  - Mining positively correlated attribute -- > potential attribute group
  - Potential group is not suitable for matching -- > synonym relationship (negative correlation)
- **Matching discovery (attribute groups)**
  - Given the potential group -- > Mining negatively correlated attribute -- > potential *n-ary* complex matching
- **Matching selection**
  - A potential matching may not be considered as correct due to existence of conflicts among matchings.
  - Select the consistent matchings from the mining result

7

Complex Matching Discovery

- Negative correlation should exists between two groups. (e.g. {destination} = {to} = {arrival city})

  $C_{\min}$ is the minimal value of the pairwise evaluation

  - $C_{\min}$ satisfies the "Apriori" feature. e.g. $C_{\min}(A,m) \le C_{\min}(A^*,m)$
    min {1,3,5} <= min {3,5}
  - $C_{\min}$ can mine both positive and negative correlations.
- A set of attributes $\{A_1,\ldots,A_n\}$ is positively correlated if

  $$C_{\min} \ge Threshold$$
- A set of attributes groups $\{G_1,\ldots,G_n\}$ is negatively correlated attribute groups if

  $$C_{\min} \ge Threshold$$

8

Algorithm N-ary (Discovering Complex Matching)

Algorithm: N-ARYSCHEMAMATCHING:
Input: InputSchemas $\mathcal{S}_I = \{S_1,\ldots,S_n\}$,
Measures m_p, m_n, Thresholds T_p, T_n
Output: Potential n-ary complex matchings
begin:
1 /* group discovery */
2 $\mathcal{G} \leftarrow$ APRIORICORRMINING($\mathcal{S}_I, m_p, T_p$)
3 /* adding groups into $\mathcal{S}_I$ */
4 **for** each $S_i \in \mathcal{S}_I$
5 **for** each $G_k \in \mathcal{G}$
6 **if** $S_i \cap G_k \neq \emptyset$ **then** $S_i \leftarrow S_i \cup \{G_k\}$
7 /* matching discovery */
8 $\mathcal{M} \leftarrow$ APRIORICORRMINING($\mathcal{S}_I, m_n, T_n$)
9 **return** $\mathcal{M}$
end

Algorithm: APRIORICORRMINING:
Input: InputSchemas $\mathcal{S}_I = \{S_1,\ldots,S_n\}$,
Measures m, Thresholds T
Output: Correlated items
begin:
1 $X \leftarrow \emptyset$
2 $\mathcal{V} \leftarrow \bigcup_{i=1}^{n} S_i, S_i \in \mathcal{S}_I$
3 **for** all $A_p, A_q \in \mathcal{V}, p \neq q$
4 **if** $m(A_p, A_q) \ge T$ **then** $X \leftarrow X \cup \{\{A_p, A_q\}\}$
5 $l \leftarrow 2$
6 /* X_l: correlated stems with length = l */
7 $X_l \leftarrow X$
8 **while** $X_l \neq \emptyset$
9 construct X_{l+1} from X_l using apriori feature
10 $X \leftarrow X \cup X_{l+1}$
11 $X_l \leftarrow X_{l+1}$
12 **return** X
end

9

Complex Matching Selection

- False semantic matching due to coincidental correlations
  - {author} = {first name, last name} M1 correct one
  - {subject}= {first name, last name} M2 wrong one

False ones should be removed

The more negatively correlated -- > higher confidence to be real synonyms

- Strategy for *ranking* the discovered mathchings:
  Score function to evaluate the discovered matchings under measure

The goal of qualifying → correlation passes some threshold
The goal of ranking → compare the strength of correlations

$C_{\max}$ The maximal measure value among pairs of groups in a matching

10

Complex Matching Selection

- Strategy for tie breaking → richer semantic information
  - Take "top K" approach
  - If $C_{\max}$ value of two groups are the same → compare their second highest to break the tie.
  - If two matching are tie after " top-k" comparison → the one with richer semantic information.
- Rule for ranking matches

1) If $s(M_j, m_n) > s(M_k, m_n)$,
2) If $s(M_j, m_n) = s(M_k, m_n)$ and $M_j \succeq M_k$
3) rank them arbitrary

11

Complex Matching Selection

Algorithm: MATCHINGSELECTION:
Input: Potential complex matchings $\mathcal{M} = \{M_1, ..., M_v\}$,
Measure m_n
Output: Selected complex matchings
begin:
1 $\mathcal{R} \leftarrow \emptyset$ /* selected n-ary complex matchings */
2 **while** $\mathcal{M} \neq \emptyset$
3 /* select the matching ranked the highest */
4 $M_t \leftarrow$ GETMATCHINGRANKFIRST($\mathcal{M}, m_n$)
5 $\mathcal{R} \leftarrow \mathcal{R} \cup \{M_t\}$
6 **for** each $M_j \in \mathcal{M}$
7 /* remove the conflicting part */
8 $M_j \leftarrow M_j - M_t$
9 /* delete M_j if it contains no matching */
10 **if** $|M_j| < 2$ **then** $\mathcal{M} \leftarrow \mathcal{M} - \{M_j\}$
11 **return** $\mathcal{R}$
end

Algorithm: GETMATCHINGRANKFIRST:
Input: Potential complex matchings $\mathcal{M} = \{M_1, ..., M_v\}$,
Measure m_n
Output: The matching with the highest ranking
begin:
1 $M_t \leftarrow M_1$
2 **for** each $M_j \in \mathcal{M}, 2 \leq j \leq v$
3 **if** $s(M_j, m_n) > s(M_t, m_n)$ **then**
4 $M_t \leftarrow M_j$
5 **if** $s(M_j, m_n) = s(M_t, m_n)$ and $M_j \succeq M_t$ **then**
6 $M_t \leftarrow M_j$
7 **return** M_t
end

- Greedy selection strategy by choosing the highest ranked matching, M_t in each iteration.

Example

M_1: {author} = {last name, first name}, 0.95
M_2: {author} = {last name}, 0.95
M_3: {subject} = {category}, 0.92
M_4: {author} = {first name}, 0.90
M_5: {subject} = {last name, first name} , 0.88
M_6: {subject} = {last name}, 0.88 and
M_7: {subject} = {first name}, 0.86.

12

Outline

- Introduction
- Motivation of integrating the deep web
- Mining Algorithm
- H-Measure
- Data Preparation Step
- Experiments
- Discussion

13

Measures for Association Patterns

#	Measure	Formula
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_i P(A_i,B_j)+\sum_i \max_j P(A_i,B_j)-\max_i P(A_i)-\max_j P(B_j)}{2-\max_i P(A_i)-\max_j P(B_j)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{A},\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{A},\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\left(P(A,B)\log(\frac{P(B\mid A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\mid A)}{P(\overline{B})}),\right.$ $\left.P(A,B)\log(\frac{P(A\mid B)}{P(A)})+P(\overline{A}B)\log(\frac{P(\overline{A}\mid B)}{P(\overline{A})})\right)$
9	Gini index (G)	$\max\left(P(A)[P(B\mid A)^2+P(\overline{B}\mid A)^2]+P(\overline{A})[P(B\mid\overline{A})^2+P(\overline{B}\mid\overline{A})^2]\right.$ $-P(B)^2-P(\overline{B})^2,$ $\left.P(B)[P(A\mid B)^2+P(\overline{A}\mid B)^2]+P(\overline{B})[P(A\mid\overline{B})^2+P(\overline{A}\mid\overline{B})^2]\right.$ $\left.-P(A)^2-P(\overline{A})^2\right)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\mid A), P(A\mid B))$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	Sajne (PS)	$\sqrt{P(A)P(B)}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\mid A)-P(B)}{1-P(B)}, \frac{P(A\mid B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{A}\overline{B})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{A}\overline{B})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$

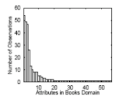
14

Correlation Measure

- Contingency table.

	A_p	$\neg A_p$	
A_q	f_{11}	f_{10}	f_{1+}
$\neg A_q$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	f_{++}

- Design of a correlation measure is empirical
- No good correlation measure
- Not uniform attribute frequency



15

Correlation Measure (Continued)

1. Non-uniform attributes → sparseness of schema data
2. Many attributes rarely used (occurring only once in the schemas → rare attribute problem
3. Highly frequently attributes → frequent attribute problem

The chosen measure should be robust against these three problems.

16

Sparseness problem

$$Lift = \frac{f_{00} f_{11}}{f_{10} f_{01}}$$

- Lift measure includes the count of co-absence
  - Not good measure for schema matching → exaggerate the effect of co-absence.

	A_p	$\neg A_p$	
A_q	5	5	10
$\neg A_q$	5	85	90
	10	90	100

(a1) Example of sparseness problem with measure *Lift*: Less positive correlation but a higher *Lift* = 17.

	A_p	$\neg A_p$	
A_q	55	20	75
$\neg A_q$	20	5	25
	75	25	100

(a2) Example of sparseness problem with measure *Lift*: More positive correlation but a lower *Lift* = 0.69.

Lift >1 → positive correlation

Lift<1 → negative correlation

f_{00} incorrectly affects the result → Lift suffers Sparseness problem

17

Rare attribute problem

Rare problem: either A_p or A_q is rarely observed →Measure should not consider A_p and A_q as highly negatively correlated.

$$Jaccard = \frac{f_{11}}{f_{11} + f_{10} + f_{01}}$$

	A_p	$\neg A_p$	
A_q	1	49	50
$\neg A_q$	1	1	2
	2	50	52

(b1) Example of rare attribute problem with measure *Jaccard*: A_p as rare attribute and *Jaccard* = 0.02.

	A_p	$\neg A_p$	
A_q	1	25	26
$\neg A_q$	25	1	26
	26	26	52

(b2) Example of rare attribute problem with measure *Jaccard*: no rare attribute and *Jaccard* = 0.02.

More Negatively correlated

18

H-Measure

$$m_n(A_p, A_q) = H(A_p, A_q) = \frac{f_{01} f_{10}}{f_{+1} f_{1+}}$$

- H-measure is as the negative correlation m_n
- H value close to 0 → highly degree of positive correlation
- H value to 1 → high degree of negative correlation
- H-measure avoids the sparseness problem by ignoring f_{00}

Frequent attribute problem

- The ability to differentiate highly frequent attributes from really correlated ones.
- Generating false positives in group discovery

	A_p	$\neg A_p$	
A_q	81	9	90
$\neg A_q$	9	1	10
	90	10	100

(c1) Example of frequent attribute problem with measure *Jaccard*: A_p and A_q are independent but a higher *Jaccard* = 0.82

	A_p	$\neg A_p$	
A_q	8	1	9
$\neg A_q$	1	90	91
	9	91	100

(c2) Example of frequent attribute problem with measure *Jaccard*: A_p and A_q are positively correlated but a lower *Jaccard* < 0.8

Outline

- Introduction
- Motivation of integrating the deep web
- Mining Algorithm
- H-Measure
- Data Preparation Step
- Experiments
- Discussion

Data Preparation

- **Form extraction**: extracting attributes entities (names and domains) from query interfaces
  - Standard Normalization: Stemming attribute names and domains
  - Normalizing irregular nouns and verbs e.g. children → child
  - Delete common stop words
- **Type recognition**
  - Problem of homonyms (the same name with different meanings) → distinguish both names and types
  - Type identification is not declared in web interfaces → type recognizer to recognize types from domain values of attributes entities
- **Syntactic Merging**
  - Name-bases merging: merging two attribute entities if they are similar in names
  - Domain-based merging: merging two attribute entities if they are similar in domain values.

22

Outline

- Introduction
- Motivation of integrating the deep web
- Mining Algorithm
- H-Measure
- Data Preparation Step
- Experiments
- Discussion

23

Experiments

- Two datasets:
  - TEL-8 dataset
    - Raw web pages over 447 deep web sources in 8 domains.
    - Each domain has about 20-70 sources
  - BAMM dataset
    - Manually extracted attribute names over 211 sources in 4 domains (around 50 sources per domain)

24

Experimental Result (result on TEL-8 Dataset)

Step	*Value of*	*Result*	C_{min}	C_{max}
group discovery	$\mathcal{U}$	G_1 = {last name (unknown), first name (any)}	0.94	
		G_2 = {title (any), keyword (any)}	0.93	
		G_3 = {last name (any), title (any)}	0.91	
		G_4 = {first name (any), catalog (any)}	0.90	
		G_5 = {first name (any), keyword (any)}	0.87	
matching discovery	$\mathcal{M}$	M_1: {author (any)} = {last name (any), first name (any)}	0.87	0.87
		M_2: {author (any)} = {last name (any)}	0.87	0.87
		M_3: {subject (string)} = {category (string)}	0.83	0.83
		M_4: {author (any)} = {last name (any), catalog (any)}	0.82	0.82
		M_5: {author (any)} = {first name (any)}	0.82	0.82
		M_6: {category (string)} = {publisher (string)}	0.76	0.76
matching selection	$\mathcal{R}$	R_1: {author (any)} = {last name (any), first name (any)}		0.87
		R_2: {subject (string)} = {category (string)}		0.83

Domain	*Final Output After Matching Selection*	*Correct?*
Airfares	{destination (string)} = {to (string)} = {arrival city (string)}	Y
	{departure date (datetime)} = {depart (datetime)}	Y
	{passenger (integer)} = {adult (integer), child (integer), infant (integer)}	P
	{from (string), to (string)} = {departure city (string), arrival city (string)}	Y
	{from (string)} = {depart (string)}	Y
	{return date (datetime)} = {return (datetime)}	Y
Movies	{artist (any)} = {actor (any)} = {star (any)}	Y
	{genre (string)} = {category (string)}	Y
	{cast & crew (any)} = {actor (any), director (any)}	Y

25

Experimental Result (result on TEL-8 Dataset)

- Attributes above a frequency threshold T

Domain	Pr (20%)	Re (20%)	Pr (10%)	Re (10%)
Books	1	1	1	1
Airfares	1	1	1	0.71
Movies	1	1	1	1
MusicRecords	1	1	0.76	1
Hotels	0.86	1	0.86	0.87
CarRentals	0.72	1	0.72	0.60
Jobs	1	0.86	0.78	0.87
Automobiles	1	1	0.93	1

- **Evaluating the H-measure**

Domain	$Pr(H)$ (10%)	$Re(H)$ (10%)	$Pr(\zeta)$ (10%)	$Re(\zeta)$ (10%)
Books	1	1	0.80	1
Airfares	1	0.71	0.79	0.61
Movies	1	1	0.93	1
MusicRecords	0.76	1	0.76	1
Hotels	0.86	0.87	0.44	0.95
CarRentals	0.72	0.60	0.68	0.62
Jobs	0.78	0.87	0.64	0.87
Automobiles	0.93	1	0.78	1

26

Conclusion

- Complex matching n-ary complex matching
- DCM framework
- Mining Algorithm
- H-measure

27

Discussion Points

- Their approach works for the same domain, How it can cross the domain boundary is an open problem.
- Choosing the threshold values is empirically, they did it by testing the algorithm with various threshold, how it can be designed more systematically?
- H-Measure were derived based on the observation of the data, it should be systematic. Does it work the same if the datasets change?
- Compare their H-measure with other measures not only one.
- In removing the conflicted groups, how do they remove the conflicted one if both groups have the same score value?

28

Questions ?

29