A two-step classification approach to unsupervised record linkage

Peter Christen

Department of Computer Science,
Faculty of Engineering and Information Technology,
ANU College of Engineering and Computer Science,
The Australian National University

Contact: **peter.christen@anu.edu.au**

Project Web site: **http://datamining.anu.edu.au/linkage.html**

Funded by the Australian National University, the NSW Department of Health, and the Australian Research Council (ARC) under Linkage Project 0453463.

Outline

- What is record linkage?
- Record linkage techniques
- The record linkage process
- Record pair comparison and classification
- Two-step record pair classification
  - Step 1: Training example selection
  - Step 2: Classification of record pairs
- Experimental results
- Outlook and future work

What is record (or data) linkage?

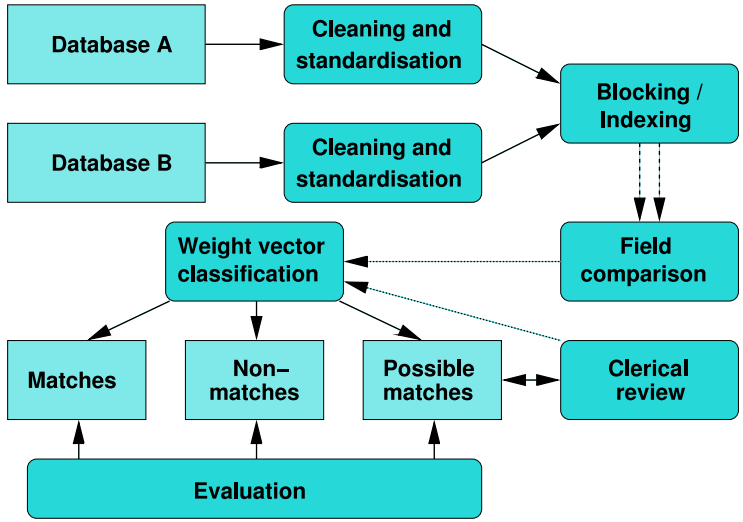
- The process of linking and aggregating records from one or more data sources representing the same entity (such as a patient, customer, or business)
- Also called *data matching*, *data integration*, *data scrubbing*, *entity resolution*, *object identification*, *merge-purge*, etc.
- Challenging if no unique entity identifiers available
  For example, which of these three records refer to the same person?

Dr Smith, Peter	*42 Miller Street 2602 O'Connor*
Pete Smith	*42 Miller St, 2600 Canberra A.C.T.*
P. Smithers	*24 Mill Street; Canberra ACT 2600*

Record linkage techniques

- Deterministic linkage
  - Exact linkage (if a *unique* entity identifier of high quality is available: has to be precise, robust, stable over time)
    Examples: *Medicare*, *ABN* or *Tax file* number (?)
  - Rules based linkage (complex to build and maintain)
- Probabilistic linkage *(Fellegi and Sunter, 1969)*
  Use available (personal) information for linkage (which can be missing, wrong, coded differently, and/or out-of-date)
  Examples: *names*, *addresses*, *dates of birth*, etc.
- Modern approaches
  Based on machine learning, data mining, artificial intelligence, and information retrieval techniques

The record linkage process



Record pair comparison

- Pairs of records are compared field (attribute) wise using different field comparison functions
  - Such as exact or approximate string (e.g. edit-distance, q-gram, Winkler), numeric, age, date, time, etc.
  - Return 1.0 for exact matches, 0.0 for total dissimilarity
- For each compared record pair a vector containing *matching weights* is calculated

```
Record 1:         ['dr', 'peter', 'paul', 'miller']
Record 2:         ['mr', 'john',  '',     'miller']
Matching weights: [0.5,  0.0,     0.0,    1.0      ]
```

- Record pairs (weight vectors) are classified into *matches*, *non-matches* (and *possible matches*)

Record pair classification

- Various machine learning techniques can be used
  - Supervised: SVM, decision trees, neural networks, learnable string comparisons, active learning, etc.
  - Un-supervised: Different *clustering* algorithms
  - Supervised techniques normally perform better
- However, in many cases there is no training data available (or only through expensive manual efforts)
- Recently, *collective* entity resolution techniques have been investigated
  - Rather than classifying each record pair independently
  - Using relational attributes (i.e. graph based)

Two-step record pair classification

- Assumptions
  - Weight vectors that have exact or high similarity values in all vector elements were most likely generated when two records were compared that refer to the same entity
  - Weight vectors with mostly low similarity values were with high likelihood generated when two records were compared that refer to different entities
- Idea: Automatically select such weight vectors as training data in a first step, and then use them to train a binary classifier in a second step
  - Combined, this will allow fully automated unsupervised record pair classification

Example records and their corresponding weight vectors

R1:	Christine	Smith	42	Main	Street
R2:	Christina	Smith	42	Main	St
R3:	Bob	O'Brian	11	Smith	Rd
R4:	Robert	Bryce	12	Smythe	Road

WV(R1,R2):	0.9	1.0	1.0	1.0	0.9
WV(R1,R3):	0.0	0.0	0.0	0.0	0.0
WV(R1,R4):	0.0	0.0	0.5	0.0	0.0
WV(R2,R3):	0.0	0.0	0.0	0.0	0.0
WV(R2,R4):	0.0	0.0	0.5	0.0	0.0
WV(R3,R4):	0.7	0.4	0.5	0.7	0.9

Step 1: Training example selection

- Weight vectors can be selected into training sets $\mathbf{W}_M$ and $\mathbf{W}_N$ in two different ways
  1. **Threshold based** Using two distance thresholds, one from the exact match value (1.0), the other from the total dissimilarity value (0.0)
     Parameters: Two thresholds: $0.0 < t_m, t_n < 1.0$
  2. **Nearest based** Select the weight vectors nearest to the exact match vector ([1.0, ..., 1.0]), and nearest to the total dissimilarity vector ([0.0, ..., 0.0])
     Parameters: Two numbers of nearest: $0 < x_m, x_n$; unique or non-unique weight vector selection; balanced or im-balanced selection ($|\mathbf{W}_M| < |\mathbf{W}_N|$)

Step 2: Classification of record pairs

- Any binary classifier can be used (in the following experiments, a linear SVM has been employed)
- One issue: the match and non-match training sets $\mathbf{W}_M$ and $\mathbf{W}_N$ are very likely linearly separable
- Related work: Similar approaches have been developed for text and Web page classification
  - Called *semi-supervised* or *partially supervised* learning
  - *PEBL* (positive example based learning): train a SVM only on positive labeled examples
  - *S-EM* (seed expectation-maximisation): add 'spy' documents from positive examples into unlabeled data

Experimental evaluation

- All techniques are implemented in the *Febrl* open source record linkage system
- Experiments using both real and synthetic data (*Secondstring* repository and *Febrl* data set generator)
- Evaluation of step 1 (training example selection)
  - Quality of example weight vectors selected into training sets $\mathbf{W}_M$ and $\mathbf{W}_N$
  - Measured as percentage of true matches in $\mathbf{W}_M$ and true non-matches in $\mathbf{W}_N$
- Evaluation of step 2 (record pair classification)
  - *F*-measure (harmonic mean of precision and recall)

Threshold based training selection

Data set	Training set	Threshold 0.1	0.3	0.5	0.7	0.9
Census	$\mathbf{W}_M$	0	100	96.2	73.4	67.9
449 + 392	$\mathbf{W}_N$	0	0	100	100	100
Restaurant	$\mathbf{W}_M$	100	98.5	4.5	0.19	0.2
864	$\mathbf{W}_N$	0	0	100	100	100
Synthetic	$\mathbf{W}_M$	0	100	100	100	100
1,000	$\mathbf{W}_N$	100	100	100	99.0	86.1
Synthetic	$\mathbf{W}_M$	100	100	100	98.0	96.5
5,000	$\mathbf{W}_N$	100	100	100	99.7	96.3
Synthetic	$\mathbf{W}_M$	100	100	100	95.5	93.6
10,000	$\mathbf{W}_N$	99.2	99.7	100	99.9	98.3

Nearest based training selection (balanced)

Data set	Training set	Non-unique nearest 1%	5%	10%	Unique nearest 1%	5%	10%
Census	$\mathbf{W}_M$	100	100	81.8	100	100	79.9
449 + 392	$\mathbf{W}_N$	100	100	100	100	100	100
Restaurant	$\mathbf{W}_M$	9.8	2.0	1.0	5.6	1.1	0.59
864	$\mathbf{W}_N$	100	100	100	100	100	100
Synthetic	$\mathbf{W}_M$	100	100	100	100	100	100
1,000	$\mathbf{W}_N$	100	96.7	95.5	100	95.9	95.5
Synthetic	$\mathbf{W}_M$	100	100	99.0	100	100	99.0
5,000	$\mathbf{W}_N$	100	99.8	99.5	99.7	99.7	99.6
Synthetic	$\mathbf{W}_M$	100	99.0	75.4	100	98.6	74.1
10,000	$\mathbf{W}_N$	100	99.8	99.7	99.8	99.8	99.7

Nearest based training selection (im-balanced)

Data set	Training set	Non-unique nearest 1%	5%	10%	Unique nearest 1%	5%	10%
Census	$\mathbf{W}_M$	100	100	100	100	100	100
449 + 392	$\mathbf{W}_N$	100	100	100	100	100	100
Restaurant	$\mathbf{W}_M$	100	100	90.8	100	76.7	58.6
864	$\mathbf{W}_N$	100	100	100	100	100	100
Synthetic	$\mathbf{W}_M$	100	100	100	100	100	100
1,000	$\mathbf{W}_N$	100	96.7	95.5	100	95.9	95.5
Synthetic	$\mathbf{W}_M$	100	100	100	100	100	100
5,000	$\mathbf{W}_N$	100	99.8	99.5	99.7	99.7	99.6
Synthetic	$\mathbf{W}_M$	100	100	100	100	100	100
10,000	$\mathbf{W}_N$	99.9	99.8	99.7	99.8	99.8	99.7

Record pair classification results

Classification approach	Data sets Cens.	Rest.	S-1,000	S-5,000	S-10,000
SVM	0.785	0.466	0.944	0.884	0.829
K-means	0.434	0.002	0.802	0.763	0.213
Threshold-0.3	0.000	0.000	0.857	0.735	0.655
Threshold-0.5	0.187	0.001	0.711	0.527	0.751
Threshold-0.7	0.171	0.704	0.826	0.744	0.492
Near-5%, NU, B	0.643	0.001	0.865	0.573	0.199
Near-5%, U, B	0.500	0.002	0.861	0.582	0.203
Near-5%, NU, IB	0.644	0.012	0.851	0.805	0.751
Near-5%, U, IB	0.511	0.005	0.849	0.807	0.756

Outlook and future work

- The proposed two-step record pair classification approach shows promising results
  - Can automatically select good quality training examples
  - Nearest-based often outperforms threshold based
- Improvements for second step (classification)
  - Apply classifier iteratively (improve training sets)
  - Randomly add additional training examples from the 'gap' between training sets (similar to 'spy' documents in *S-EM* text classification)
- More experiments on different data are needed
- Also to be done is to investigate scalability