Automatic training example selection for scalable unsupervised record linkage

Peter Christen

Department of Computer Science,
The Australian National University,
Canberra, Australia

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 challenges
- 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?

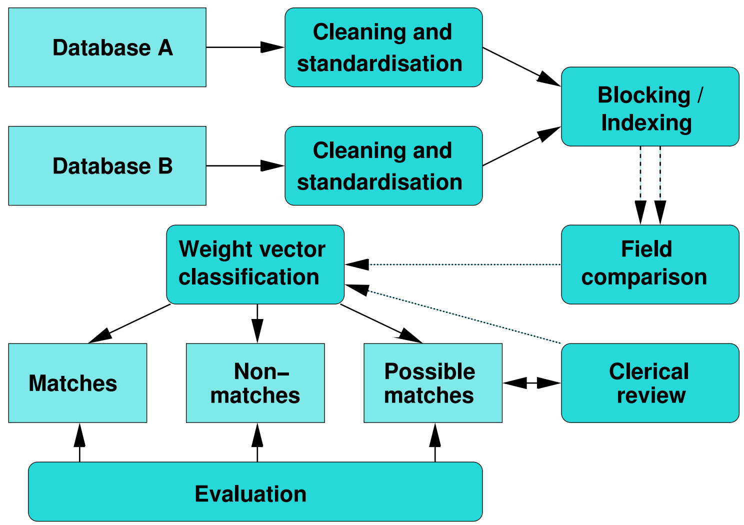
- The process of linking and aggregating records from one or more data sources representing the same entity (such as a patient, customer, or business)
  - Also called *data matching*, *data scrubbing*, *entity resolution*, *object identification*, *merge-purge*, etc.
- Challenging if no unique entity 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 challenges

- Real world data is dirty
  (typographical errors and variations, missing and out-of-date values, different coding schemes, etc.)
- Scalability
  - Naïve comparison of all record pairs is $\mathbf{O(n^2)}$
  - Some form of blocking, indexing or filtering required
- No training data in many linkage applications
  - No data sets with known true match status
  - Possible to manually prepare training data (but, how accurate will manual classification be?)

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 similarity, 0.0 for total dissimilarity
- For each compared record pair a *weight 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]

- Weight vectors (record pairs) are classified into *matches*, *non-matches* (and *possible matches*)

Record pair classification

- Traditionally, matching weights are summed, and two thresholds are use for classification
- Various machine learning techniques have been investigated
  - Supervised: SVM, decision trees, neural networks, learnable string comparisons, active learning, etc.
  - Un-supervised: Different *clustering* algorithms
- Recently, *collective* entity resolution techniques have been investigated
  - Rather than classifying each record pair independently
  - Using relational attributes (i.e. graph based)
  - However, not all data is relational

Two-step record pair classification

- Assumptions
  - Weight vectors that have exact or high similarity values in all 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 examples 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

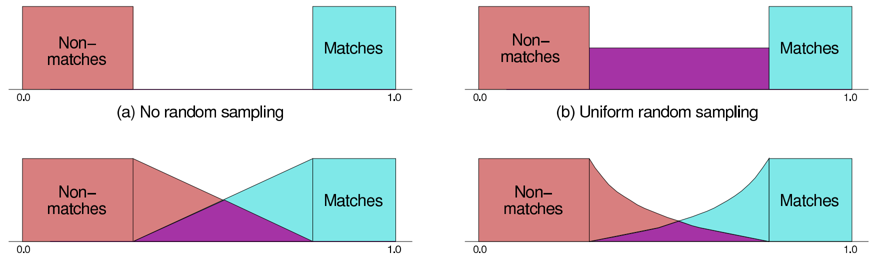
Records and weight vectors example

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	0.3	0.5	0.7	0.9

Step 1: Training example selection

- Weight vectors can be selected using either *thresholds* or *nearest based*
- Training examples are likely linearly separable
- Idea: randomly add more training examples (from *gap* between match and non-match examples)



(a) No random sampling (b) Uniform random sampling (c) Linear random sampling (d) Exponential random sampling

Step 2: Classification of record pairs

- Any binary classifier can be used (in the following experiments, a linear SVM has been employed)
- Question investigated here: *Does the random inclusion of additional weight vectors improve classification accuracy?*
- 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, improve iteratively
  - *S-EM* (seed expectation-maximisation): add 'spy' documents from positive examples into unlabeled data

Experimental evaluation

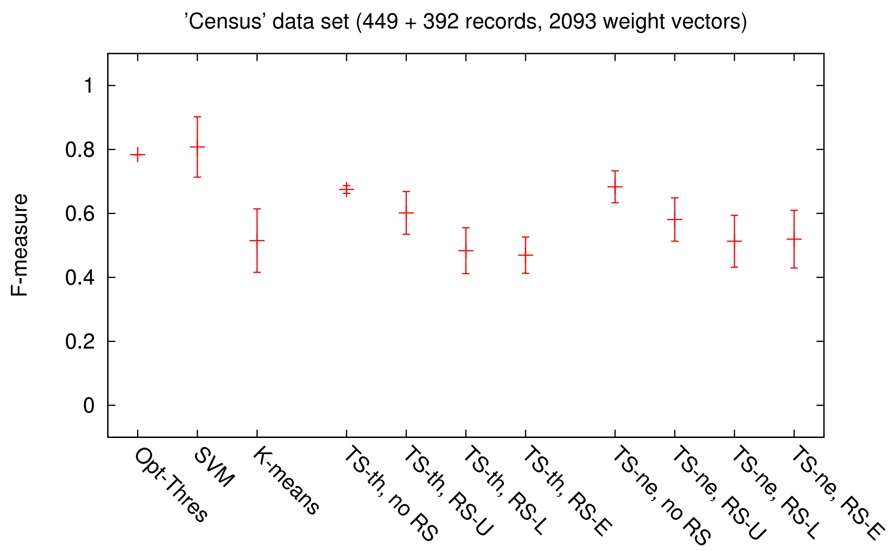
- All techniques are implemented in the *Febrl* open source record linkage system (available from: https://sourceforge.net/projects/febrl/)
- Experiments using both real and synthetic data (*Secondstring* repository and *Febrl* data set generator)
- Evaluation of step 1 (training example selection)
  - Percentage of true matches and true non-matches in the training example sets
- Evaluation of step 2 (record pair classification)
  - *F*-measure (harmonic mean of precision and recall) (average and standard-deviation are shown in graphs)

Quality of weight vectors selected

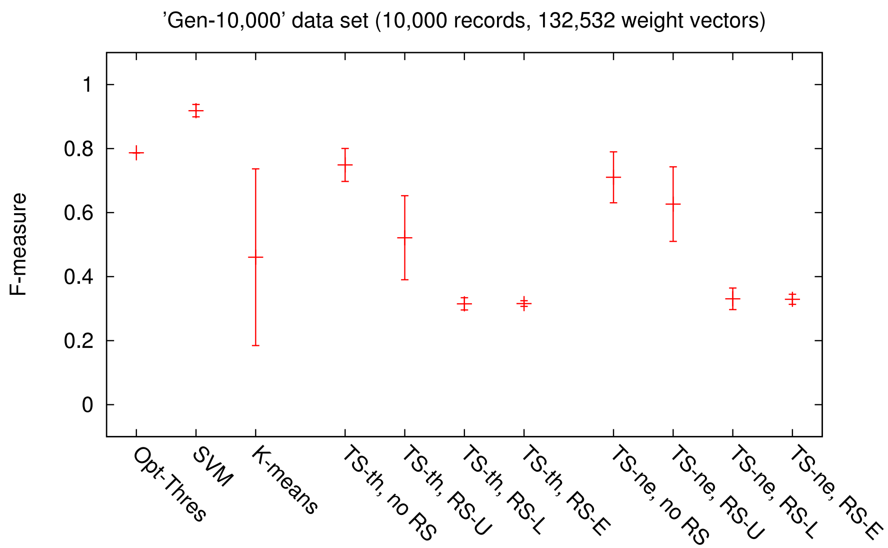
Data sets	Thresholds		Nearest	
	0.3	0.5	1%	10%
Census	100/–	96.2/100	100/100	100/100
Restaurant	98.5/–	4.5/100	100/100	58.6/100
Gen-1,000	100/100	100/100	100/100	100/95.5
Gen-2,500	100/100	100/100	100/99.0	100/98.2
Gen-5,000	100/100	100/100	100/99.7	100/99.6
Gen-10,000	100/99.7	100/100	100/99.8	100/99.7

- Results given here are percentage values for match/non-match sets

Record pair classification for Census



Record pair classification for Gen-10,000



Outlook and future work

- The proposed two-step record pair classification approach shows promising results
  - Can automatically select good quality training examples
  - Random inclusion of additional weight vectors does **not** improve classification accuracy (unlike improvements in Web and text classification)
- Improvements for second step (classification)
  - Apply classifier iteratively (as done in *PEBL* approach)
  - Investigate nearest-neighbour based classification
- More experiments on different data are needed
  - Also investigate the scalability of this approach