Australian National University

COMP3430 / COMP8430
Data wrangling

In person lecture week 7

(Lecturer: Peter Christen)

Some administrative things

- **We are still marking assignment 1**
  - Aim to release marks by end of next week
  - Some initial feedback: **Follow the specifications** (maximum page numbers, naming of files, etc.), and **read questions very carefully**!
- Assignment 2, and record linkage (COMP3430) and data wrangling (COMP8430) projects online

Record linkage project (**COMP3430**)

- 20% of final course mark, due week 11
  (Sunday 21 October, 11:55 pm)
- **Focus is on understanding and evaluating a record linkage project**
  - Justify and describe your choice of technique
  - Evaluate different approaches
  - Identify a best possible approach to link the provided data sets (and submit your linked file)
  - Appropriately present results (using tables and plots)

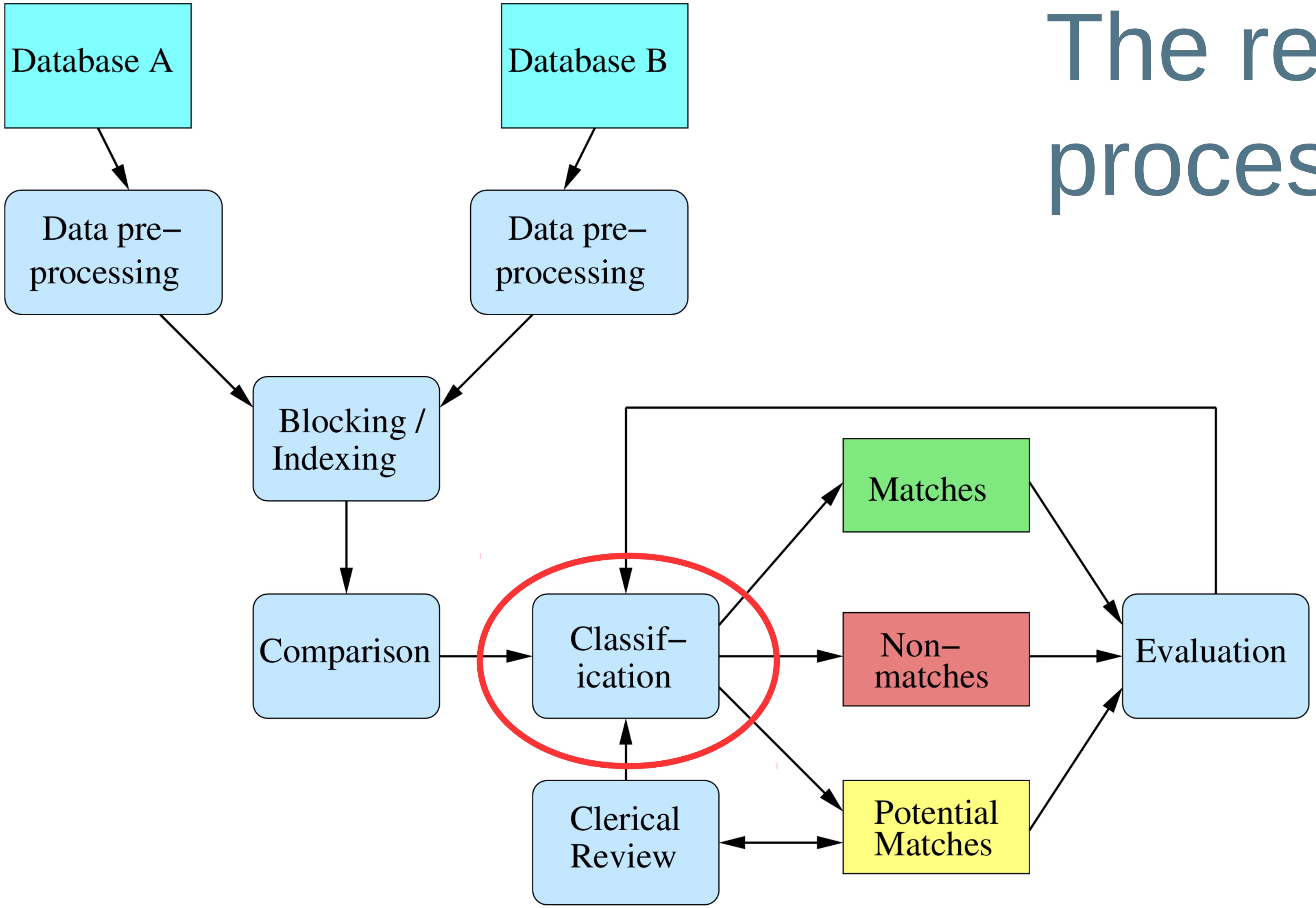
Data wrangling project (**COMP8430**)

- 30% of final course mark, due week 11
  (Sunday 21 October, 11:55 pm)
- **Focus is on understanding and evaluating a data wrangling project**
  - On data sets of your choice (justify and describe your choice)
  - **Data wrangling based on an assumed end-use of your data sets** (all wrangling must be done with regard to this end-use)
  - **At least two data sets, that need to be somehow integrated**
  - Appropriately presented results (using tables and plots)

Questions from Wattle forum

- For probabilistic classification, what is the rationale of using 2 as logarithm base?
- Probabilistic classification and threshold-based classification seem to be similar to each other. I notice the difference is the former gives different weights to true matches and true non-matches. Is this the only difference?
- In the textbook, Equation 6.7 leads to optimal decision making. What does this mean?
- For probabilistic record linkage, whether there will be a threshold for match and non-match for the similarity value of individual attributes. Match and Non-Match result in different weights, while using approximate (string) comparison functions cannot get the binary result.

The record linkage process



Classifying record pairs (1)

- The comparison step generates one vector of similarities (also known as *weight vector*) for each of compared record pair
- The elements of such vectors are the calculated similarities (exact or approximate)

- For example: (assuming edit distance calculations)

	Tim	Paul	Miller	23	Main	Street	Dickson
	Tim	P	Miller	4/23	Main	St	Dixon
Exact comparison:	1.0	0.0	1.0	0.0	1.0	0.0	0.0
Approximate comparison:	1.0	0.25	1.0	0.5	1.0	0.4	0.57

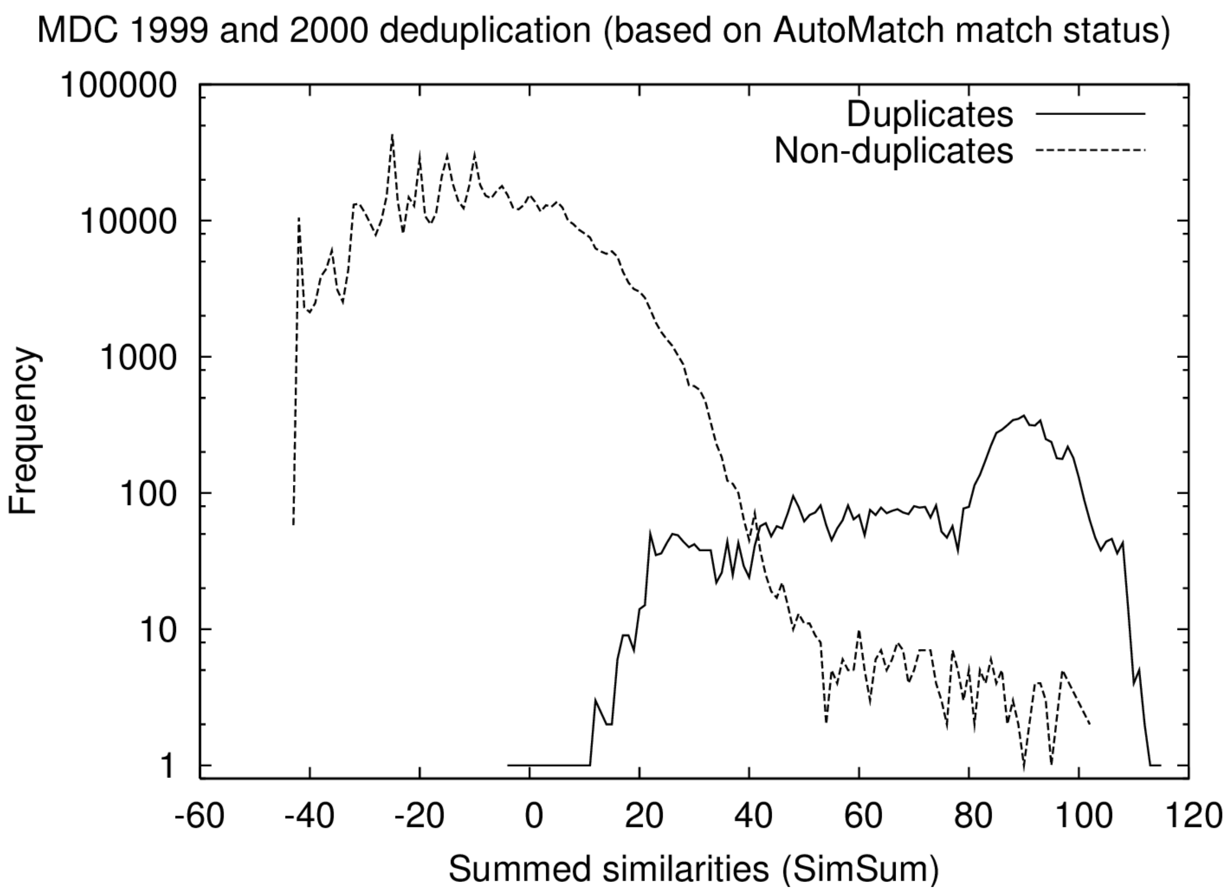
Classifying record pairs (2)

- Classifying record pairs can be based on (a) summing the calculated similarities into a single similarity values, or (b) using the full vector of similarities

	Tim	Paul	Miller	23	Main	Street	Dickson	
	Tim	P	Miller	4/23	Main	St	Dixon	Sum:
Exact comparison:	1.0	0.0	1.0	0.0	1.0	0.0	0.0	3.0 / 7
Approximate comparison:	1.0	0.25	1.0	0.5	1.0	0.4	0.57	4.72 / 7

Example histogram of summed similarities

- Deduplication of a health data set with different *weights* attached to different similarities, and where the true match status was determined using the commercial record linkage software *AutoMatch.*
  (from Christen, 2012)



Threshold based classification (1)

- Is generally applied on summed similarities
- Can either use one or two similarity thresholds
  - One threshold *t*: $0 \leq t \leq sim_{max}$, where sim_{max} is equal to the number of similarities in the vectors

    (a) Record pairs with a similarity of at least *t* → *Classified match*

    (b) Record pairs with a similarity below *t* → *Classified non-match*

  - Two thresholds t_l *and* t_u: $0 \leq t_l < t_u \leq sim_{max}$

    (a) Record pairs with a similarity of at least t_u → *Classified match*

    (b) Record pairs with a similarity below t_l → *Classified non-match*

    (c) Record pairs with a similarity between t_l *and* t_u → *Classified potential match*

Threshold based classification (2)

- If similarities are simply summed then each attribute has the same importance (or same *weight*)
  - Does having the same gender say as much about two records being about the same person as having the same postcode?
- A weighted sum approach provides more weight to attributes that contain more information
  - Weights can be based on domain knowledge
  - Or they can be calculated based on the number of unique values in an attribute *a*:

    $w_a = log(number\ of\ unique\ attribute\ values)$

Threshold based classification (3)

- Total similarity is then a weighted sum:

$$sim(rec_i, rec_j) = \sum_a sim(rec_i[a], rec_j[a]) * w_a,$$

  where w_a is the weight for attribute a
- To normalise this similarity into the 0..1 interval we can divide $sim(rec_i, rec_j)$ by $\sum_a w_a$
- Further weight calculations take the frequencies of values into account
  - Two records with the common surname 'Smith' are less likely to refer to the same person compared to two records with the rare surname 'Dijkstra'

Probabilistic classification (1)

- Known as *probabilistic record linkage*
  - Basic ideas were introduced by Newcombe and Kennedy in 1962
  - Theoretical foundation by Fellegi and Sunter in 1969
- Basic idea:
  - Compare common record attributes (or fields) using approximate (string) comparison functions
  - Calculate matching weights based on frequency ratios (global or value specific ratios) and error estimates
  - Sum of the matching weights is used to classify a pair of records as a *match*, *non-match*, or *potential match* (using two thresholds)
- Problems: Estimating errors, find optimal thresholds, assumption of independence, and manual clerical review

Probabilistic classification (2)

- A ratio R is calculated for each compared record pair $r = (a,b)$ in the product space $A \times B$:

$$R = P(\gamma \in \Gamma \mid r \in M) \,/\, P(\gamma \in \Gamma \mid r \in U),$$

where M and U are the sets of true matches and true non-matches, and γ (gamma) is an agreement pattern in the comparison space Γ (Gamma), with:

$A \times B = \{(a, b) : a \in A, b \in B\}$ for files (data sets) A and B
$M = \{(a, b) : a = b, a \in A, b \in B\}$ True matches
$U = \{(a, b) : a \neq b, a \in A, b \in B\}$ True non-matches

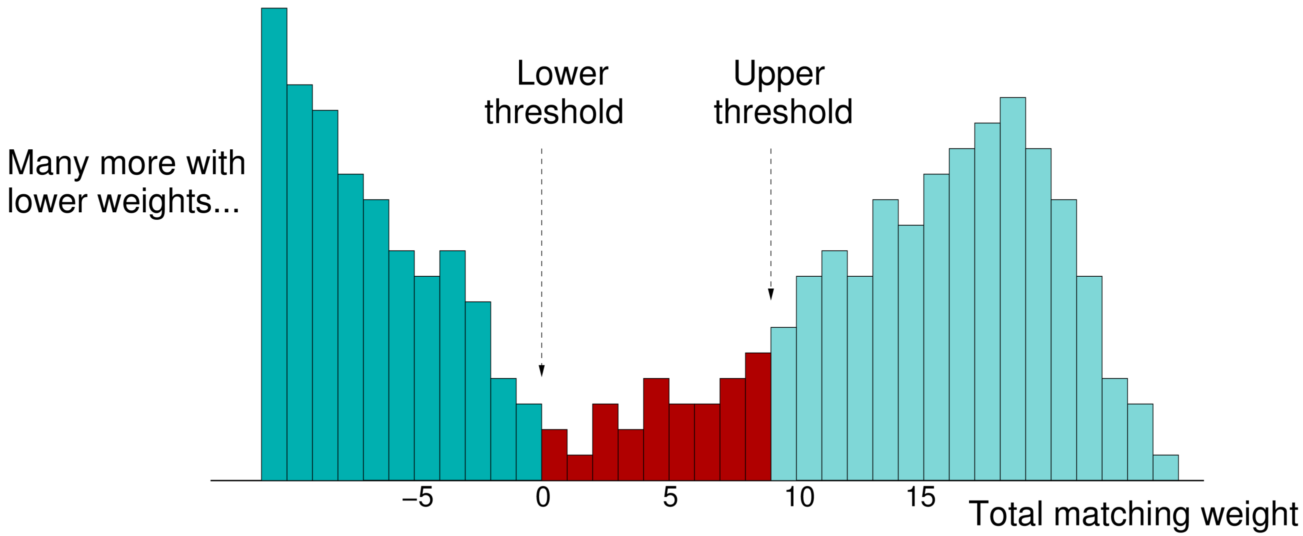
Probabilistic classification (3)

- Fellegi and Sunter proposed the following decision rule:

  $R \geq t_u \rightarrow$ r is classified as a match

  $t_l < R < t_u \rightarrow$ r is classified as a potential match

  $R \leq t_l \rightarrow$ r is classified as a non-match



Probabilistic classification (4)

- Assuming conditional independence between attributes allows to calculate individual attribute-wise probabilities
  $m_i = P([a_i = b_i\ , a \in A, b \in B] \mid r \in M)$ and
  $u_i = P([a_i = b_i\ , a \in A, b \in B] \mid r \in U)$,
- where a_i and b_i are the values of attribute *i* being compared
- Based on these *m*- and *u*-probabilities, we calculate a *matching weight* w_i for attribute *i* as:
  $w_i = log_2(m_i / u_i)$ if $a_i = b_i$ Agreement weight
  $w_i = log_2((1\text{-}m_i) / (1\text{-} u_i))$ if $a_i \neq b_i$ Dis-agreement weight

Weight calculation example

- Assume two data sets with a 3% error in attribute *month of birth*
- Probability that two matched records (representing the same person) have the same month value is 97% (m_i)
- Probability that two matched records do not have the same month value is 3% ($1 - m_i$)
- Probability that two (randomly picked) un-matched records have the same month value is 1/12 = 8.3% (u_i)
- Probability that two un-matched records do not have the same month value is 11/12 = 91.7% ($1 - u_i$)
- Agreement weight: $log_2(m_i / u_i) = log_2(0.97 / 0.083) = 3.54$
- Disagreement weight $log_2((1-m_i) / (1- u_i)) = log_2(0.03 / 0.917) = -4.92$

Cost based classification (1)

- In record linkage classification we can make two types of mistakes

  (1) A record pair that is a true match (same entity) is classified as a non-match (**false negative**)

  (2) A record pair that is a true non-match (different entities) is classified as a match (**false positive**)
- Traditionally it is assumed both types of errors have the same costs
- **Question**: *In which applications / situations do these two types of errors have different costs?*