Review 1: Entity Resolution, Blocking, Precision and Recall

Olivier Binette

Entity Resolution

Databases, Records and Attributes

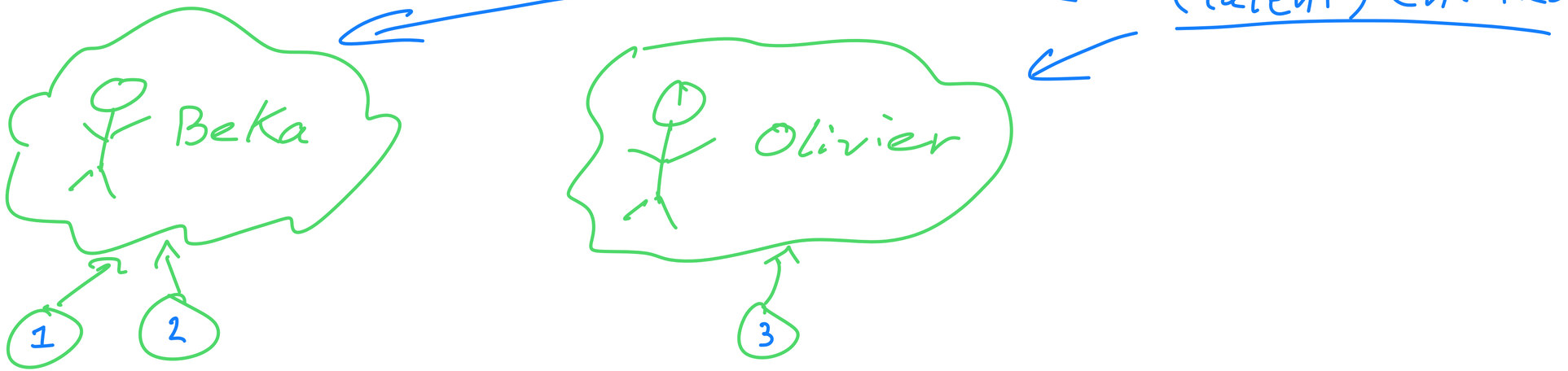
Database A

	Name	b.y.	occupation
1.	Beka S.	–	Professor
2.	Rebecca C.S.	–	Statistician
3.	Binette, O.	95	PhD Student

} attributes

} records

(latent) entities



Entity Resolution

Two records are said to be coreferent or to match if they refer to the same entity.

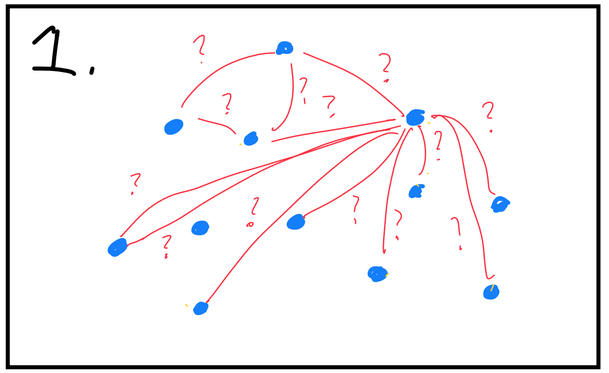
Goal of Entity Resolution:

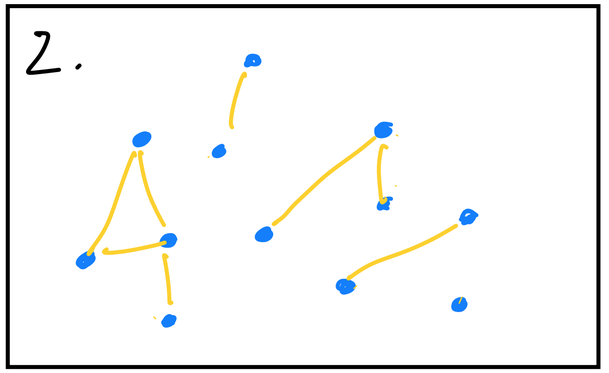
Cluster coreferent records <==> identify matching pairs

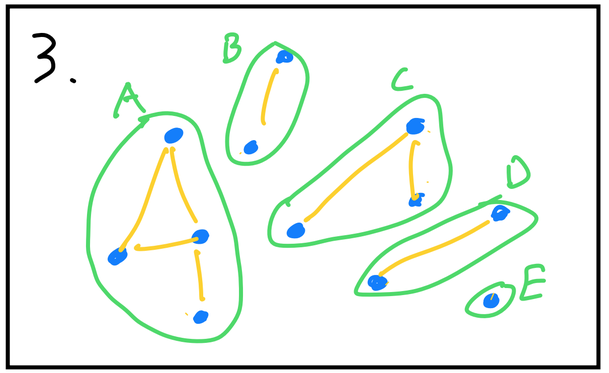
Entity Resolution

Typical approach to ER:

1. Consider all pairs of records
   - The $\binom{N}{2}$ pairs are the comparison space
2. Classify each pair as being a match (link) or a non-match (don't link)
   - This gives a graph
3. Assign a unique entity ID to each record based on the resulting structure
   - E.g. use connected components of the graph







Entity Resolution

Problem:

The comparison space is large ($O(N^2)$)

↳ Many pairs to consider!

Solution:

Blocking...

Blocking

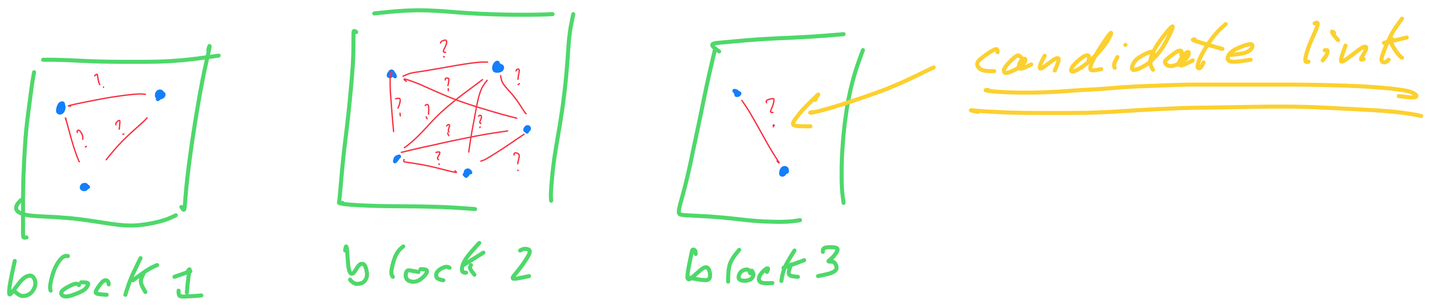
Blocking

Goal of blocking:

- Quickly ($< O(N^2)$) reduce the comparison space

Trick:

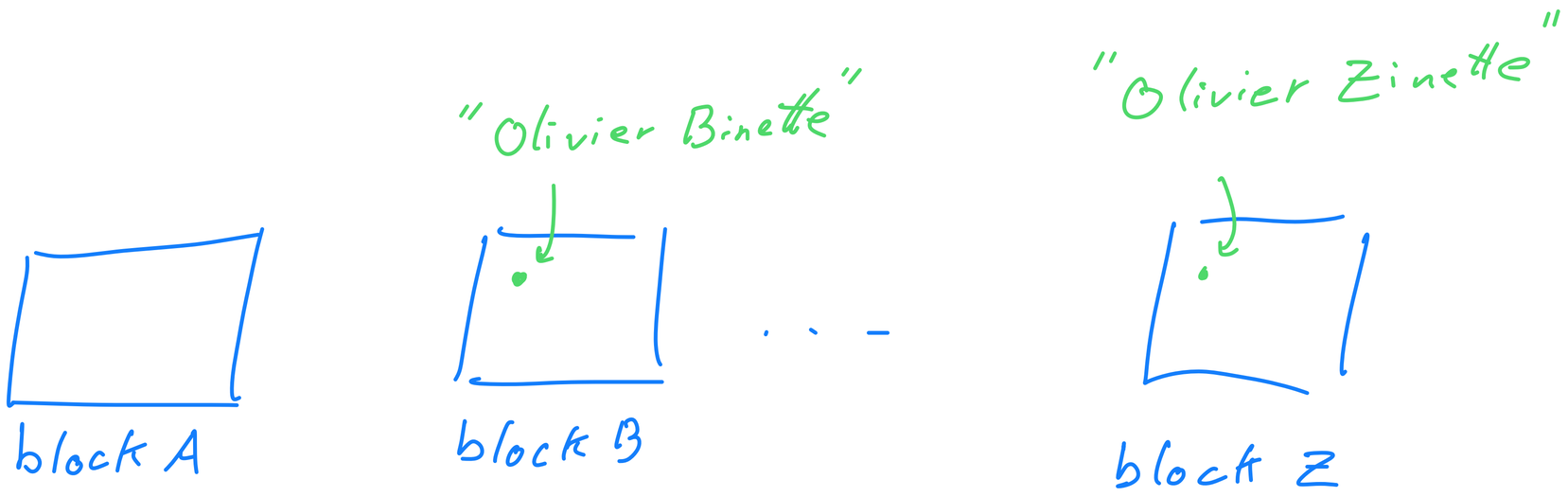
- Place each record into a "block"
  - ↳ You want matching records in the same block
- Only make comparisons within blocks
  - ↳ Here we focus on blocks that partition the set of records



Blocking

Example:

- Block by last name initial
  - Each record is placed in a block given the initial of the listed last name.



Blocking

The challenge of blocking:

Balance the tradeoff between:

1. Small blocks (fewer comparisons & more efficient)
2. Few mistakes (don't place matching records in different blocks)

How to quantify this tradeoff?

1. Reduction ratio
2. Recall

Reduction ratio

Suppose you have N records.

- Without blocking, you need $\binom{N}{2}$ comparisons
- With K blocks, each of size $b_1, b_2, \ldots, b_K$, you need $\sum_{i=1}^{K} \binom{b_i}{2}$ comparisons

Reduction ratio:

$$RR = \frac{\binom{N}{2} - \sum_{i=1}^{K} \binom{b_i}{2}}{\binom{N}{2}} = \text{"\% of comparisons that blocking has eliminated"}$$

Recall

Recall = "% of matching pairs that are in the same block"

low recall => many mistakes will necessarily be made

perfect recall (100%) => no accuracy loss in doing blocking.

Precision and Recall

(more generally)

Precision and Recall

Consider a test (e.g. covid test)

↳ Result is positive (P) or negative (N)

→ Patient is sick (C) or not sick ($\bar{C}$)

	C	$\bar{C}$
P	TP	FP
N	FN	TN

$$\text{precision} = \mathbb{P}(C \mid P) = \frac{\mathbb{P}(C \cap P)}{\mathbb{P}(P)} = \frac{TP}{TP + FP}$$

"= % of positive patients which are actually sick"

$$\text{recall} = \mathbb{P}(P \mid C) = \frac{\mathbb{P}(P \cap C)}{\mathbb{P}(C)} = \frac{TP}{TP + FN}$$

"= % of sick patients which have a positive result".

Application to blocking

- We have record pairs
- Each is a candidate pair (P) or not (F)
  - ↳ in the same block
- Each is actually a match (M) or not ($\bar{M}$)

	M	$\bar{M}$
P	TP	FP
F	FN	TN

$$\text{recall} = \frac{TP}{TP + FN}$$

$$= \frac{\#\ P \cap M}{\#\ M}$$

$$=$$ "% of matches which are candidate pairs"

Computing Recall

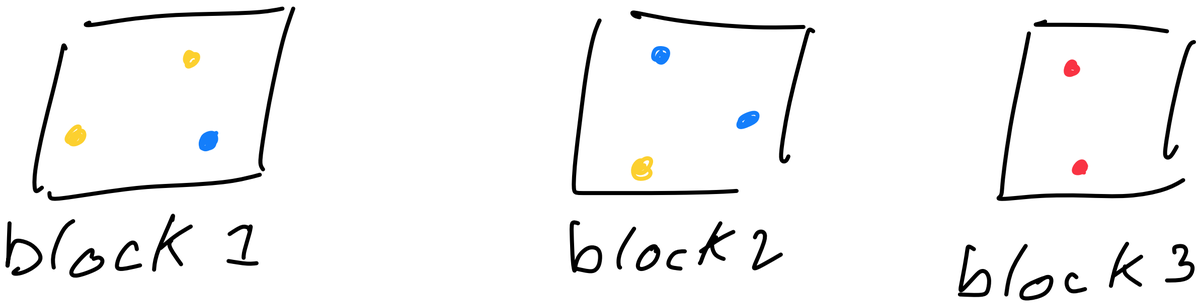
How you compute is tied to your data representation

Here suppose we have:

- A variable "blockID" for each record indicating block membership
- A variable "entityID" representing the true entity ID for the record.

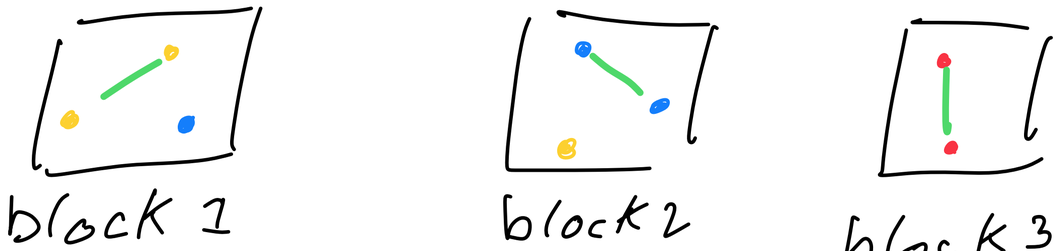
→ These are called membership vectors

→ Candidate pairs and matching pairs are defined implicitely



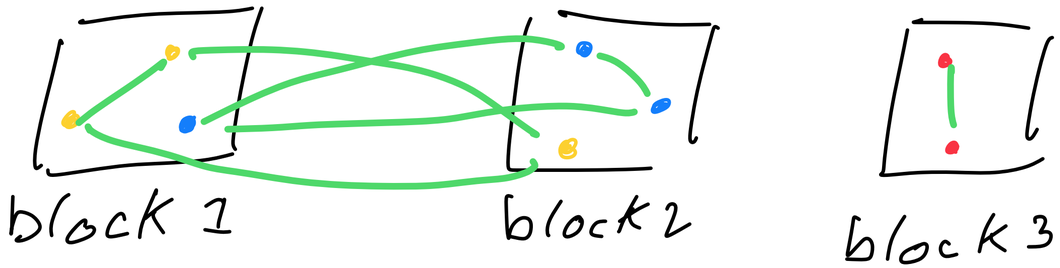
Computing Recall

TP = 3



• = entity 1
• = entity 2
• = entity 3

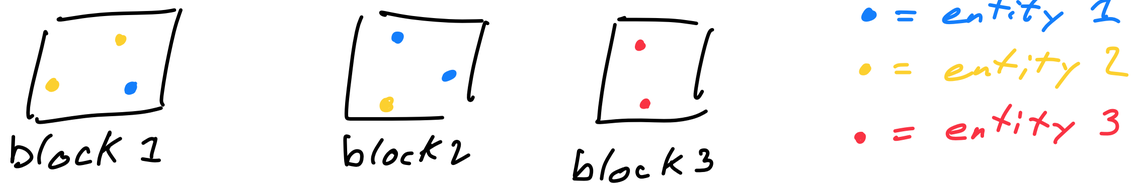
matching pairs = 7



• = entity 1
• = entity 2
• = entity 3

recall = 3/7

Recall in R




```
ct = xtabs(~ blockID + entityID)
```

	ent. 1	ent. 2	ent. 3
block 1	2	1	0
block 2	1	2	0
block 3	0	0	2

cs = colSums(ct) = [3 3 2]

# of each entity

TP = ?

matching pairs = ?

Is this efficient?

Computing precision

$$precision(blockID, entityID) = recall(entityID, blockID)$$

why ?