Neighbourhood Blocking for Record Linkage

Dan Elias

Josiah Poon

ADC19, January 2019

Record linkage

Indexing

Neighbourhood Blocking

Scalability & quality comparisons

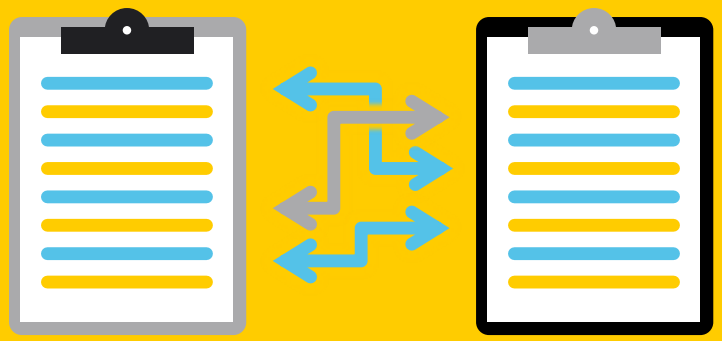
Conclusion

Indexing in Record Linkage

Record linkage

Indexing

Existing indexing methods



Record Linkage

- Finding pairs of database records that refer to the same entity
- Definitive key fields not used / absent
- Applications
  - Data integration
  - Deduplication
  - Fraud detection

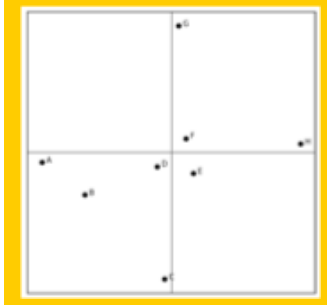
Given name	Surname	DOB	SSID	Address	Height	Weight
Catherine	Bourke	15/03/1958	3984257	42 Peach Crs	173	76
Cathy	Smythe	15/03/1958	398425	42 Peach Crs	172.5	
Timothy	Bourke	06/12/1959	3939872	42 Peach Crs	180	82.5
Tim	Bourk	06/12/1995	3939872		181	86

Indexing

- Selection of record pairs to be compared
- All possible record pairs (Full index) quadratic in number of records
- True matches are rare
- Objectives
  - High recall (low precision OK)
  - Exclude most record pairs

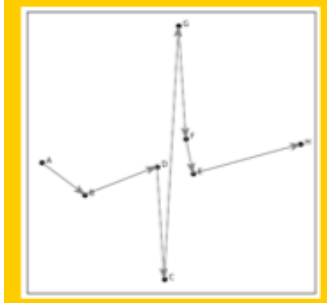
Item	Quantity
Total records	100,000
Duplicates	5,000
Total record pairs	4,999,950,000
True matches	5,000
True matches / total pairs	1 / 999,990

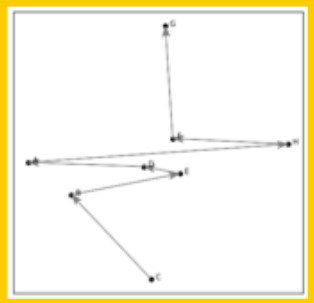
Existing indexing methods



Standard blocking (= inner join)

- Possibly via many:1 mapping (eg: discretization, soundex codes)
- Excludes nearby pairs that straddle boundaries





Sorted Neighbourhood

- Generalization of Standard Blocking
- Adds: proximity in a (single) ordering of blocks
- Includes very distant pairs

Other methods

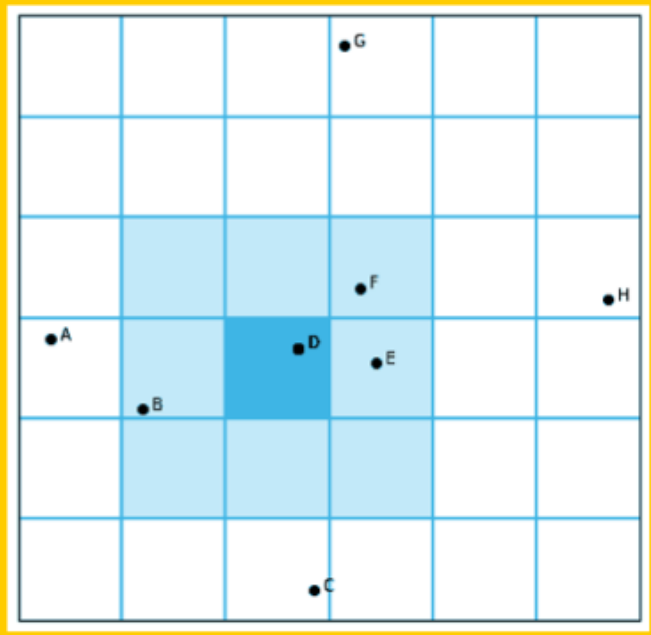
- Unique inner join via some other many:many mapping
  - eg: q-grams, prefix/suffix arrays, canopy clusters
- Progressive blocking – prioritize blocks by density of matches found so far

Neighbourhood Blocking

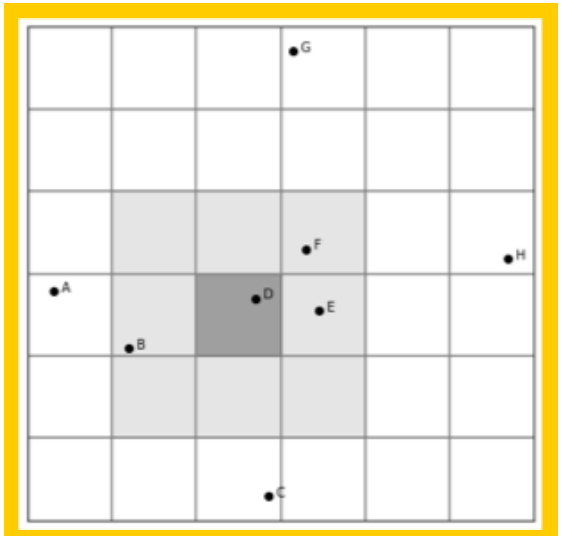
Matching criteria

Algorithm

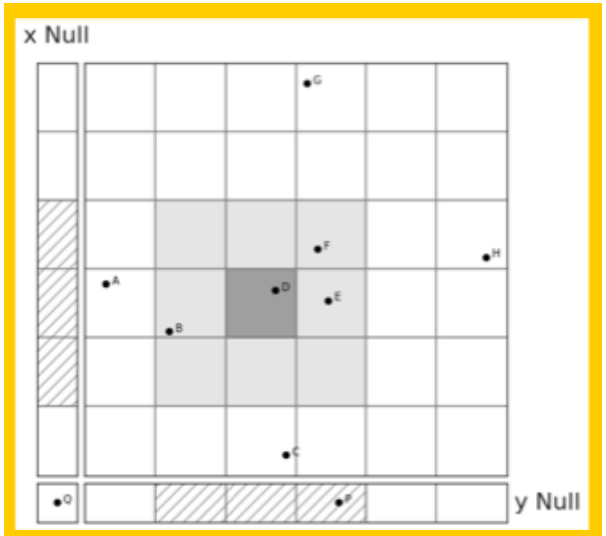
Theorems



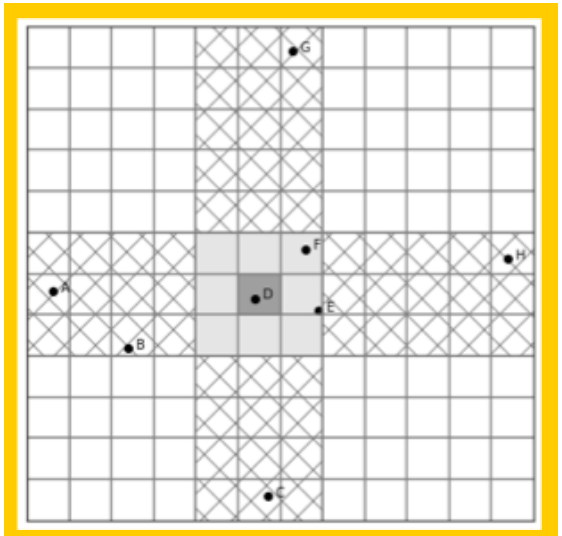
Matching criteria



Simultaneous proximity in multiple orderings



Limited wildcard matching of missing values



Limited unmatched (populated) fields

Parameters

- Blocking fields, and rank distance limit for each one
- Maximum number of wildcard matches (missing values)
- Maximum number of populated field mismatches

Algorithm

1. Replace Blocking Key Values with integer ranks (nulls for missing values)
2. Distinct BKV combinations → block definitions
3. Make Linkage index for block definitions. Either:
   - BKVs maximally coarse → Full Index
   - Otherwise:
     - Coarsen blocks using integer division, adjust rank distance lmits
     - Produce Neighbourhood Blocking index on coarsened blocks

   Recursion depth: log (max cardinality)
4. Compare block pairs selected in indexing step, determine matching ones
5. Produce inner join of original records via table of block matches

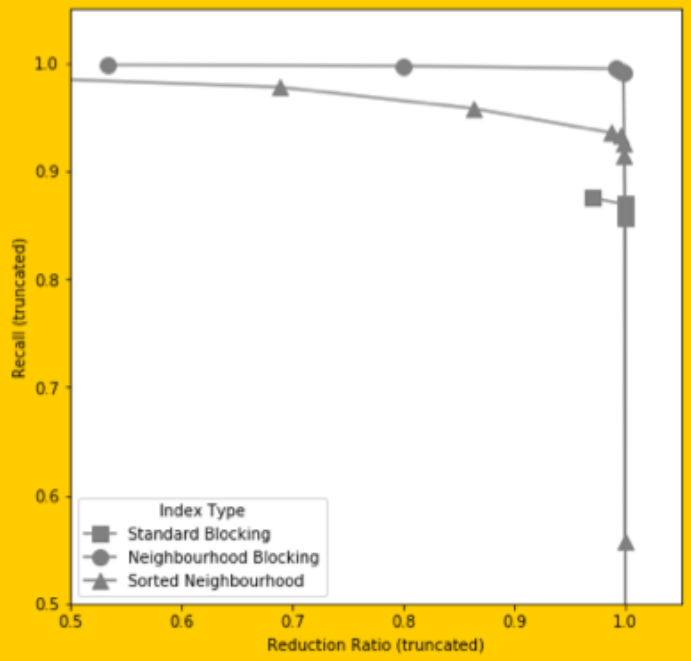
Theorems

- Inclusion distance (discretized continuous-valued fields)
  - Distance = lowest of: (unit of discretization) x (rank mismatch limit)
  - All record pairs with Euclidean distance below this are included in the index
- Superset of coarsened Standard Blocking
  - When: adjacent blocks coalesced by no more than (rank mismatch limit + 1) : 1
- Index size vs Standard Blocking (“large, uniform dataset”)
  - Product of: (2 * rank mismatch limit + 1) for each blocking key

Comparisons

Index quality

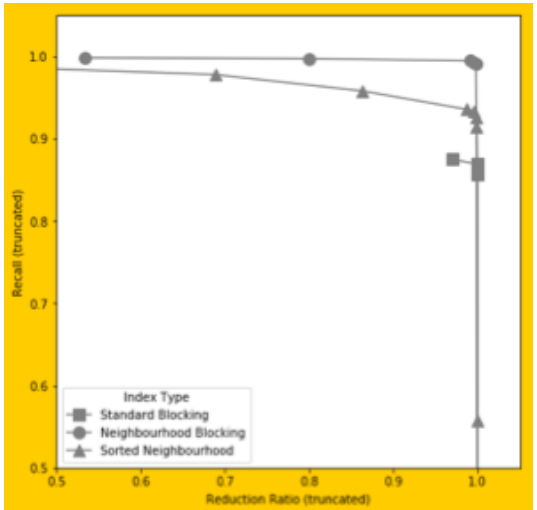
Scalability



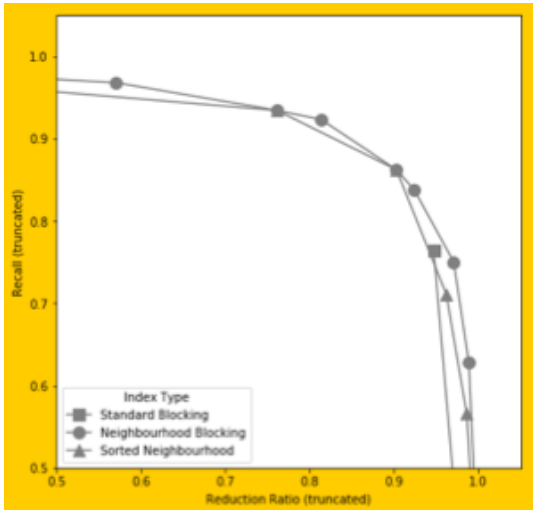
Index quality - benchmark datasets

Tradeoffs between

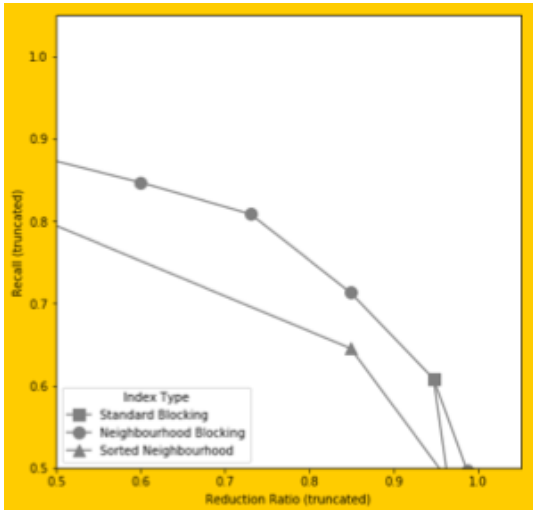
- Recall (% true matches included in index)
- Reduction ratio (% size reduction from Full Index)



FEBRL 3



Abt-Buy



Amazon-Google Products

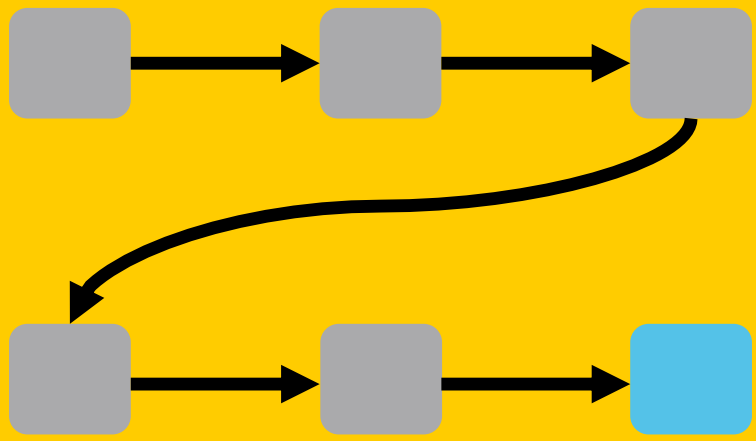
Scalability – random datasets

Lines of best fit – index production time vs index size

- Random datasets up to 1 million rows, up to 10 columns, various cardinalities
- Sparsity (records / block) has a big impact on this measure

Method	Dataset constraint	Intercept (seconds)	Slope (seconds / million row pairs)	R^2
NB – no wildcards or adjacency		4.92	3.18	0.33
NB - 1 wildcard		9.23	3.02	0.09
Full		1.19	1.08	0.99
Sorted Neighbourhood		1.22	10.34	0.96
Standard Blocking		1.55	3.23	0.82
NB	Non-sparse	0.26	3.20	0.99
Standard Blocking	Non-sparse	0.23	3.13	0.99

Conclusion



Neighbourhood Blocking

- Generalization of Standard Blocking and Sorted Neighbourhood Indexing
  - Index quality often better (can't be worse)
  - Scalability similar (can't be better)
- Applicability indicators
  - Proximity meaningful in multiple fields → apply all simultaneously
  - Continuous-valued fields → finer discretization
  - Missing values → allow some wildcard matching
  - Many blocking fields → include all and allow some mismatches
- Open source implementation
  - Python Record Linkage Toolkit (`pip install recordlinkage`)

Q & A