Probabilistic Blocking and Distributed Bayesian Entity Resolution

Ted Enamorado
Washington University in Saint Louis

Rebecca C. Steorts
Duke University

Sept. 23rd, 2020

Entity resolution is the process of merging large noisy databases to remove duplicate entities , often in the absence of a unique identifier.

This problem is fundamentally important in medicine, official statistics, human rights and modern slavery, voter registration, and many other applications.

Background

Deterministic methods are very popular as they are easily accessible across multiple disciplines and very scalable.

They do not account for the error of the entity resolution process.

Binette and **Steorts** (2020), Under Review

Probabilistic Entity Resolution

- Designed to control error rates (precision and recall)
- Designed to model data with distortions and errors

Probabilistic Entity Resolution

- Designed to control error rates (precision and recall)
- Designed to model data with distortions and errors

Existing open-source implementations need to scale to billions of records

Probabilistic Entity Resolution

- Designed to control error rates (precision and recall)
- Designed to model data with distortions and errors

Existing open-source implementations need to scale to billions of records

1. `fastLink` (Enamorado et al., 2019)
2. `dblink` (Marchant et al., 2020)
3. `dblinkR` (Marchant et al., 2020)

Our goal is to propose a two-stage method that will scale and have a balance regarding uncertainty propagation. This is called:

`fastLink` (blocking) + `dblink` (linkage)

Enamorado and **Steorts** (2020), PSD

Road to Improving Probabilistic Entity Resolution

1. fastLink
2. dblink
3. fastLink + dblink
4. Results:
    - Validation Study: `RLdata10000`
    - Empirical Application: National Long Term Care Study (`NLTCS`)

Road to Improving Probabilistic Entity Resolution

1. fastLink
2. dblink
3. fastLink + dblink
4. Results:
   - Validation Study: `RLdata10000`
   - Empirical Application: National Long Term Care Study (`NLTCS`)

Agreement Patterns

- Two data sets ($\mathcal{A}$ and $\mathcal{B}$) with variables in common

Agreement Patterns

- Two data sets ($\mathcal{A}$ and $\mathcal{B}$) with variables in common
- Agreement value in field a for a pair (i, j)

$$\rho_a(i,j) = \begin{cases} \texttt{agree} \\ \\ \texttt{disagree} \end{cases}$$

	Name			
	First	Last	Age	Street
Data set $\mathcal{A}$				
1	James	Smith	35	Devereux St.
Data set $\mathcal{B}$				
7	James	Smit	43	Dvereux St.
	agree	agree	disagree	agree

Agreement Patterns

- Two data sets ($\mathcal{A}$ and $\mathcal{B}$) with variables in common
- Agreement value in field a for a pair (i, j)

$$\rho_a(i,j) = \begin{cases} \texttt{agree} \\ \\ \texttt{disagree} \end{cases}$$

	Name			
	First	Last	Age	Street
Data set $\mathcal{A}$				
1	James	Smith	35	Devereux St.
Data set $\mathcal{B}$				
7	James	Smit	43	Dvereux St.
	agree	agree	disagree	agree

Agreement pattern $\rho(i,j) = \{\rho_1(i,j), \rho_2(i,j), \ldots, \rho_K(i,j)\}$

The Fellegi-Sunter Model of PRL

- **We observe** the agreement patterns $\gamma(i, j)$
- **We do not observe** the matching status

$$C(i, j) = \begin{cases} \text{non-match} \\ \text{match} \end{cases}$$

The Fellegi-Sunter Model of PRL

- **We observe** the agreement patterns $\gamma(i,j)$
- **We do not observe** the matching status

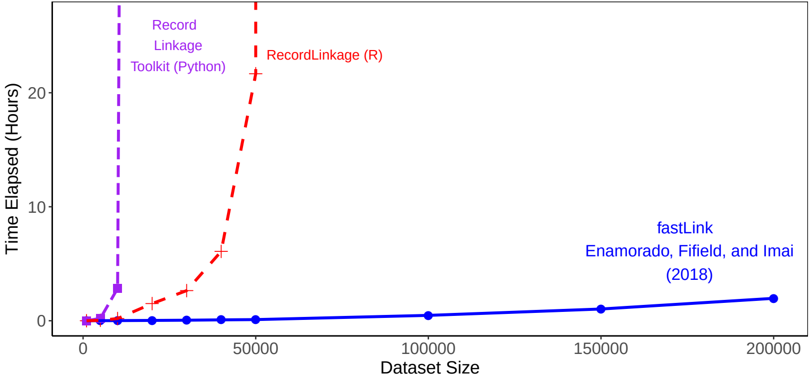
$$C(i,j) = \begin{cases} \text{non-match} \\ \text{match} \end{cases}$$

Mixture Model

$$\begin{aligned} C(i,j) &\overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\mu) \\ \rho(i,j) \mid C(i,j) = \text{non-match} &\overset{\text{i.i.d.}}{\sim} \mathcal{F}(\pi_{\text{NM}}) \\ \rho(i,j) \mid C(i,j) = \text{match} &\overset{\text{i.i.d.}}{\sim} \mathcal{F}(\pi_{\text{M}}) \end{aligned}$$

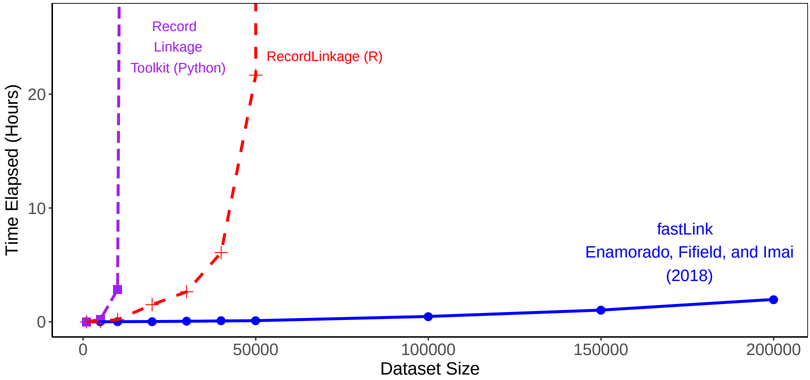
- Where λ, π_{M}, π_{NM} are estimated via the EM algorithm

Runtime Comparison



- Data sets of equal size
- Variables in common: first and last name; house number, street name and zip code; age

Runtime Comparison



- Data sets of equal size
- Variables in common: first and last name; house number, street name and zip code; age
- **Key:** `Sparse matrix` representation of a `hash table`

Enamorado, Fifield, and Imai (2019), APSR, In Press

Road to Improving Probabilistic Entity Resolution

1. fastLink

2. dblink

3. fastLink + dblink

4. Results:

   - Validation Study: `RLdata10000`

   - Empirical Application: National Long Term Care Study (`NLTCS`)

Distributed end-to-end Bayesian Entity Resolution (dblink)

- The goal of dblink:

  Scaling Bayesian ER methods to millions of records without sacrificing accuracy and crucially giving uncertainty of the ER task

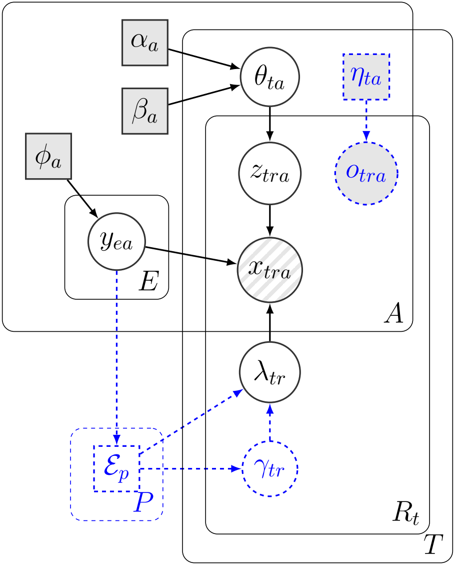
  Marchant, Kaplan, Elzar, Rubinstein, **Steorts** (2020), JCGS, In Press

Distributed end-to-end Bayesian Entity Resolution

1. Proposing the first joint blocking and entity resolution model that scales to millions of records.
2. Utilizes auxiliary partitions (blocks) that induce conditional independencies between latent entities, which enables distributed inference at the partition-level.
3. A blocking function (responsible for partitioning the entities), which groups similar entities together while achieving well-balanced partitions.
4. Application of partially-collapsed Gibbs sampling in the context of distributed computing.
5. Improving the overall computational efficiency.
6. Applying the proposed methodology to six synthetic and real data sets, including a case study of the 2010 decennial census.
7. Open source code available in Apache Spark and R.

Marchant, Kaplan, Elzar, Rubinstein, **Steorts** (2020), JCGS, In Press

Distributed end-to-end Bayesian Entity Resolution



Road to Improving Probabilistic Entity Resolution

1. fastLink
2. dblink
3. fastLink + dblink
4. Results:
   - Validation Study: `RLdata10000`
   - Empirical Application: National Long Term Care Study (`NLTCS`)

`fastLink-dblink`

- We propose a two-stage approach that combines the strengths of `fastLink` and `dblink` where:

    1. Instead of resorting to traditional blocking strategies we use `fastLink` to look for potential matches

    2. We use `dblink` to determine the co-reference structure of the linkages

- `fastLink` provides fast blocking and `dblink` provides exact uncertainty propagation

Road to Improving Probabilistic Entity Resolution

1. fastLink
2. dblink
3. fastLink + dblink
4. Results:
   - Validation Study: `RLdata10000`
   - Empirical Application: National Long Term Care Study (`NLTCS`)

RLdata10000

- Validation based on RLdata10000 from the R-package RecordLinkage

- It contains 10% duplicates

- Linkage fields include: first and last name; day, month, and year of birth

- For fastLink-dblink we consider two blocking approaches: loose (FDR < 10%) and strict (FDR < 1%)

Results

Table 1: Comparison of Matching Quality. “ARI” stands for adjusted Rand index and “Err. # clust.” is the percentage error in the number of clusters.

Dataset	Method	Pairwise measures			Cluster measures	
		Precision	Recall	F1-score	ARI	Err. # clust.
RLdata10000	dblink	0.63	1.00	0.78	0.78	−10.97%
	fastLink-L	0.94	0.98	0.96	—	—
	fastLink-S	0.96	0.98	0.97	—	—
	fastLink-dblink-L	0.94	1.00	0.97	0.97	-0.34%
	fastLink-dblink-S	0.96	1.00	0.98	0.98	-0.17%

Road to Improving Probabilistic Entity Resolution

1. fastLink
2. dblink
3. fastLink + dblink
4. Results:
    - Validation Study: `RLdata10000`
    - Empirical Application: National Long Term Care Study (`NLTCS`)

NLTCS

- NLTCS is a longitudinal study of health and well-being of those in the U.S. older than 65
- 3 waves, which account for 57,077 observations
- Hard test: data has been anonymized. Can we recover the true linkage structure?
- Linkage fields: full date of birth, state, location of doctor's office, and gender
- The number of unique individuals in the data is 34,945

Results

Table 2: Comparison of matching quality. “ARI” stands for adjusted Rand index and “Err. # clust.” is the percentage error in the number of clusters.

Dataset	Method	Pairwise measures			Cluster measures	
		Precision	Recall	F1-score	ARI	Err. # clust.
NLTCS	dblink	0.83	0.91	0.87	0.87	−22.09%
	fastLink-L	0.80	0.91	0.80	—	—
	fastLink-S	0.91	0.91	0.91	—	—
	fastLink-dblink-L	0.87	0.94	0.90	0.90	-13.01%
	fastLink-dblink-S	0.91	1.00	0.95	0.95	2.79%

Concluding Remarks

- Motivated by the need to scale ER to large data sets (in fast and accurate ways) we have proposed a two-stage approach to blocking and ER.
- `fastLink-dblink` is an open-source pipeline that could be useful for applied researchers on how to improve their own
- There are lots of opportunities to improve upon `fastLink-dblink` e.g., automating the pipeline
- As always, we recommend caution when using ER methods so that personal privacy is never at risk.

Computational Complexity of `fastLink-dblink`

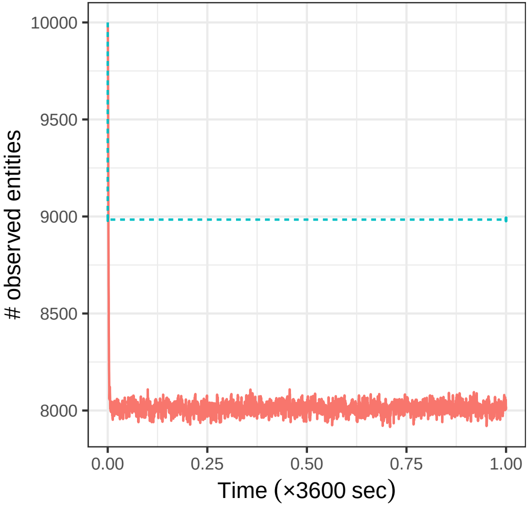
- Assume without loss of generality that each dataset is of equal size N
- Let N_{max} represent the total number of records in all databases
- Let $N^*_{max} \ll N_{max}$ is the number of records classified as possible matches by fastLink
- Let S_G denote the total number of MCMC iterations
- **Theorem:** The computational complexity of `fastLink-dblink` is

$$O(\Upsilon N_{max}(N_{max} - 1)) + O(S_G N^*_{max}), \quad \text{where} \quad \Upsilon = \frac{\omega}{2Q}.$$

where ω represents the share of unique values per linkage field and Q the number of threads available on your computer

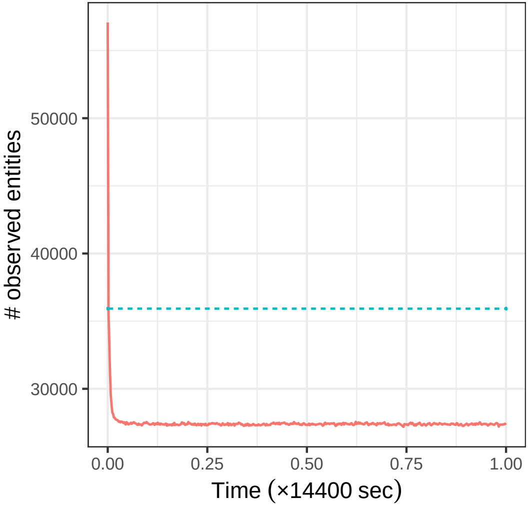
Appendix

- Convergence rates: Number of unique entities in `RLdata10000`



Appendix

- Convergence rates: Number of unique entities in `NLTCS`


# observed entities
Time (×14400 sec)