Efficient and Scalable Bayesian Bipartite Matching through Fast Beta Linkage (`fabl`)

Brian Kundinger, Jerome Reiter, Rebecca Steorts

Duke University

November 8, 2021

Table of Contents

What is Record Linkage?

- Record linkage is the task of identifying duplicate records over noisy datasets.
- Easy with unique identifiers, difficult when faced with errors
- **Bipartite matching** is the specific goal of matching one record in one dataset to most one match in another dataset

Record Linkage in Practice

Duke TODAY

search

Topics

PUBLISHED SEPTEMBER 17, 2018 IN RESEARCH

MAKING SENSE OF SYRIA'S MURKY DEATH TOLL

Record Linkage in Practice

DNC Announces New National Record Linkage System

APRIL 24, 2020

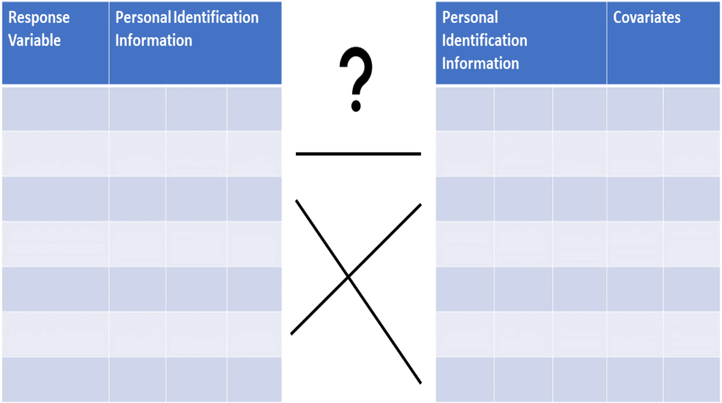
Algorithm developed by DNC expert in the field of record linkage will increase organizing efficiency by 9 percent and provide campaigns with more comprehensive view of the overall electorate

Linkage for Downstream Analysis

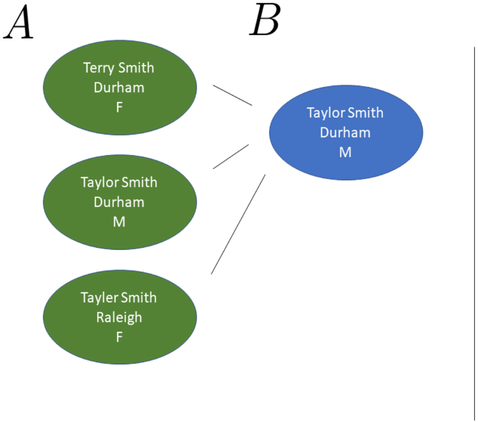
Response Variable	Personal Identification Information		

Personal Identification Information			Covariates	

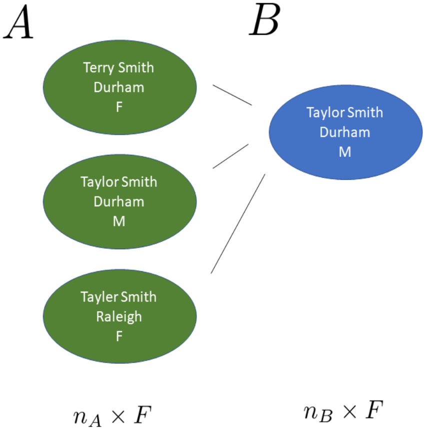
Linkage for Downstream Analysis



Linkage through Comparison Vectors

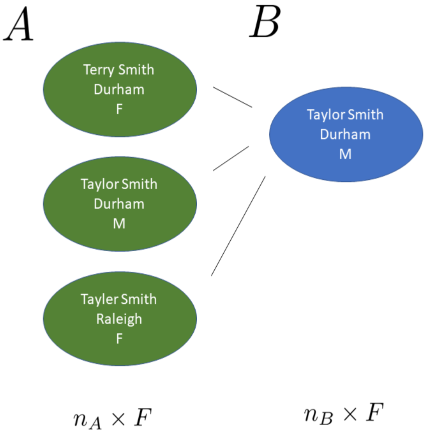


Linkage through Comparison Vectors



- n_A, n_B records in A, B
- $F = 4$ features for comparison
  - First name
  - Last name
  - City
  - Gender
- $L = \{3, 3, 2, 2\}$ levels of comparison

Linkage through Comparison Vectors



Γ

$$\gamma_{11} = [3, 1, 1, 2]$$

$$\gamma_{21} = [1, 1, 1, 1]$$

$$\gamma_{31} = [2, 1, 2, 2]$$

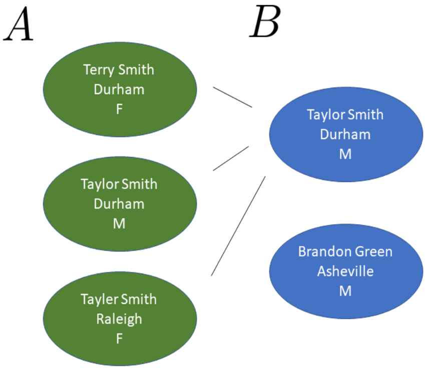
$n_A n_B \times F$

Linkage through Comparison Vectors

Represent linkage structure through vector $\mathbf{Z} = \{Z_1, \ldots, Z_{n_B}\}$, where

$$Z_j = \begin{cases} i, & \text{if records } i \in A \text{ and } j \in B \text{ match;} \\ n_A + 1, & \text{if record } j \in B \text{ has no match in } A; \end{cases}$$

Linkage through Comparison Vectors

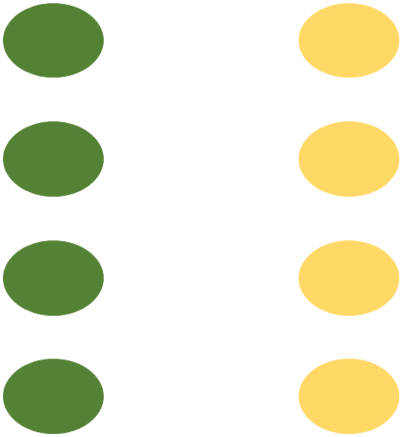


$n_A \times F$ $\quad$ $n_B \times F$

$$Z_1 = 2$$

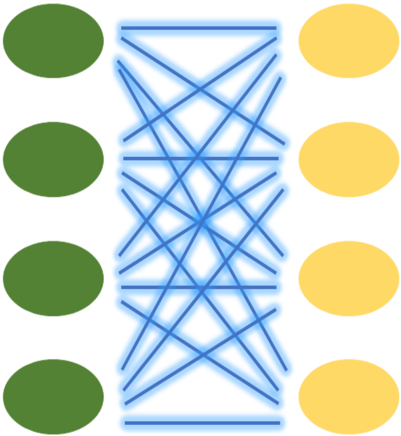
$$Z_2 = n_A + 1$$

Fellegi and Sunter (1969)



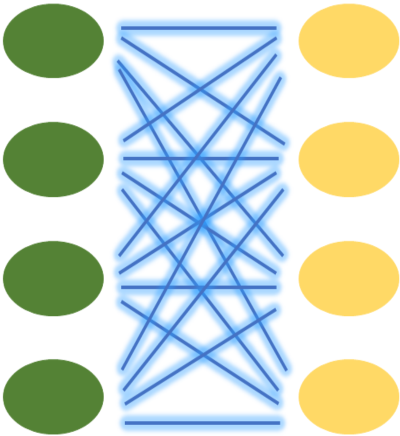
Fellegi and Sunter (1969)

$n_A n_B$ independent decisions



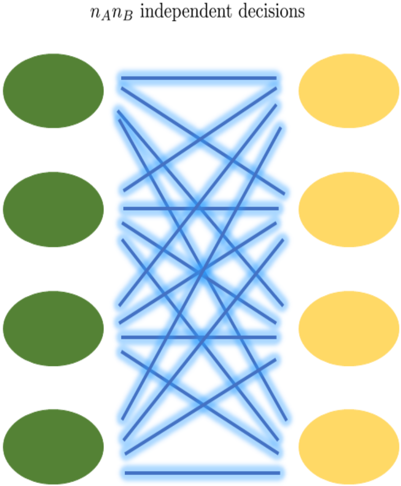
Fellegi and Sunter (1969)

$n_A n_B$ independent decisions



- scalable to large datasets (`fastlink`, Enamorado et al 2019)

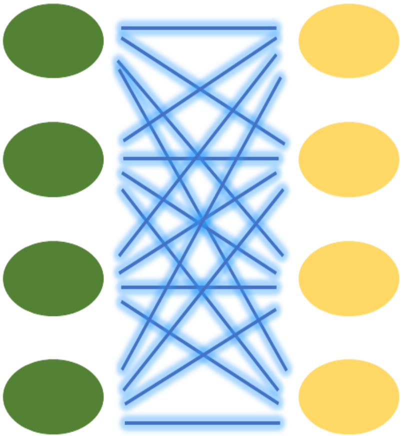
Fellegi and Sunter (1969)



- scalable to large datasets (`fastlink`, Enamorado et al 2019)
- not bipartite, requires post-processing

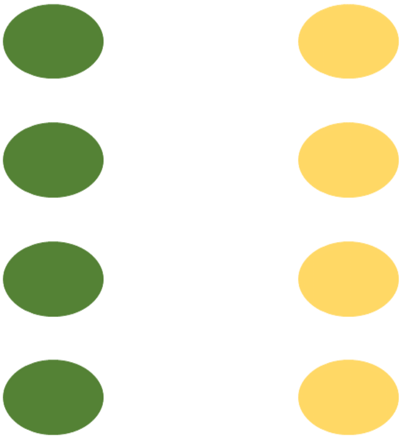
Fellegi and Sunter (1969)

$n_A n_B$ independent decisions



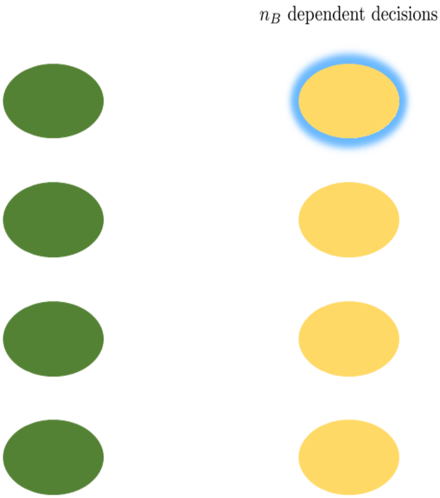
- scalable to large datasets (`fastlink`, Enamorado et al 2019)
- not bipartite, requires post-processing
- overmatches, leading to inaccurate parameter estimation

Sadinle (2017)



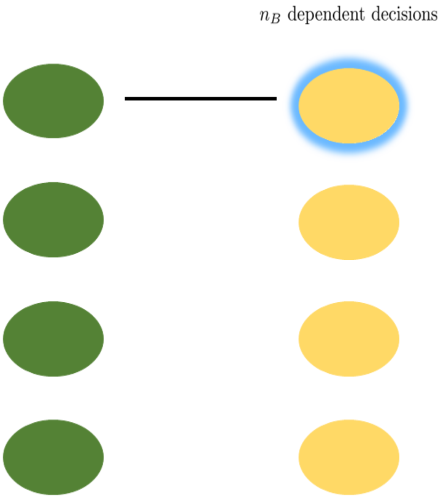
- Beta Record Linkage (BRL)

Sadinle (2017)



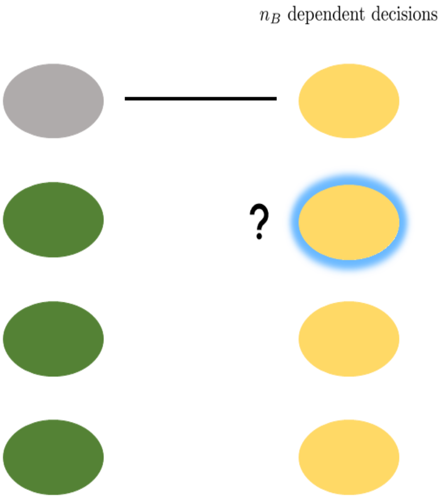
- Beta Record Linkage (BRL)

Sadinle (2017)



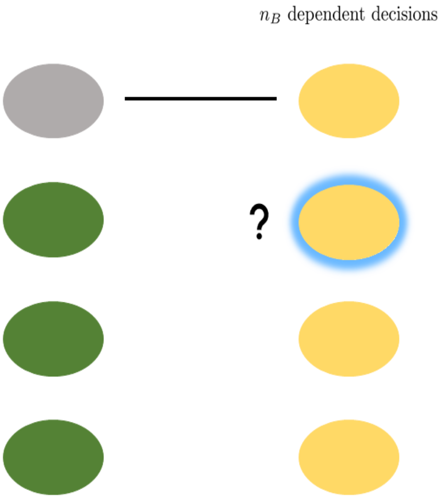
- Beta Record Linkage (BRL)

Sadinle (2017)



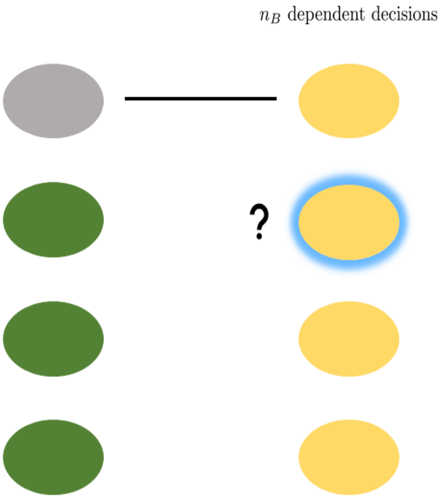
- Beta Record Linkage (BRL)

Sadinle (2017)



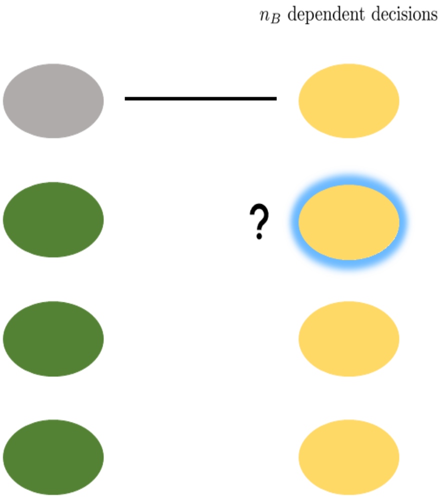
- Beta Record Linkage (BRL)
- strictly enforces one-to-one matching, no post-processing

Sadinle (2017)



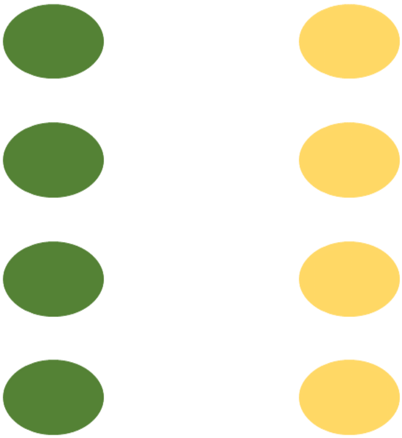
- Beta Record Linkage (BRL)
- strictly enforces one-to-one matching, no post-processing
- high accuracy for linkage and other parameters

Sadinle (2017)



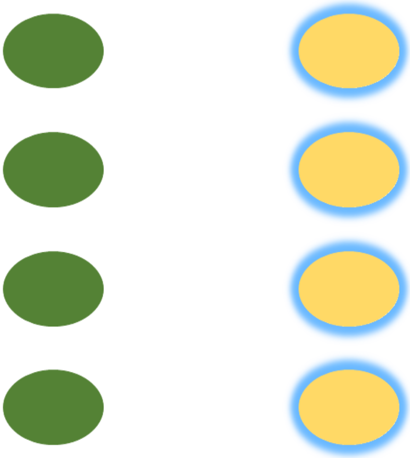
- Beta Record Linkage (BRL)
- strictly enforces one-to-one matching, no post-processing
- high accuracy for linkage and other parameters
- inherently serial, not scalable to large linkage tasks

Our Contribution - Fast Beta Linkage

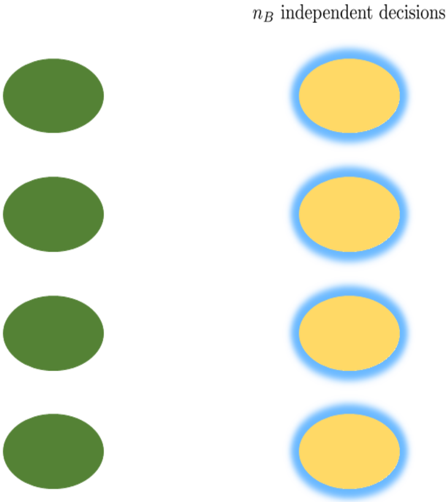


Our Contribution - Fast Beta Linkage

n_B independent decisions

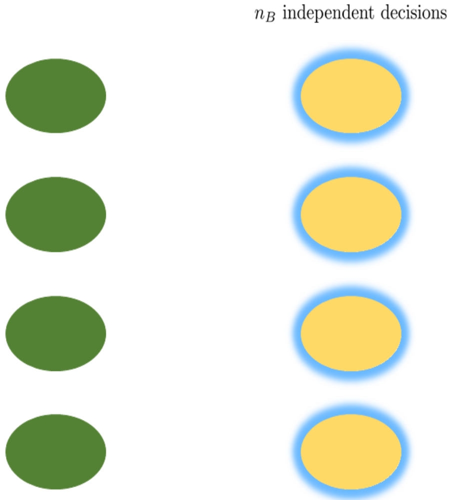


Our Contribution - Fast Beta Linkage



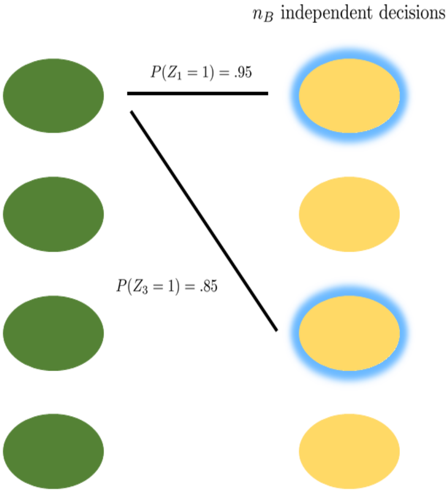
- relaxation proposed by Heck Wortman (2019)

Our Contribution - Fast Beta Linkage



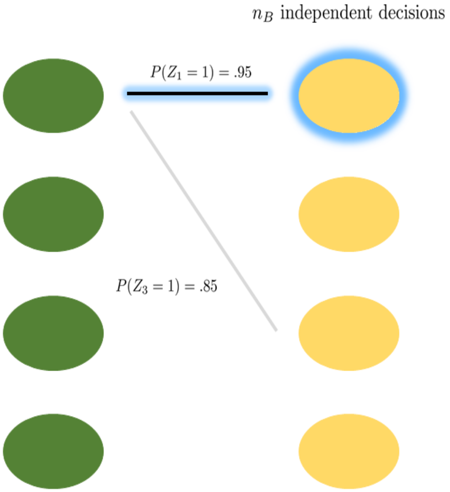
- relaxation proposed by Heck Wortman (2019)
- minimal loss of accuracy, large computational gains

Our Contribution - Fast Beta Linkage



- relaxation proposed by Heck Wortman (2019)
- minimal loss of accuracy, large computational gains
- allows for "one to many" matchings

Our Contribution - Fast Beta Linkage



- relaxation proposed by Heck Wortman (2019)
- minimal loss of accuracy, large computational gains
- allows for "one to many" matchings
- simple postprocessing to obtain bipartite matching

Table of Contents

Fast Beta Linkage (`fabl`)

$$P(\Gamma|\mathbf{Z},\mathbf{m},\mathbf{u}) = \prod_{j=1}^{n_B}\prod_{i=1}^{n_A}\left[\prod_{f=1}^{F}\prod_{l=1}^{L_f} m_{fl}^{I(Z_j=i)} u_{fl}^{I(Z_j\neq i)}\right]^{I(\gamma_{ij}^f=l)}$$

$$\mathbf{m_f} \sim \text{Dirichlet}(\alpha_{f1},\ldots,\alpha_{fL_f})$$

$$\mathbf{u_f} \sim \text{Dirichlet}(\beta_{f1},\ldots,\beta_{fL_f})$$

$$Z_j|\pi \begin{cases} \frac{\pi}{n_A} & z_j \leq n_A; \\ 1-\pi & z_j = n_A + 1 \end{cases}$$

$$\pi \sim \text{Beta}(\alpha_\pi, \beta_\pi)$$

Model specification allows for **hashing** of comparison vectors, **distributed** computing, and **storage efficient indexing (SEI)**

Hashing

- Recognize there are at most $P = \prod_{f=1}^{F} L_f$ unique agreement patterns, regardless of number of records (Enamorado et al 2019).

Hashing

- Recognize there are at most $P = \prod_{f=1}^{F} L_f$ unique agreement patterns, regardless of number of records (Enamorado et al 2019).
  - $L = \{3, 3, 2, 2\}$ implies 36 unique patterns

$\underline{p}$	$\underline{h_p}$
1	$[1, 1, 1, 1]$
2	$[1, 1, 1, 2]$
$\vdots$	$\vdots$
36	$[3, 3, 2, 2]$

Hashing

- Recognize there are at most $P = \prod_{f=1}^{F} L_f$ unique agreement patterns, regardless of number of records (Enamorado et al 2019).
    - $L = \{3, 3, 2, 2\}$ implies 36 unique patterns
- When (i, j) pair exhibits agreement pattern p, say $(i, j) \in h_p$.

$\underline{p}$	$\underline{h_p}$
1	$[1, 1, 1, 1]$
2	$[1, 1, 1, 2]$
$\vdots$	$\vdots$
36	$[3, 3, 2, 2]$

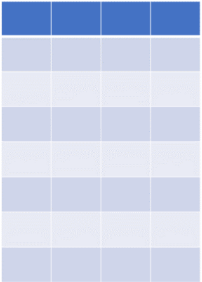
Hashing

- Recognize there are at most $P = \prod_{f=1}^{F} L_f$ unique agreement patterns, regardless of number of records (Enamorado et al 2019).
  - $L = \{3, 3, 2, 2\}$ implies 36 unique patterns
- When (i, j) pair exhibits agreement pattern p, say $(i, j) \in h_p$.
- Allows us to compute sufficient statistics and reduce computational complexity from $O(n_A \times n_B)$ to $O(P \times n_B)$

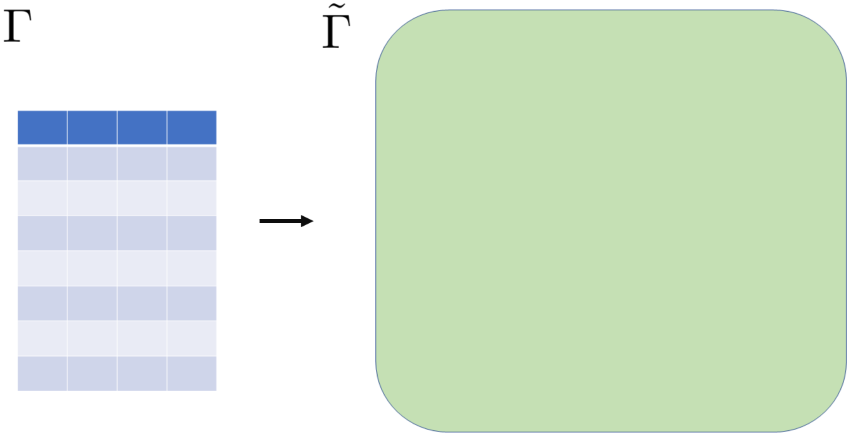
p	h_p
1	$[1, 1, 1, 1]$
2	$[1, 1, 1, 2]$
$\vdots$	$\vdots$
36	$[3, 3, 2, 2]$

Hashing

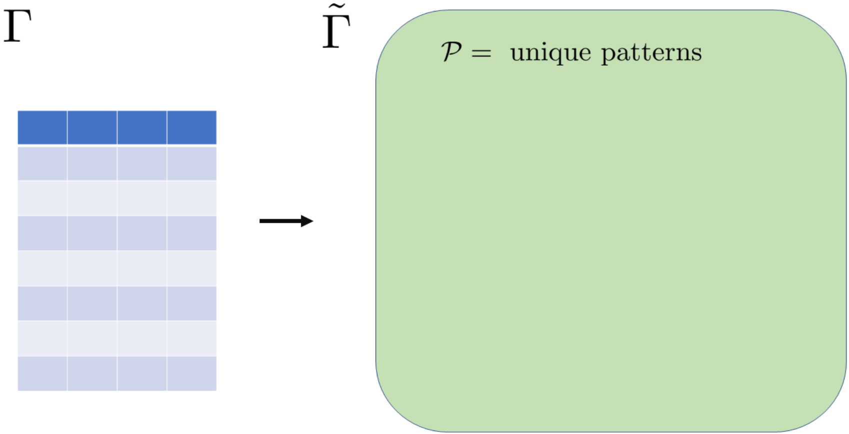
Γ



Hashing

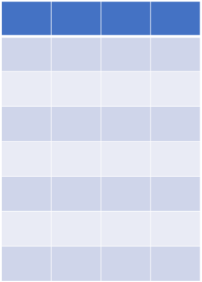


Hashing



Hashing

Γ



$\rightarrow$

$\tilde{\Gamma}$

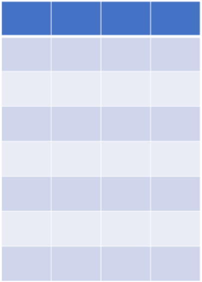
$\mathcal{P} =$ unique patterns

$\mathcal{R} = \left\{ \{ r_{j_p} \}_{p=1}^{P} \right\}_{j=1}^{n_2}$

$$\begin{aligned} r_{j_p} &= \{ i \in A \,|\, (i,j) \in h_p \} \\ &= \text{records in } A \text{ that share agreement} \\ &\quad \text{pattern } p \text{ with record } j \end{aligned}$$

Hashing

Γ



$\rightarrow$

$\tilde{\Gamma}$

$$\mathcal{P} = \text{ unique patterns}$$

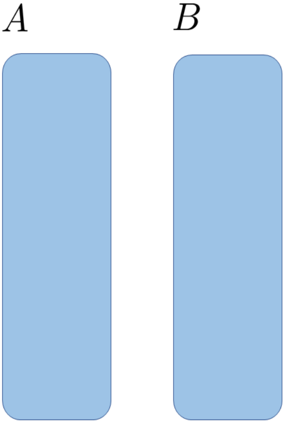
$$\mathcal{R} = \left\{ \{r_{j_p}\}_{p=1}^{P} \right\}_{j=1}^{n_2}$$

$$\begin{aligned} r_{j_p} &= \{i \in A | (i,j) \in h_p\} \\ &= \text{ records in } A \text{ that share agreement} \\ &\text{pattern } p \text{ with record } j \end{aligned}$$

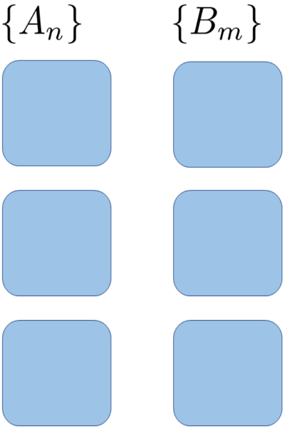
$$\mathcal{H} = \left\{ \{H_{j_p}\}_{p=1}^{P} \right\}_{j=1}^{n_2}$$

$$\begin{aligned} H_{j_p} &= ||r_{j_p}|| = \sum_{i=1}^{n_A} \mathbf{1}_{(i,j) \in h_p} \\ &= \text{ number of records in } A \\ &\text{that share agreement pattern } p \text{ with record } j \end{aligned}$$

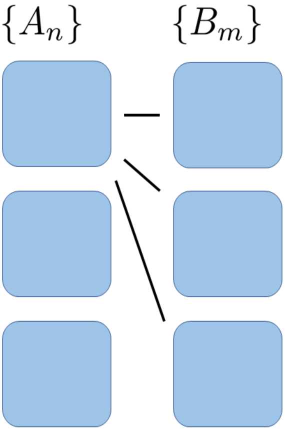
Hashing to Scale to Large Linkage Tasks



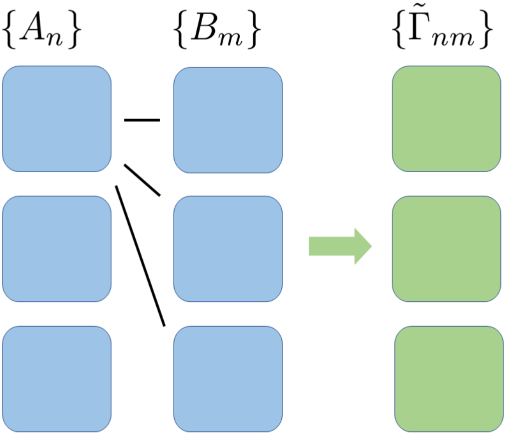
Hashing to Scale to Large Linkage Tasks



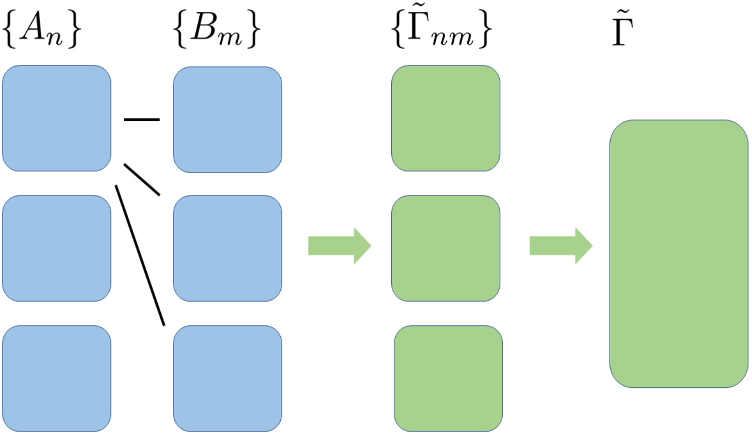
Hashing to Scale to Large Linkage Tasks



Hashing to Scale to Large Linkage Tasks



Hashing to Scale to Large Linkage Tasks



Hashing for Speed (Efficient Gibbs Sampling)

Sample by record

$$Z_j | \Phi, \Gamma, \pi \propto \begin{cases} \frac{\pi}{n_A} w_{ij} & z_j \leq n_A; \\ 1 - \pi & z_j = n_A + 1 \end{cases}$$

$$\Phi = \{\mathbf{m}, \mathbf{u}\}$$

$$w_{ij} = \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}} \right)^{I(\gamma_{ij}^f = l)}$$

$$= \frac{P(\boldsymbol{\gamma}_{ij} | Z_j = i)}{P(\boldsymbol{\gamma}_{ij} | Z_j \neq i)}$$

Hashing for Speed (Efficient Gibbs Sampling)

Sample by record

$$Z_j|\Phi, \Gamma, \pi \propto \begin{cases} \frac{\pi}{n_A} w_{ij} & z_j \leq n_A; \\ 1-\pi & z_j = n_A + 1 \end{cases}$$

Complexity depends on n_A

$$\Phi = \{\mathbf{m}, \mathbf{u}\}$$

$$w_{ij} = \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}} \right)^{I(\gamma_{ij}^{f} = l)}$$

$$= \frac{P(\boldsymbol{\gamma_{ij}} | Z_j = i)}{P(\boldsymbol{\gamma_{ij}} | Z_j \neq i)}$$

Hashing for Speed (Efficient Gibbs Sampling)

Sample by record

$$Z_j|\Phi,\Gamma,\pi \propto \begin{cases} \frac{\pi}{n_A} w_{ij} & z_j \leq n_A; \\ 1-\pi & z_j = n_A + 1 \end{cases}$$

Complexity depends on n_A

Sample by agreement pattern

$$h\left(Z_j\right) \mid \Phi, \tilde{\Gamma}, \pi \propto \begin{cases} \frac{\pi}{n_A} w_p \times H_{j_p} & p \leq P; \\ 1-\pi & p = P+1 \end{cases}$$

$$\Phi = \{\mathbf{m}, \mathbf{u}\}$$

$$\begin{aligned} w_{ij} &= \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}}\right)^{I(\gamma_{ij}^f = l)} \\ &= \frac{P(\boldsymbol{\gamma}_{ij}|Z_j = i)}{P(\boldsymbol{\gamma}_{ij}|Z_j \neq i)} \end{aligned}$$

Hashing for Speed (Efficient Gibbs Sampling)

Sample by record

$$Z_j|\Phi,\Gamma,\pi \propto \begin{cases} \frac{\pi}{n_A} w_{ij} & z_j \leq n_A; \\ 1-\pi & z_j = n_A + 1 \end{cases}$$

Complexity depends on n_A

$\Rightarrow$

Sample by agreement pattern

$$h(Z_j) \mid \Phi, \tilde{\Gamma}, \pi \propto \begin{cases} \frac{\pi}{n_A} w_p \times H_{j_p} & p \leq P; \\ 1-\pi & p = P+1 \end{cases}$$

Complexity *does not* depend on n_A

$$\Phi = \{\mathbf{m}, \mathbf{u}\}$$

$$\begin{aligned} w_{ij} &= \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}} \right)^{I(\gamma_{ij}^f = l)} \\ &= \frac{P(\boldsymbol{\gamma}_{ij}|Z_j = i)}{P(\boldsymbol{\gamma}_{ij}|Z_j \neq i)} \end{aligned}$$

Hashing for Speed (Efficient Gibbs Sampling)

Sample by record

$$Z_j|\Phi,\Gamma,\pi \propto \begin{cases} \frac{\pi}{n_A} w_{ij} & z_j \leq n_A; \\ 1-\pi & z_j = n_A + 1 \end{cases}$$

Complexity depends on n_A

$\Rightarrow$

Sample by agreement pattern

$$h(Z_j) \mid \Phi, \tilde{\Gamma}, \pi \propto \begin{cases} \frac{\pi}{n_A} w_p \times H_{j_p} & p \leq P; \\ 1-\pi & p = P+1 \end{cases}$$

Complexity *does not* depend on n_A

Sample by record *given* pattern

$$Z_j \mid h(Z_j) \propto \begin{cases} 1 & i \in r_{j_p} \\ 0 & \text{otherwise} \end{cases}$$

$$\Phi = \{\mathbf{m}, \mathbf{u}\}$$

$$w_{ij} = \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}}\right)^{I(\gamma_{ij}^f = l)} = \frac{P(\boldsymbol{\gamma}_{ij}|Z_j = i)}{P(\boldsymbol{\gamma}_{ij}|Z_j \neq i)}$$

Hashing for Speed (Efficient Gibbs Sampling)

Sample by record

$$Z_j|\Phi,\Gamma,\pi \propto \begin{cases} \frac{\pi}{n_A} w_{ij} & z_j \le n_A; \\ 1-\pi & z_j = n_A + 1 \end{cases}$$

Complexity depends on n_A

$\Rightarrow$

Sample by agreement pattern

$$h(Z_j) \mid \Phi, \tilde{\Gamma}, \pi \propto \begin{cases} \frac{\pi}{n_A} w_p \times H_{j_p} & p \le P; \\ 1-\pi & p = P+1 \end{cases}$$

Complexity *does not* depend on n_A

Sample by record *given* pattern

$$Z_j \mid h(Z_j) \propto \begin{cases} 1 & i \in r_{j_p} \\ 0 & \text{otherwise} \end{cases}$$

Complexity *does not* depend on n_A

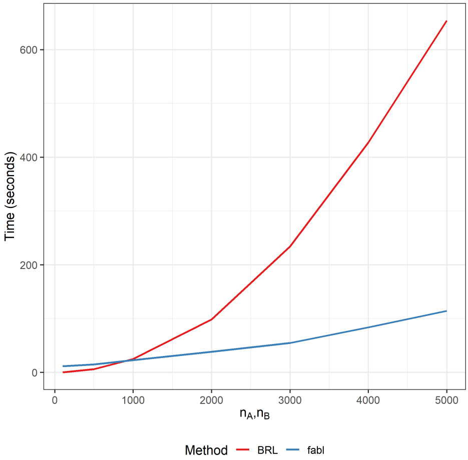
$$\Phi = \{\mathbf{m}, \mathbf{u}\}$$

$$\begin{aligned} w_{ij} &= \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}}\right)^{I(\gamma_{ij}^{f}=l)} \\ &= \frac{P(\boldsymbol{\gamma}_{ij}|Z_j = i)}{P(\boldsymbol{\gamma}_{ij}|Z_j \ne i)} \end{aligned}$$

Table of Contents

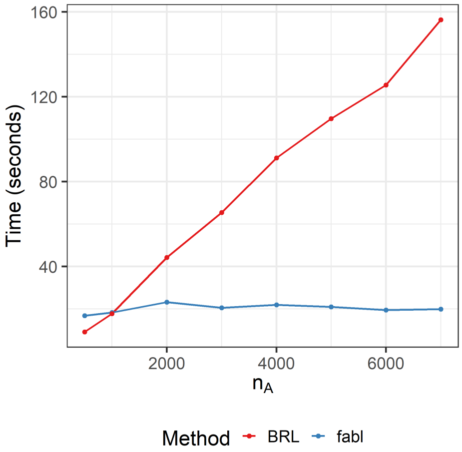
Speed Simulation 1

- $F = 5$ comparison fields
- $\mathrm{L} = \{2, 2, 2, 2, 2\}$, all binary comparisons
- 32 possible patterns
- Increase both n_A and n_B



Speed Simulation 2

- $F = 5$ comparison fields
- $L = \{2, 2, 2, 2, 2\}$, all binary comparisons
- 32 possible patterns
- Fix $n_B = 500$, increase n_A



Accuracy Simulation

- Sadinle (2017) used 900 simulated linkage tasks to show accuracy of `BRL`
- Find matches across two datasets, each with 500 records and 4 fields in common.
- One, two or three errors across matching records
- 10% matching, 50% matching, or 90% matching
- Calculate recall, precision, and F-measure

Accuracy Simulation

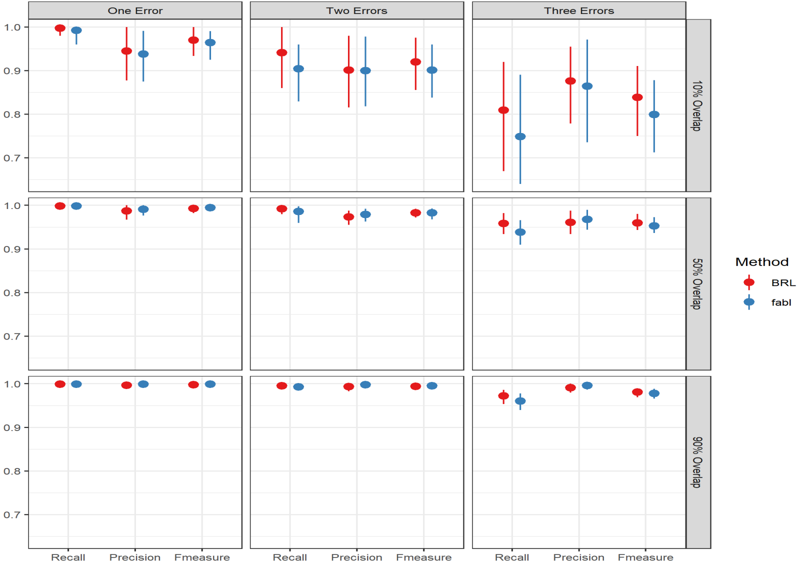


Table of Contents

Benefits of `fabl`

- Faster computation for larger linkage tasks
- Accurate estimation of linkage structure $\mathbf{Z}$, and additional parameters $\mathbf{m}$ and $\mathbf{u}$
- Bayesian model with natural uncertainty quantification

Extensions and Future Directions

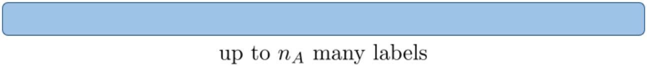
- Linkage when reliability of information and rates of matching differs by subgroup in the data
- Linkage when there are duplicates within datasets
- Linkage over blocked data (allows for much larger linkage tasks)

Table of Contents

Storage Efficient Indexing

- Storing $n_A \times n_B$ hashed values burdensome for large tasks
- Overwhelming majority of record pairs show nonagreement

$r_{j_p} =$



Storage Efficient Indexing

- Storing $n_A \times n_B$ hashed values burdensome for large tasks
- Overwhelming majority of record pairs show nonagreement

$r_{j_p} =$

up to n_A many labels

$r_{j_p}^{\mathrm{SEI}} =$ at most S many labels

Storage Efficient Indexing

- Storing $n_A \times n_B$ hashed values burdensome for large tasks
- Overwhelming majority of record pairs show nonagreement
- Correct counts for calculations stored in $\mathcal{H}$

$r_{j_p} =$

up to n_A many labels

$r_{j_p}^{\mathrm{SEI}} =$ at most S many labels