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

Google Version

Brian Kundinger, Jerome Reiter, Rebecca Steorts

Duke University

June 7, 2022

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

DNC Announces New National Record Linkage System

APRIL 24, 2020

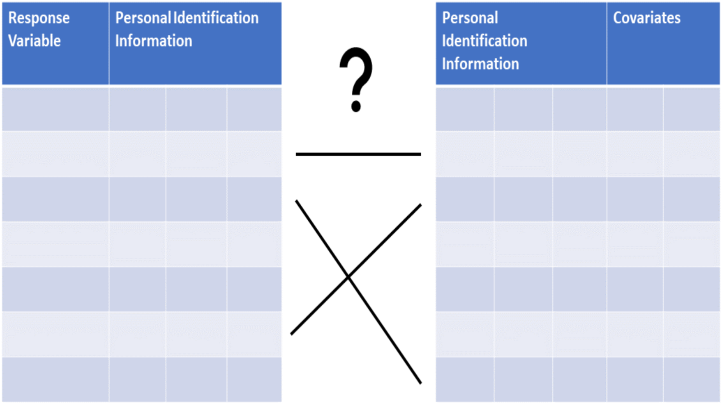
Algorithm developed by DNC expert in the field of record linkage will increase organizing efficiency by 9 percent and provide campaigns with more comprehensive view of the overall electorate

Linkage for Downstream Analysis

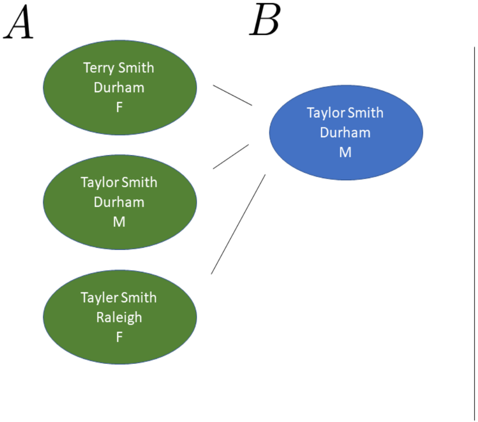
Response Variable	Personal Identification Information		

Personal Identification Information			Covariates	

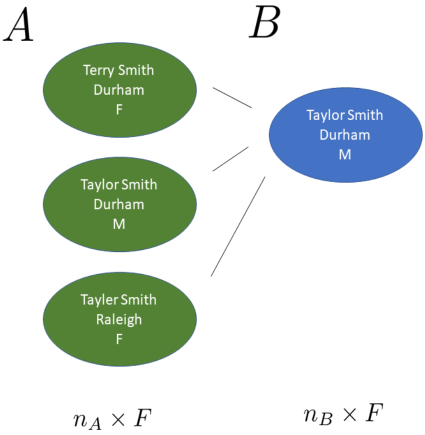
Linkage for Downstream Analysis



Linkage through Comparison Vectors

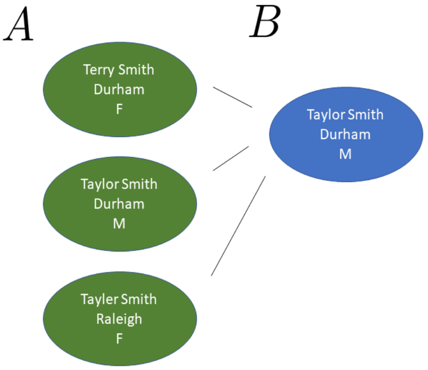


Linkage through Comparison Vectors



- 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



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

Γ

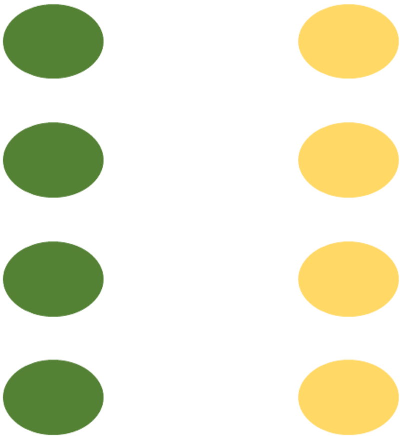
$$\gamma_{11} = [3, 1, 1, 2]$$

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

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

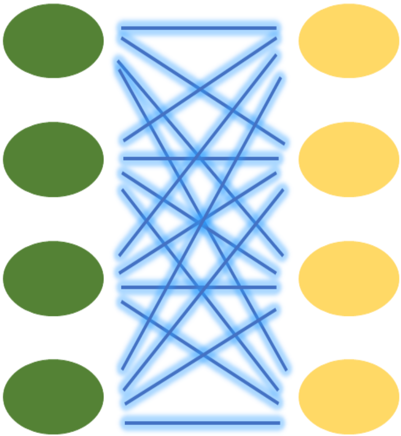
$n_A n_B \times F$

Fellegi and Sunter (1969)

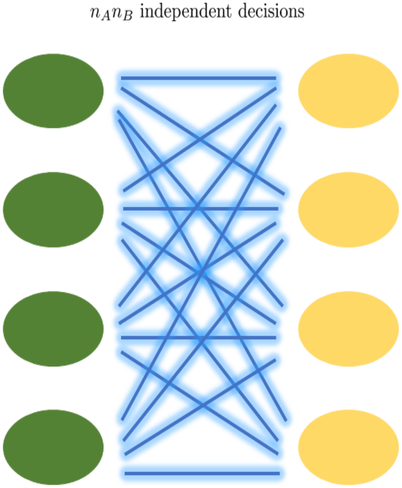


Fellegi and Sunter (1969)

$n_A n_B$ independent decisions

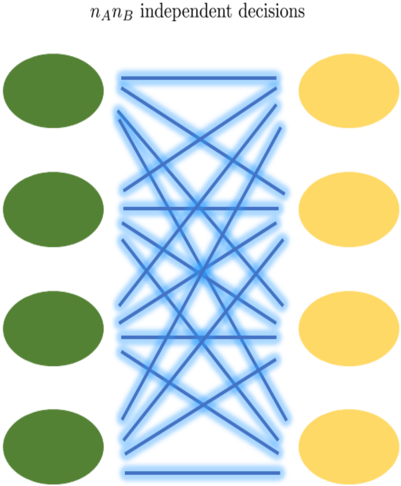


Fellegi and Sunter (1969)



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

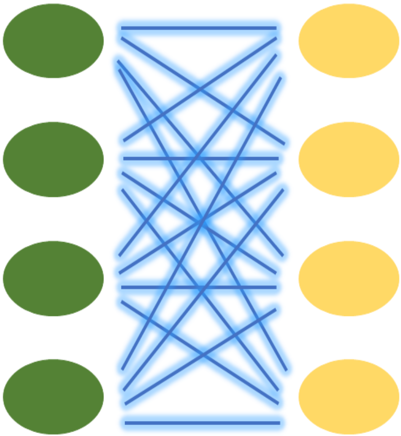
Fellegi and Sunter (1969)



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

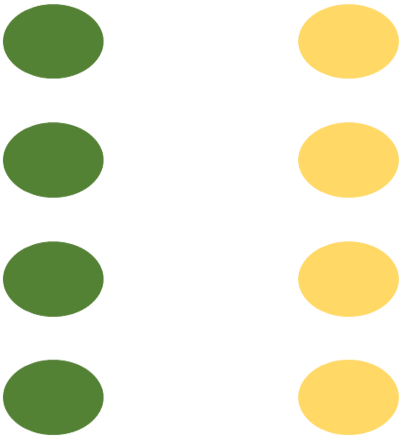
Fellegi and Sunter (1969)

$n_A n_B$ independent decisions



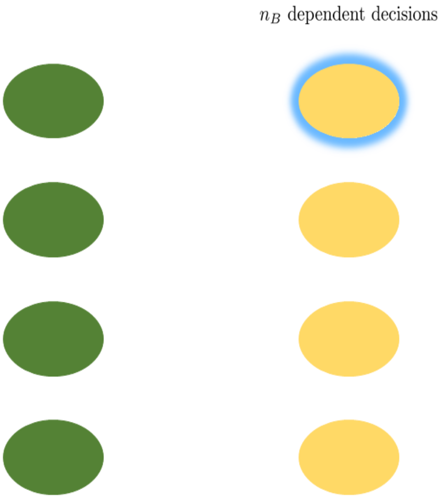
- scalable to large datasets (`fastlink`, Enamorado et al 2019)
- not bipartite, requires post-processing
- overmatches, leading to inaccurate parameter estimation

Sadinle (2017)



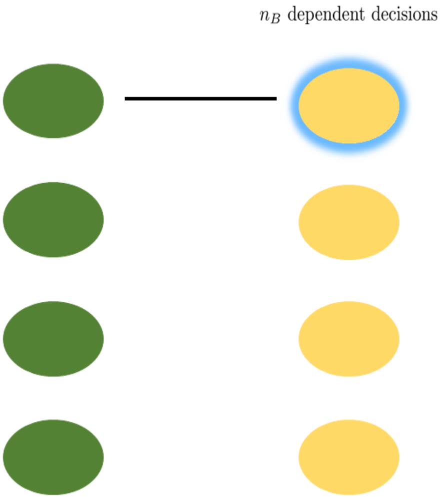
- Beta Record Linkage (BRL)

Sadinle (2017)



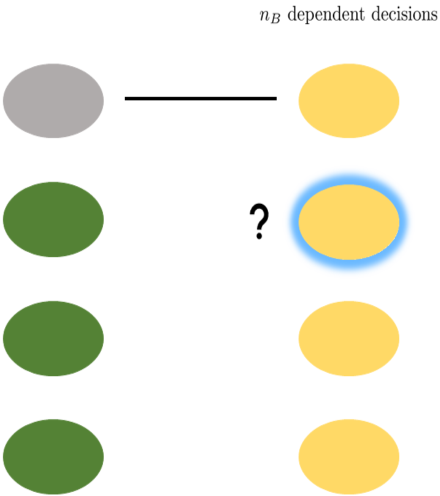
- Beta Record Linkage (BRL)

Sadinle (2017)



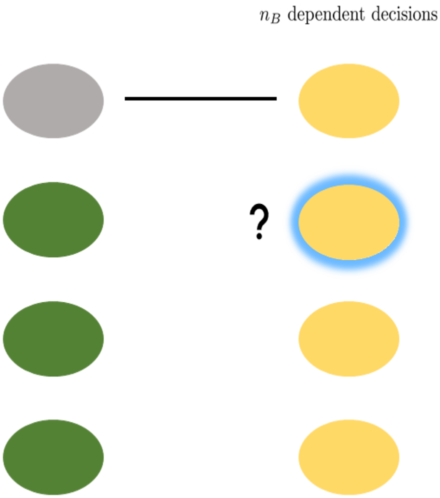
- Beta Record Linkage (BRL)

Sadinle (2017)



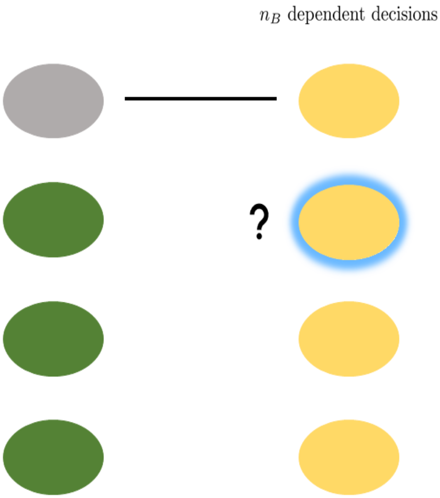
- Beta Record Linkage (BRL)

Sadinle (2017)



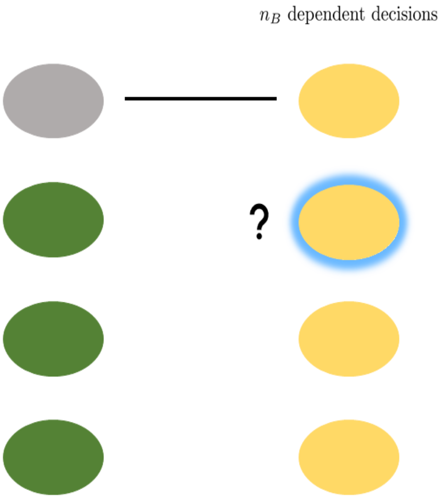
- Beta Record Linkage (BRL)
- strictly enforces one-to-one matching, no post-processing

Sadinle (2017)



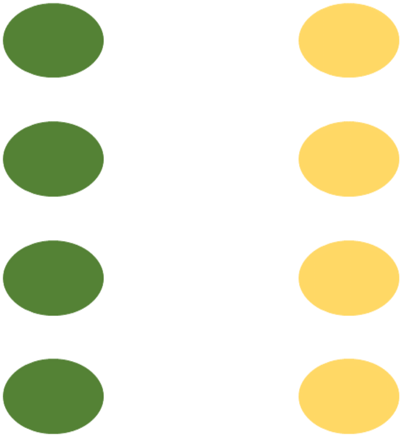
- Beta Record Linkage (BRL)
- strictly enforces one-to-one matching, no post-processing
- high accuracy for linkage and other parameters

Sadinle (2017)

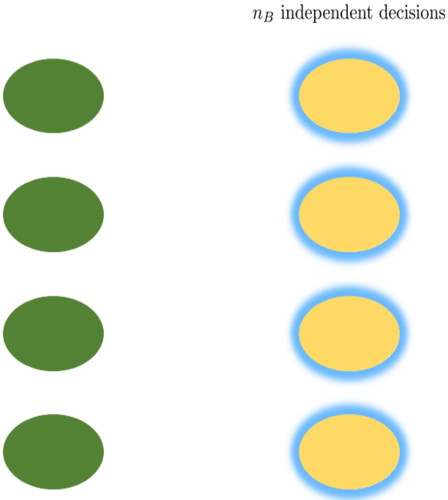


- Beta Record Linkage (BRL)
- strictly enforces one-to-one matching, no post-processing
- high accuracy for linkage and other parameters
- inherently serial, not scalable to large linkage tasks

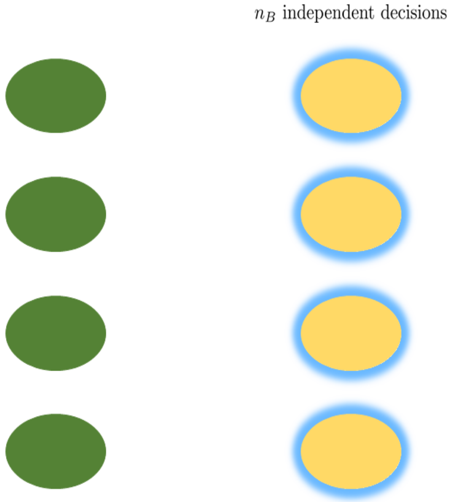
Our Contribution - Fast Beta Linkage



Our Contribution - Fast Beta Linkage

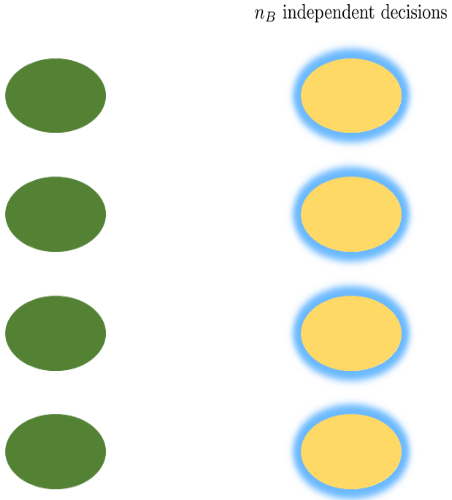


Our Contribution - Fast Beta Linkage



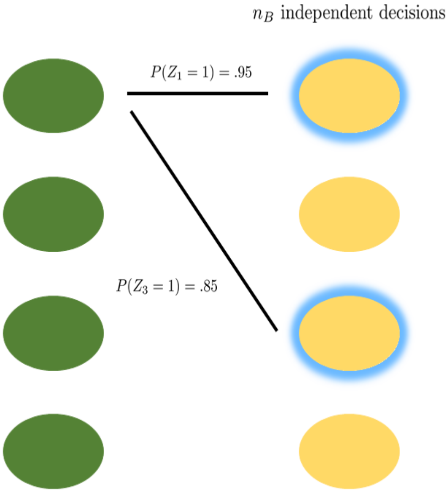
- relaxation proposed by Heck Wortman (2019)

Our Contribution - Fast Beta Linkage



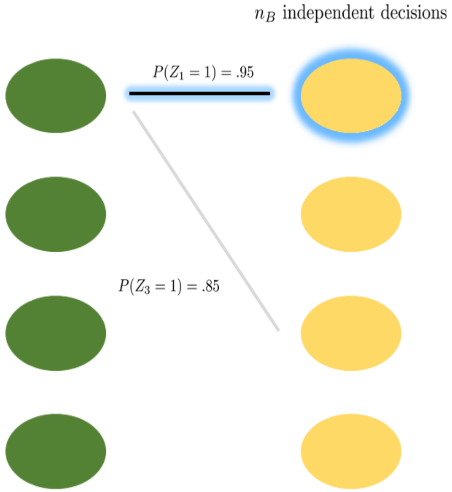
- relaxation proposed by Heck Wortman (2019)
- minimal loss of accuracy, large computational gains

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`)

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

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

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

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

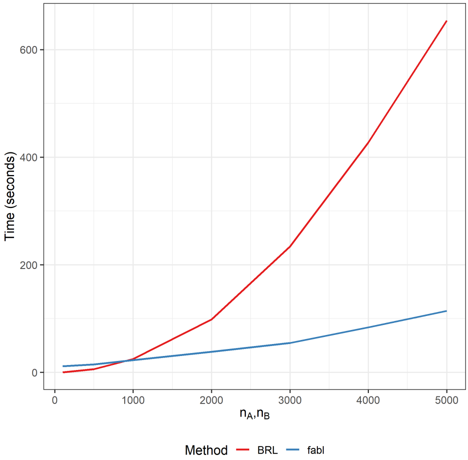
$$\pi \sim \text{Beta}(\alpha_\pi, \beta_\pi)$$

Note: the first three lines are common to many record linkage models, and bear similarities to a larger family of *latent class models.* The last two lines are the newly developed prior distribution for the set of matches.

Table of Contents

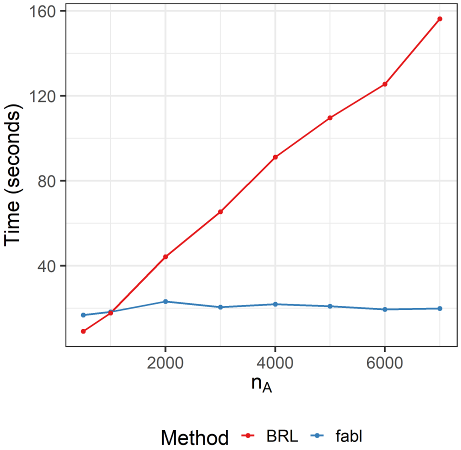
Speed Simulation 1

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



Speed Simulation 2

- $F = 5$ comparison fields
- $\mathrm{L} = \{2, 2, 2, 2, 2\}$, 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

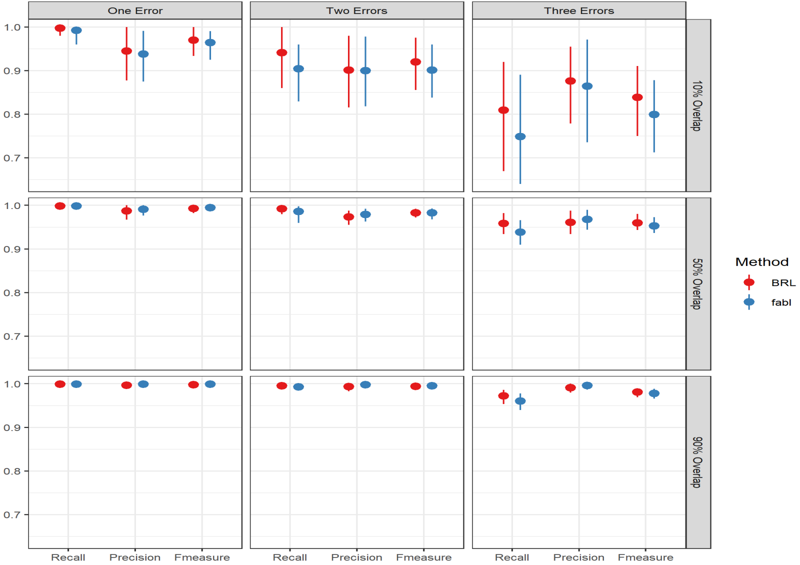


Table of Contents

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

Glaring Questions

- Can you do record linkage with arbitrary amounts of duplicated files?
- Isn't $n_A \times n_B$ record pairs computationally infeasible?
- Can you account situations when the reliability of information differs throughout the data?
- Can you do record linkage on the records themselves, rather than transforming to comparison vectors?

Glaring Questions

- Can you do record linkage with arbitrary amounts of duplicated files?
  - Yes! See Aleshin-Guendel and Sadinle (2021)
- Isn't $n_A \times n_B$ record pairs computationally infeasible?
- Can you account situations when the reliability of information differs throughout the data?
- Can you do record linkage on the records themselves, rather than transforming to comparison vectors?

Glaring Questions

- Can you do record linkage with arbitrary amounts of duplicated files?
  - Yes! See Aleshin-Guendel and Sadinle (2021)
- Isn't $n_A \times n_B$ record pairs computationally infeasible?
  - Yes! My work with the DNC is forthcoming, Kundinger, Wortman, Reiter, and Steorts (2022)
- Can you account situations when the reliability of information differs throughout the data?

- Can you do record linkage on the records themselves, rather than transforming to comparison vectors?

Glaring Questions

- Can you do record linkage with arbitrary amounts of duplicated files?
  - Yes! See Aleshin-Guendel and Sadinle (2021)
- Isn't $n_A \times n_B$ record pairs computationally infeasible?
  - Yes! My work with the DNC is forthcoming, Kundinger, Wortman, Reiter, and Steorts (2022)
- Can you account situations when the reliability of information differs throughout the data?
  - Yes! I have a heirarchical model in the works. Hopefully Kundinger et al (2023)
- Can you do record linkage on the records themselves, rather than transforming to comparison vectors?

Glaring Questions

- Can you do record linkage with arbitrary amounts of duplicated files?
  - Yes! See Aleshin-Guendel and Sadinle (2021)
- Isn't $n_A \times n_B$ record pairs computationally infeasible?
  - Yes! My work with the DNC is forthcoming, Kundinger, Wortman, Reiter, and Steorts (2022)
- Can you account situations when the reliability of information differs throughout the data?
  - Yes! I have a heirarchical model in the works. Hopefully Kundinger et al (2023)
- Can you do record linkage on the records themselves, rather than transforming to comparison vectors?
  - Yes! See `d-blink` package from Marchant et al (2021)

Table of Contents

Latent Class Analysis

The record linkage model presented is a special case of a "latent class model". In my work, we attempt to classify record pairs into two distinct classes: matching, and non-matching pairs.

One can imagine however, that we can look at articles instead of records, look at blocks of text instead of record pairs, and look at attributes of those blocks instead of comparison vectors. This may provide a strategy for **entirely unsupervised** byline detection.

Additionally, if model parameters are accurate, we can train the model on a small set of articles, and then apply those parameters out onto a larger set of articles, for nearly instantaneous latent class analysis.