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

Brian Kundinger

Introduction to Record Linkage

- Record linkage is the task of identifying duplicate records over noisy datasets.
- Easy with unique identifiers, difficult when faced with errors
- Far ranging applications in business, public health, and human rights

Examples

Monkey Cage • Analysis

Georgia's 'exact match' law could potentially harm many eligible voters.

Georgia gubernatorial candidates Stacey Abrams, left, and Brian Kemp on May 20 in Atlanta. (John Amis/AP)

By Ted Enamorado

October 20, 2018

DNC Announces New National Record Linkage System

APRIL 24, 2020

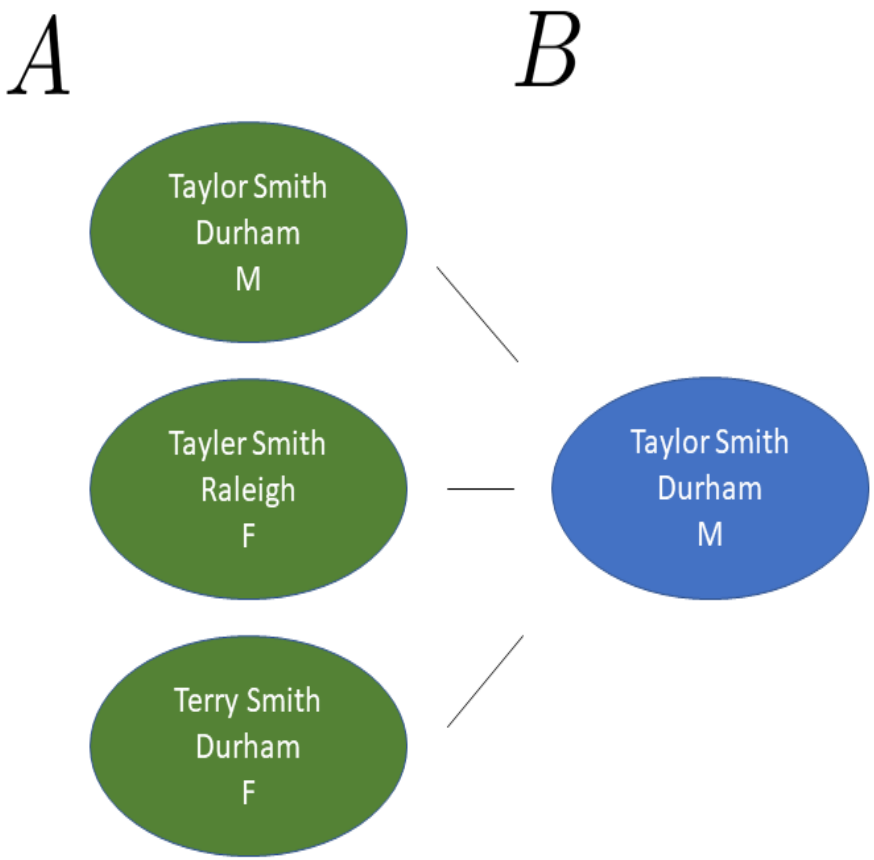
Algorithm developed by DNC expert in the field of record linkage will increase organizing efficiency by 9 percent and provide campaigns with more comprehensive view of the overall electorate

UNIQUE ENTITY ESTIMATION WITH APPLICATION TO THE SYRIAN CONFLICT

BY BEIDI CHEN[*,1], ANSHUMALI SHRIVASTAVA[*,2] AND REBECCA C. STEORTS[†,3]

Rice University[*] and Duke University[†]

Fellegi and Sunter



Γ

$$\gamma_{11} = \{1, 1, 1, 1\}$$

$$\gamma_{21} = \{2, 1, 2, 2\}$$

$$\gamma_{31} = \{3, 1, 1, 2\}$$

$n_A n_B \times F$

Previous work

- Fellegi and Sunter (1969)
  - independent classification of all $n_A n_B$ record pairs. Transitive closure acheived through postprocessing
- Enamorado et al (2019): `fastlink`
  - provides efficient and scalable FS model
- Sadinle (2017) - Beta Record Linkage (`BRL`)
  - Bayesian method for bipartite matching, strictly enforces one-to-one matching. Does not scale well to large linkage tasks

Notation

- File A with records indexed $i \in \{1, \ldots, n_A\}$ and file B with records $j \in \{1, \ldots, n_B\}$. We use F features for linkage, with L_f possible levels of agreement on feature f.
- $\Gamma \in \mathbb{R}^{n_A n_B \times F}$ matrix of comparison vectors where $\gamma_{ij}^f \in \{1, \ldots, L_f\}$
- $Z_j = \begin{cases} i, & \text{if records } i \in A \text{ and } j \in B \text{ match;} \\ n_A + 1, & \text{if record } j \in B \text{ has no match in } A; \end{cases}$
- $m_{fl} = P\left(\gamma_{ij}^f = l | Z_j = i\right)$
- $u_{fl} = P\left(\gamma_{ij}^f = l | Z_j \neq i\right)$
- $\lambda = P(Z_j \leq n_A)$

Fast Beta Linkage (fabl)

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

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

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

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

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

- Model specification allows for **parallel/distributed** computing, **hashing** of comparison vectors, 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. When (i, j) pair exhibits agreement pattern p, say $h(i, j) = p$.
- Reduce data to sufficient statistics
  - $r_{p_j} = \{i \mid (i, j) \in h_p\}$
  - $H_{p_j} = ||r_{p_j}||$
  - $H_p = \sum_j H_{p_j}$
- Run Gibbs sampler at level of agreement patterns, not record pairs
  - Sample the agreement of pattern $h(z_j, j)$, instead of record label z_j.
  - Use number of matches for each pattern to update m and u
  - Back fill record labels at the end through r_{p_j}
  - Computational complexity $O(P \times n_B \times F)$ instead of $O(n_A \times n_B \times F)$

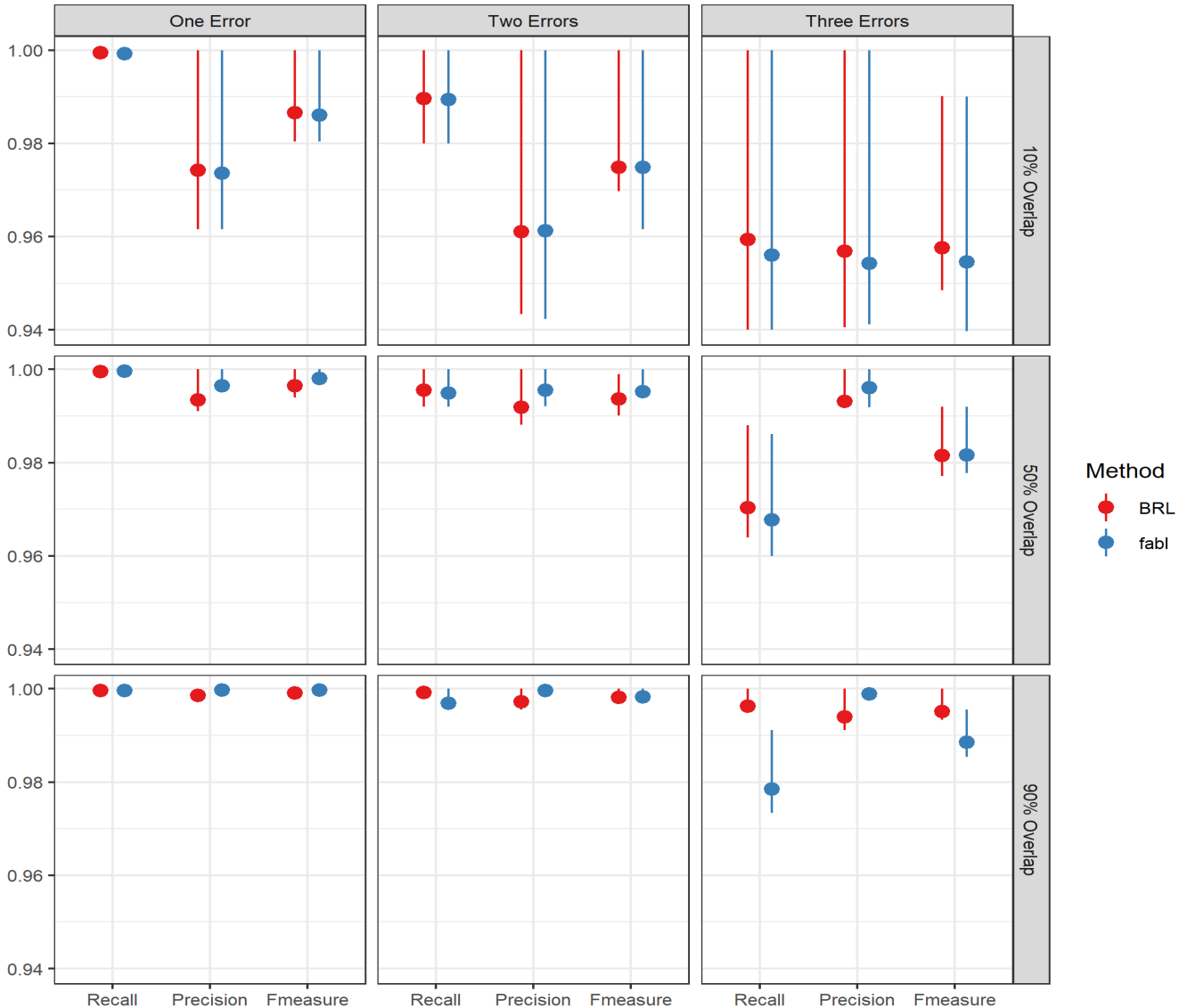
Managing Large Data

- **Distributed Computing** - Partition data in to chunks $\{A_I\}$ and $\{B_J\}$. Compare records, hash results, compute summary statistics in parallel, and synthesize results
- **Storage Efficient Indexing (SEI)** - Store at most small number R many record labels in each r_{p_j}, remove highly unlikely record labels from memory. Proper weights for calculations maintained through summary statistics $\{H_p\}$ and $\{H_{p_j}\}$.
- Hashing plus SEI can reduce memory requirements by $>99\%$.
  - Simulation of $20,000 \times 20,000$ linkage task with 4 fields. Naive approach requires 6.4GB of storage for all-to-all comparisons, hashing and SEI requires 90MB.

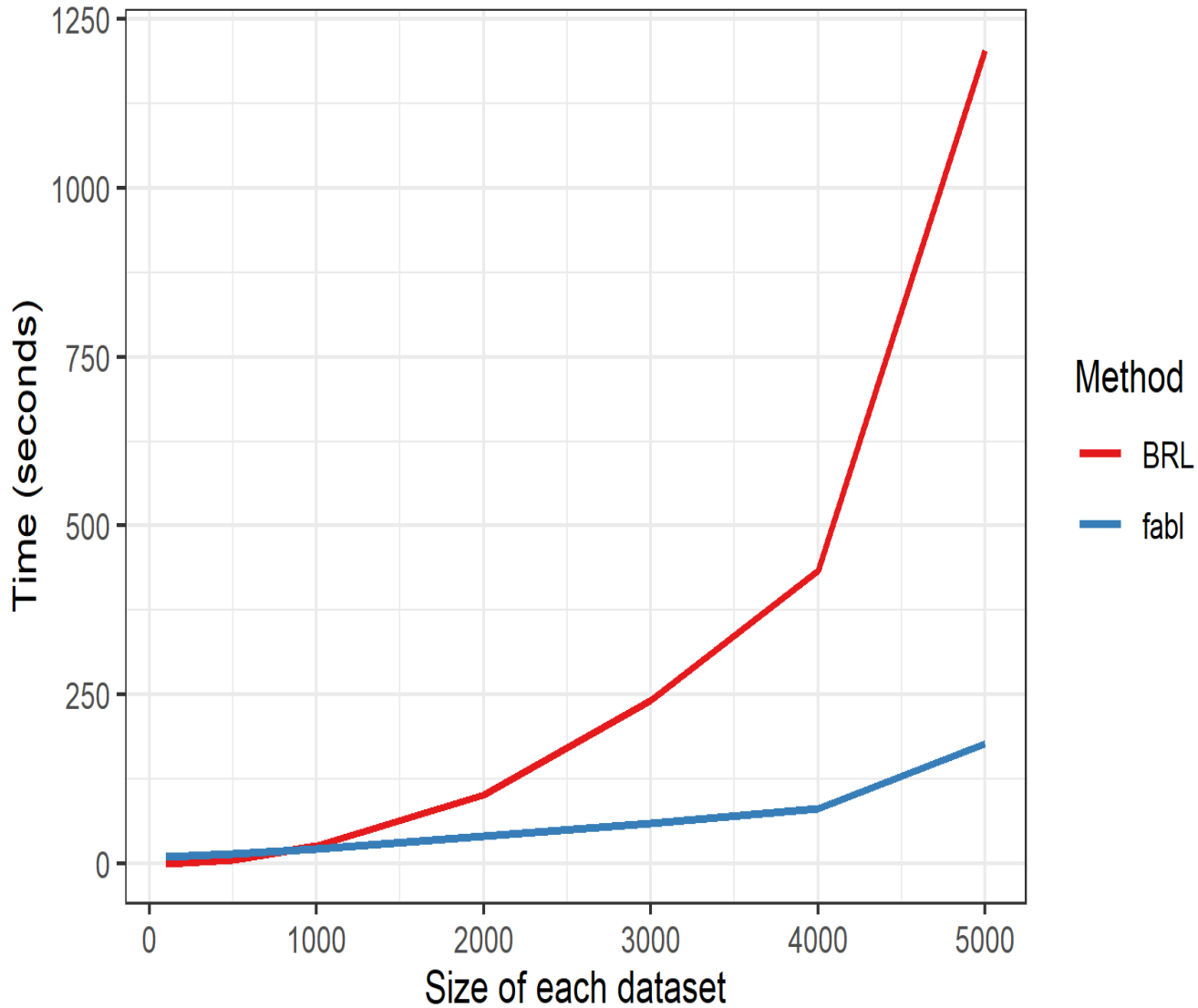
Three Simulations

- We compare `fabl` against `BRL` in three simulation studies
  - Measure precision and recall on 100 simulated datasets and varying levels of error and duplication across files
  - Measure speed when both n_A and n_B are increasing
  - Measure speed when n_A is increasing and $n_B = 500$ is fixed.

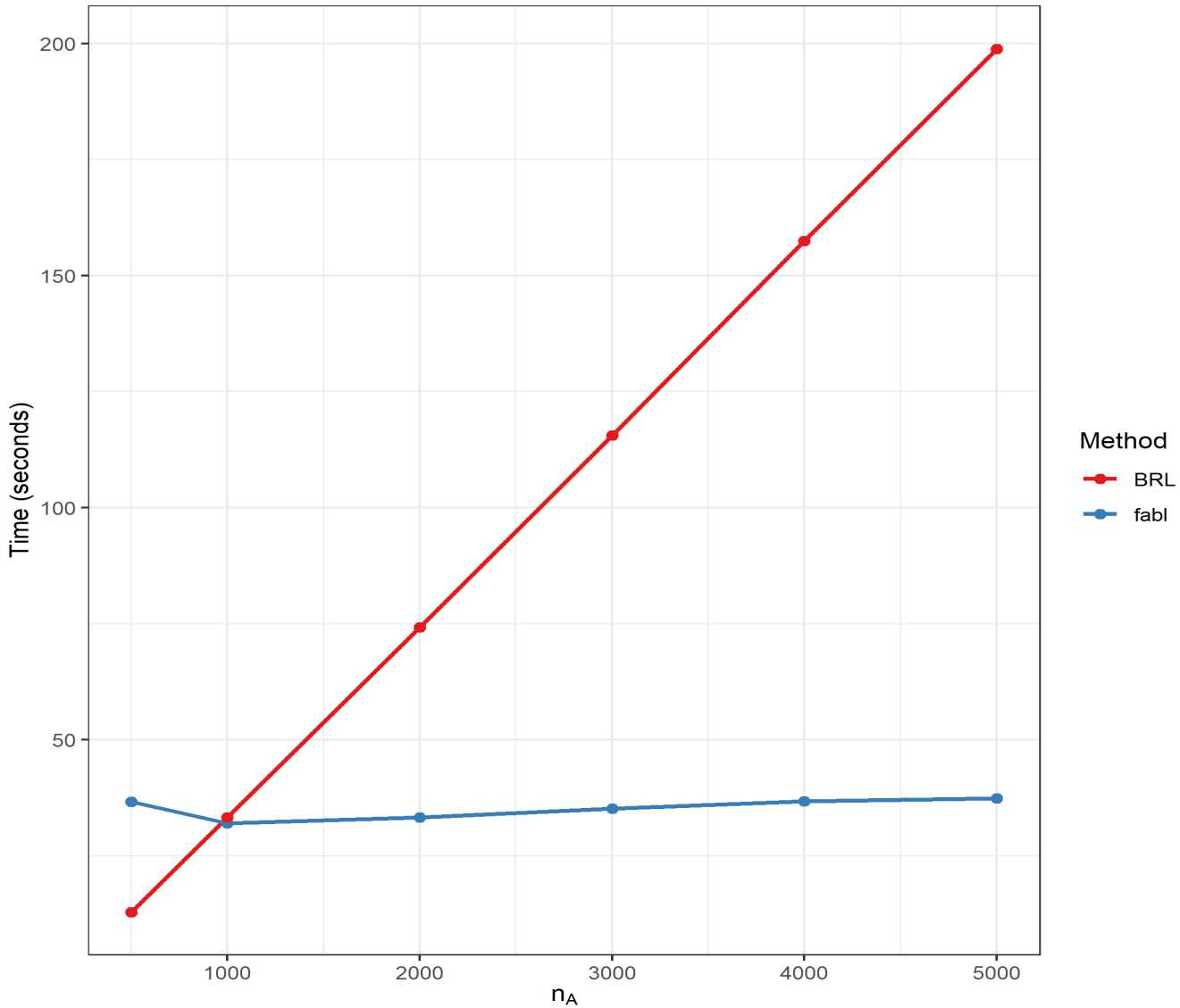
Simulation Results 1



Simulation Results 2



Simulation Results 3



Extentions and Directions

- Linkage in the face of duplicates within datasets
- Models that allow reliability of information to differ by subgroup in the data
- Linkage over blocked data (allows for much larger linkage tasks)

Questions?

Additional Slides

- Details of intuition behind m and u parameters
- Loss function and Bayes estimate
- Details of one-to-one matching; conflict resolution under `fabl`;
- El Salvador case study (time under different values of P, violations of one-to-one matching)
- Further details on distributed computation of comparison vectors and storage efficient indexing