Beta Linkage on Blocked Data

Brian Kundinger

Overview

- Big Picture
- Review of Base Model
- Modified Model for Blocking
- Challenges From Blocking
- Informative Priors
- "Train and Test" Method for Scalability

Big Picture

The biggest drawback to the Fellegi-Sunter approach of record linkage is the extremely large number of comparison vectors that must be created. This can be heavily reduced through blocking, but this blocking is not well addressed in the literature. Generally, one performs blocking on a reliable (usually categorical) field, and then applies a record linkage algorithm independently on each block of data.

Intensive Blocking

However, if you block on more discerning fields, like first and last name you end up with many more, and much smaller, blocks. It is often mathematically impossible to run independent record linkage over such small blocks.

(If a block consists of one Brian Kundinger in file A and one Brian Kundinger in file B, how can I separate pairs into matches and nonmatches? How could I estimate the $2\sum_{f=1}^{F} L_f + 1$ parameters of the model on only one pair?)

Previous approaches run the record linkage algorithm unmodified over this filtered data all at once (Wortman 2021, Sadinle 2014). However, we have seen poor estimation of u parameters, and thus poor overall performance.

Fast Beta Linkage

By now, we are all familiar with the model used in Fast Beta Linkage. For fields $f \in \{1, \ldots, F\}$ and levels $l \in \{1, \ldots, L_f\}$ we adopt the following likelihood and prior distributions.

$$P(\Gamma|\mathbf{Z}, \mathbf{m}, \mathbf{u}, \pi) = \prod_{j=1}^{n_B} \prod_{i=1}^{n_A} \mathbf{m}_{ij}^{\mathbf{1}_{z_j=i}} \mathbf{u}_{ij}^{\mathbf{1}_{z_j \neq i}}$$

$$\mathbf{m^f} \sim \text{Dirichlet}(\alpha^{f1}, \ldots, \alpha^{fL_f})$$

$$\mathbf{u^f} \sim \text{Dirichlet}(\beta^{f1}, \ldots, \beta^{fL_f})$$

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

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

Model Specification Under Blocking

Under blocking, the likelihood becomes

$$P(\Gamma|\mathbf{Z},\mathbf{m},\mathbf{u},\pi) = \prod_{j=1}^{n_B}\prod_{i\in N_j}\mathbf{m}_{ij}^{\mathbf{1}_{z_j=i}}\mathbf{u}_{ij}^{\mathbf{1}_{z_j\neq i}}$$

where N_j is the set of candidate records in A for record $j \in B$ after blocking. For the new likelihood, we need to modify the prior:

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

Gibbs Sampling

Under this new likelihood and prior, the full conditional for the components of the linkage parameter *mathbfZ* becomes

$$P\left(Z_j^{(s+1)} = i | \mathbf{m}, \mathbf{u}, \mathbf{Z^{(s)}}\right) \propto \begin{cases} w_{ij} & i \leq n_A; \\ N_j \frac{N_B - D + \beta_\pi}{D + \alpha_\pi} & i = n_A + 1 \end{cases}$$

where N_B is the number of records in the smaller dataset. Note here that N_j differs across the records in B. If you are blocking on first and last name and come across a rare name (Brian Kundinger), you'll have a small N_{BrianK}. However, for a more common name like Tom Smith, you'll have a large N_{TomS}.

Thus, records that have been more fine distinguished through blocking require less information in the other features in order to be deemed a match!

U-Estimation

We have found that blocking led to poor estimation of the u parameters in the model. Intutively, this is because blocking removes a large number of non-matching record pairs from the data, so there are less pairs available to estimate these parameters.

A little more rigorously, linking n_A records to n_B records results in $n_A n_B$ total record pairs. Under the assumption of one-to-one matching, at most n_B of these pairs are matches, so at least $n_A(n_B - 1)$ are non matches. Note that this number is **huge**.

Under blocking however, we are not guaranteed any nonmatching pairs at all! If we have unique values for blocking criteria, then we can end up in a situation with 0 nonmatching record pairs. In this setting, the model cannot separate pairs into matches and nonmatches (because there no nonmatches!)

Data Driven Priors for U-Parameter

Instead of needing $n_A(n_B - 1)$ nonmatching pairs that would result from all-to-all comparions, or allowing for the possibility of 0 nonmatching pairs, we seek a middle ground. For each feature f, we take R random records in B and compute agreement levels with S random records $i \notin N_j \subset A$. This is a form of *bootstrapping* that simulates the behavior of nonmatching pairs. We use these agreement level counts as a *data driven prior* on the u parameters.

Intuitively, this procedure is meant to recover the mean of the u parameters that would have resulted from all-to-all comparisons, but with higher variance (because there are less counts). However, we have seen that the u parameters are generally *absurdly precise*, this is a negligible sacrifice.

Over the summer, we found that this step greatly improved model performance.

Data Driven Prior on Rate of Matching

Note that we can write $N = N_{\text{in}} + N_{\text{out}}$, where N_{in} is the number of records in B that pass through blocking, and N_{out} is the number of records that don't.

The Beta linkage method uses a parameter π for the overall rate of matching across two datasets. If we use blocking, and remove the N_{out} records in B that have no plausible match in A, this parameter becomes biased upwards.

However, we can maintain the relevant information from these nonmatching records through an informative prior on the rate of matching! If $\pi \sim \text{Beta}(\alpha_\pi, \beta_\pi)$, simply set $\beta_\pi = N_{\text{out}}$. Then, calculations for the rate of matching parameter are essentially equivalent to if they done without blocking!

Elements of Paper

- Plot showing how increasing "prior counts" of nonmatching pairs through R and S leads to parameters that converge to those found under all-to-all comparisons.
- Simulation study showing
  - Results without blocking
  - Results with blocking, but no informative priors (should be poor)
  - Results with blocking and informative priors (should nearly recover results without blocking)
- Memory comparisons (less pairs to compute) and time comparisons
- Data
  - Same data Jody is using for other paper
  - Data from North Carolina State Board of Elections

Bonus piece

In fastLink, one of the ways the increase scalability is by learning model parameters on a subset of record pairs, and then applying those parameters to the full set of record pairs in order to determine matching status (see paragraph before S4 in the appendix here).

One issue I see with doing this under fastLink that it would bias the estimate of the rate of matching parameter. In order for a matching record pair to make it into the algorithm is for the relevant record to be sampled from *both* datasets. Thus I suspect that this rate of matching parameter would be biased downward.

With blocking however, we (mostly) ensure that a record and its match are both in the analysis we conduct. We can then run the model on a subsample of the blocks, and apply those parameters to the rest of the blocks after. This was pretty successful when we tested it out over the summer, and provides *massive* gains in scalability.

Notes about other methods

This "train and test" method wouldn't be possible under the Sadinle (2017) model because its not possible to simply apply parameters in a way that respects one-to-one matching.

Also, I don't think it would work under `d-blink` either, because there are no parameters that you can simply apply to records to get posterior match probabilities.

So I think there would be really interesting comparisons between `fastlink`, `BRL`, and `d-blink`!