BAYESIAN RECORD LINKAGE

RACHEL ANDERSON

1. Introduction

Ask any economist who works with survey or administrative data how they handle imperfect matches when performing a one-to-one merge, and you will get a different answer. Some pick the matches that they believe are most likely to be correct. Some estimate the same model using multiple configurations of matched data. Others avoid the issue entirely, by dropping all matches that are flagged during the automated merge processes in R or Stata. There is no one-size-fits-all solution for data merging, nor a formal theory about how subjective decisions in the merging process may introduce bias or uncertainty into estimation and inference in economic models.

This "'merging" problem of joining records from multiple data sources that describe the same entity appears in many fields, such as statistics, computer science, operations research, and epidemiology, under many names, such as record linkage, data linkage, entity resolution, instance identification, de-duplication, merge/purge processing, and entity reconciliation. In econometrics, these topics fall under the umbrella of "data combination", and its importance is growing with the increasing availability of large administrative datasets (Ridder and Moffitt, 2007). While there are many record linkage techniques, there is little understanding of how these techniques may bias inference, or how researchers should handle these issues in practice. Examining Bayesian record linkage tools is a promising place to start, as they provide a coherent framework for incorporating uncertainty at every point of the analysis.

In this paper, I review and implement two Bayesian record linkage procedures from Larsen and Rubin (2001) and Sadinle (2017) to obtain a posterior distribution over the bipartite matching/sets of matches vs. non-matches. I use simulated data (and maybe a real data set if I have time). I analyze adjustments for what makes a better match and compare them to traditional techniques.

Then, with these posteriors, I perform a basic regression analysis using methods from Lahiri and Larsen (2005) and Tancredi and Liseo (2015) Tancredi and Liseo (2011) about how to propagate uncertainty from the matches into a basic regression analysis.

I (1) review the two methods (2) implement both, and compare results (3) write a research plan – how I will continue the analysis

2. Setup

Consider two datafiles X_1 and X_2 that record information from two overlapping sets of individuals or entities. The files originate from two record-generating processes that may induce errors and missing values, so that identifying which individuals are represented in both X_1 and X_2 is nontrivial. I assume that individuals appear at most once in each datafile, so that the goal of record linkage is to identify which records in files X_1 and X_2 refer to the same entities.

Suppose that files X_1 and X_2 contain n_1 and n_2 records, respectively, and without loss of generality that $n_1 \geq n_2$. Denote also the number of entities represented in both files as n_M, so that $n_2 \geq n_M \geq 0$.

Following the probabilistic record linkage framework of Fellegi and Sunter (1969), we say that the set of ordered record pairs $X_1 \times X_2$ is the union of two disjoint sets, *matches* (M) and *non-matches* (U):

$$M = \{(i,j) : i \in X_1, j \in X_2, i = j\}$$
$$U = \{(i,j) : i \in X_1, j \in X_2, i \neq j\}$$

Hence, the formal goal of record linkage is to identify whether an arbitrary record pair $(i,j) \in X_1 \times X_2$ belongs to M or U.

To perform this task, each record pair is evaluated according to L different comparison criteria, which are the result of comparing data fields for records i and j. For example, if a record pair (i,j) represents two individuals, the pair may be evaluated according to whether they share a first name or have the same birthday. These comparisons are represented by a *comparison vector*,

$$\gamma_{\mathbf{ij}} = (\gamma_{ij}^1, \ldots, \gamma_{ij}^\ell, \ldots, \gamma_{ij}^L)$$

where each comparison field γ_{ij}^ℓ may be binary-valued, as in "i and j have the same birthday," or use levels to account for partial agreement between strings (see Winkler, 1990, for details). The models presented herein use only binary comparison vectors, however they may be extended to allow for partial agreement using the methods from Sadinle (2017).

The probability of observing a particular configuration of $\gamma_{\mathbf{ij}}$ can be modeled as arising from the mixture distribution:

$$P(\gamma_{\mathbf{ij}}) = P(\gamma_{\mathbf{ij}}|M)p_M + P(\gamma_{\mathbf{ij}}|U)p_U \tag{1}$$

where $P(\gamma_{ij}|M)$ and $P(\gamma_{ij}|U)$ are the probabilities of observing the pattern γ_{ij} conditional on the record pair (i,j) belonging to M or U, respectively. The proportions p_M and $p_U = 1 - p_M$ are the marginal probabilities of observing a matched or unmatched pair. Applying Bayes' Rule, we obtain the probability of $(i,j) \in M$ conditional on observing $\gamma_{\mathbf{ij}}$,

$$P(M|\gamma_{\mathbf{ij}}) = \frac{p_M P(\gamma_{\mathbf{ij}}|M)}{P(\gamma_{\mathbf{ij}})} \tag{2}$$

so that if we can estimate the variables in (1), we can estimate the probability that any two records refer to the same entity in (2).

As shown by Fellegi and Sunter (1969), it is possible to use the estimated probabilities to construct an "optimal" matching, given any threshold for false positive and false negative match rates. Conversely, the probabilities also allow us to estimate the false positive rate for any configuration of matches (Gelman et al., 2014). Given the usefulness of the quantities in (2), the next sections will introduce two methods for estimating them.

3. Simplifying assumptions

Let $\mathbf{\Gamma} \equiv \{\gamma_{\mathbf{ij}} : (i,j) \in X_1 \times X_2\}$ denote the set of comparison vectors for all records pairs $(i,j) \in X_1 \times X_2$.

3.1. **Blocking.** Note that $\mathbf{\Gamma}$ contains potentially $n_1 \times n_2$ elements, so that calculating $\mathbf{\Gamma}$ may be computationally expensive when X_1 or X_2 is large. In practice, researchers partition $X_1 \times X_2$ into "blocks," such that only records belonging to the same block are attempted to be linked, and records belonging to different blocks are assumed to be nonmatches. For example, postal codes and household membership are often used to define blocks when linking census files (Herzog et al., 2007). Importantly, the blocking variables should be recorded without error, and sometimes there are none available.

This paper assumes that no blocking is used; or, alternatively, that records are already divided into blocks that can be analyzed independently using the methods outlined below.

3.2. **Conditional independence.** In principle, we can model,

$$\gamma_{\mathbf{ij}} \mid M \sim \text{Dirichlet}(\delta_{\mathbf{M}})$$
$$\gamma_{\mathbf{ij}} \mid U \sim \text{Dirichlet}(\delta_{\mathbf{U}})$$

However, there are $2^L - 1$ possible configurations of each $\gamma_{\mathbf{ij}}$, so that $\delta_{\mathbf{M}}$ and $\delta_{\mathbf{U}}$ may be very high-dimensional if we want to allow weights to vary across different comparison criteria.

A common assumption in the literature is that the comparison fields ℓ are defined so that γ_{ij}^{ℓ} are independent across ℓ conditional on match status. This implies:

$$P(\gamma_{ij}|C) = \prod_{\ell=1}^{L} P(\gamma_{ij}^{\ell}|C)^{\gamma_{ij}^{\ell}}(1 - Pr(\gamma_{ij}^{\ell}|C))^{1-\gamma_{ij}^{\ell}} \qquad C \in \{M, U\} \tag{3}$$

Hence the number of parameters used to describe each mixture class is reduced to L.

Larsen and Rubin (2001) have shown how to relax this assumption using log-linear models, but for now I assume conditional independence to ease computation.

4. Two Bayesian Approaches to Record Linkage

4.1. **Mixture Models.** Since membership to M or U is not actually observed, a convenient way of simultaneously estimating p_M, p_U and classifying record pairs as matches or non-matches is via mixture modeling, with mixture distributions $P(\gamma_{\mathbf{ij}}|M)$ and $P(\gamma_{\mathbf{ij}}|U)$.

For convenience, denote $p_{M\ell} = P(\gamma_{ij}^{\ell}|M)$ and $p_{U\ell} = P(\gamma_{ij}^{\ell}|U)$. Assuming conditional independence across ℓ (and global parameters that do not vary by block, if using blocked records), a convenient prior distribution is the product of independent Beta distributions,

$$p_M \sim \text{Beta}(\alpha_M, \beta_M)$$
$$p_{M\ell} \sim \text{Beta}(\alpha_{M\ell}, \beta_{M\ell}),\ \ell = 1, \ldots, L$$
$$p_{U\ell} \sim \text{Beta}(\alpha_{U\ell}, \beta_{U\ell}),\ \ell = 1, \ldots, L$$

For $i = 1, \ldots, n_1 \in X_1$, $j = 1, \ldots, n_2 \in X_2$ define the parameter of interest as,

$$I_{ij} = \begin{cases} 1, & \text{if records } i \in X_1 \text{ and } j \in X_2 \text{ refer to the same entity;} \\ 0, & \text{otherwise} \end{cases}$$

If $(p_M, p_{M\ell}, p_{U\ell})$ are known, then $P(I_{ij} = 1|\gamma_{\mathbf{ij}}(p_M, p_{M\ell}, p_{U\ell}))$ is distributed as in (2). Alternately, if the match indicators $\mathbf{I}$ were known, the posterior distributions of $(p_M, \mathbf{p_{M\ell}}, \mathbf{p_{U\ell}})$ would be:

$$p_M|I \sim \text{Beta}\left(\alpha_M + \sum_{(i,j)} I_{ij},\ \beta_M + \sum_{(i,j)}(1 - I_{ij})\right) \tag{4}$$

$$p_{M\ell}|I \sim \text{Beta}\left(\alpha_{M\ell} + \sum_{(i,j)} I_{ij}\gamma_{ij}^{\ell},\ \beta_{M\ell} + \sum_{(i,j)} I_{ij}(1 - \gamma_{ij}^{\ell})\right), \quad \ell = 1, \ldots, L \tag{5}$$

$$p_{U\ell}|I \sim \text{Beta}\left(\alpha_{U\ell} + \sum_{(i,j)}(1 - I_{ij})\gamma_{ij}^{\ell},\ \beta_{U\ell} + \sum_{(i,j)}(1 - I_{ij})(1 - \gamma_{ij}^{\ell})\right), \quad \ell = 1, \ldots, L \tag{6}$$

Based on these ideas, Larsen and Rubin (2001) proposed a Bayesian version of record linkage for the mixture model approach, that uses a Gibbs Sampling scheme[1] for simulating the posterior distribution of $I, (p_M, p_{M\ell}, p_{U\ell})$.

4.2. **Beta Record Linkage.** As noted by Larsen (2005), a significant limitation of the mixture model approach is that it does not explicitly enforce one-to-one matching in the likelihood. Intuitively, if one-to-one matching is desired, then the matching status of (i, j) should depend on the the matching status of (i, j') for $j \neq j'$. The Gibbs Sampler in Appendix A.1 certainly violates this, since the components of $\mathbf{I}$ are sampled independently in every iteration.

When the goal is one-to-one matching, Sadinle (2017) noted that the parameter of interest $\mathbf{I}$ is better characterized as a *bipartite matching*. Furthermore, he provides conditions under which the mixture model approach, combined with an optimal assignment of record pairs obtained from a linear sum assignment problem, produces the MLE of the bipartite matching.

Formally, the bipartite matching can be represented as $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_{n_2})$, where

$$Z_j = \begin{cases} i, & \text{if records } i \in X_1 \text{ and } j \in X_2 \text{ refer to the same entity;} \\ n_1 + i, & \text{if record } j \in X_2 \text{ does not have a match in } X_1 \end{cases}$$

This notation is also convenient computationally because it reduces the size of the matching parameter from $n_1 \times n_2$ to n_2.

Same as in the mixture model approach, the prior probability that $j \in X_2$ has a match is

$$I(Z_j \leq n_1) \overset{i.i.d}{\sim} \text{Bernoulli}(p_M), \quad p_M \sim \text{Beta}(\alpha_M, \beta_M \tag{7}$$

The prior on p_M implies $n_M(Z) = \sum_{j=1}^{n_2} I(Z_j \leq n_1)$, the number of matches according to $\mathbf{Z}$ is distributed as:

$$n_M(\mathbf{Z}) \sim \text{Beta-Binomial}(n_2, \alpha_M, \beta_M) \tag{8}$$

after marginalizing over p_M.

Conditioning on $\{I(Z_j \leq n_1)\}_{j=1}^{n_2}$, all possible bipartite matchings are considered to be equally likely, so

$$Pr\left(\mathbf{Z}|n_M\right) = \left(\frac{n_1!}{(n_1 - n_M)!}\right)^{-1} \tag{9}$$

[1]See Appendix A.1

Together conditions (7)-(9) imply the following prior over $\mathbf{Z}$:

$$Pr(\mathbf{Z}|\alpha_M, \beta_M) = \frac{(n_1 - n_M(\mathbf{Z}))!}{n_1!} \frac{\text{Beta}(n_M(\mathbf{Z}) + \alpha_M,\ n_2 - n_M(\mathbf{Z}) + \beta_M)}{\text{Beta}(\alpha_M, \beta_M)} \tag{10}$$

It is because of the prior in (10) that Sadinle (2017) refers to his methods as *beta record linkage*.

To estimate the model, I use the Gibbs Sampling scheme outlined in Larsen (2005), which I have reproduced in Appendix A.2. It is very similar to the Gibbs Sampler for estimating the mixture model, however the drawing from the conditional distribution of $\mathbf{Z}$ is significantly more challenging (and computationally intensive) than sampling from the conditional distribution of $\mathbf{I}$.

5. Simulation Study

To compare the performance of the mixture model approach with the beta record linkage model, I wrote Python scripts that implement the Gibbs Samplers described in Appendices A.1 and A.2. I also wrote scripts to generate fake data using different parameter combinations so I could study how the methods perform across multiple datasets. Detailed documentation for all relevant programs is available in `README.md` submitted with this paper.

5.1. **Data.** For each combination of parameters in Table 1, I generated a dataset by randomly selecting $n_M = p_M^* n_2$ records from X_2 and n_M records from X_1 and assigning them as matches.

I must emphasize that this means p_M^* controls the amount of overlap between the two files. When $p_M^* = 0.9$ and $n_1 = n_2 = 10$, this means all but one record in X_2 has a match in X_1. In the mixture model, these same parameter values would imply $p_M = 0.09$ according to (1), since there are only 9 record pairs referring to matches out of 100 possible candidate pairs. In the beta record linkage model, however, p_M^* refers to the proportion of records in X_2 that have a match. Unfortunately, it is not clear that I can compare the posteriors of p_M across the two types of models for these reasons.[2]

For these record pairs, I generated $\gamma_{\mathbf{ij}}$ according to

$$\gamma_{ij}^{\ell} \mid M \overset{i.i.d.}{\sim} \text{Bernoulli}(p_{M\ell}), \quad \ell = 1, \dots L$$

For the remaining $(n_1 n_2 - n_M)$ record pairs, I generated $\gamma_{\mathbf{ij}}$ from

$$\gamma_{ij}^{\ell} \mid U \overset{i.i.d.}{\sim} \text{Bernoulli}(p_{U\ell}), \quad \ell = 1, \dots L$$

For this paper, I tested $p_{M\ell} = 0.9$ and $p_{U\ell} = 0.2$ for all ℓ and all datasets, however my Python programs allow for more flexible choices.

5.2. **Priors.** This paper tests only flat priors, ($\alpha_M = \beta_M = \alpha_{M\ell} = \beta_{M\ell} = \alpha_{U\ell} = \beta_{U\ell} = 1$), however, my programs allow for any valid combination of hyperparameters.

In future tests, I would like to try restricting the following restrictions suggested by Larsen (2005):

[2]Admittedly, this is a significant mistake on my part that I realized only after analyzing the results.

TABLE 1. Parameter combinations and time to perform 100,000 iterations (in seconds)

L	n_1	n_2	p_M^*	p_M	$p_{M\ell}$	$p_{U\ell}$	MM	BRL
2	10	10	0.2	0.02	0.9	0.2	237	4174
3	10	10	0.2	0.02	0.9	0.2	292	4315
4	10	10	0.2	0.02	0.9	0.2	329	4361
5	10	10	0.2	0.02	0.9	0.2	379	4417
2	20	20	0.5	0.05	0.9	0.2	900	14967
3	20	20	0.5	0.05	0.9	0.2	1274	18643
4	20	20	0.5	0.05	0.9	0.2	1490	18618
5	20	20	0.5	0.05	0.9	0.2	1469	15470
2	10	10	0.9	0.09	0.9	0.2	242	4246
3	10	10	0.9	0.09	0.9	0.2	289	4255
4	10	10	0.9	0.09	0.9	0.2	334	4327
5	10	10	0.9	0.09	0.9	0.2	382	4249

(1) Logically, the range of p_M should be restricted to be less than or equal to the smaller of the two file sizes divided by the number of pairs under the blocking structure. In my application, which does not use blocking, this would mean truncated any draws that violate $p_M \leq \frac{\min\{n1,n2\}}{n1n_2} = \frac{1}{n_1}$, since I assumed $n_1 \geq n_2$.

(2) The probability of a record pair agreeing on a comparison field should be no weakly greater among matches than among nonmatches. That is,

$$P(\gamma_{ij}^{\ell}|M) \geq P(\gamma_{ij}^{\ell}|U)$$

for all ℓ. This can be achieved by truncating the draws for $p_{U\ell}$ in the Gibbs sampler by ignoring draws that violate this condition, or by scaling $p_{M\ell}$ to be in the range $(0, p_{M\ell})$.

5.3. **Computation.** Due to the long run time required to estimate the beta record linkage model, I performed my posterior simulation using the Adroit cluster hosted by Princeton Research Computing. Included in my code directory is a Python script `job_writer.py` that reads in a CSV file of parameter combinations and outputs a SLURM file that can be submitted to the cluster to run the simulations.

For each dataset, I perform 100,000 iterations for each the mixture model and beta record linkage Gibbs samplers. Prior to submitting these jobs to the cluster, I performed the analysis on my local machine and estimated the run time to be approximately linear in $n_1 \times n_2 \times L \times$ # iterations. This does not seem to be entirely true on the cluster, since $(L = 5, n_1 = n_2 = 20)$ completed faster than $(L = 3, n_1 = n_2 = 20)$.

Finally, I compute all THINGS IN LOGS ** mention from BDA3 HERE

6. RESULTS

6.1. **Convergence.**

6.1.1. *Mixture Models.* To evaluate convergence for the mixture models, I look at the trace plots of $(p_M, p_{M\ell}, p_{U\ell})$. A common pattern across all chains is that p_M converges to values near 0 or 1 within a few initial draws, and then stays close to that value for all remaining draws, as in Figure 1. Given the fact that the true values of p_M are less than 0.09 for all datasets, this is not concerning. The fact that p_M sometimes converges to 1 is an artifact of the label degeneracy problem for mixture models; although I label my parameters with M and U, my model cannot tell the difference.

Additionally, one of $\mathbf{p_{M\ell}}$ or $\mathbf{p_{U\ell}}$ oscillates between 0 and 1 for all draws, while the other oscillates between 0.15 and 0.35. Because the values of p_M are so low, $\mathbf{p_{M\ell}}$ is not well identified, and so its posterior is the same as the prior (Figure 3). My Gibbs sampler, in turn, classifies almost all record pairs as non-matches, so that I effectively estimate the parameters $\mathbf{p_{U\ell}}$ using the entire sample. The resulting posteriors are approximately Normal, with $\hat{\mu} \approx 0.22, \hat{\sigma} \approx 0.04$ across $p_{U\ell}$.

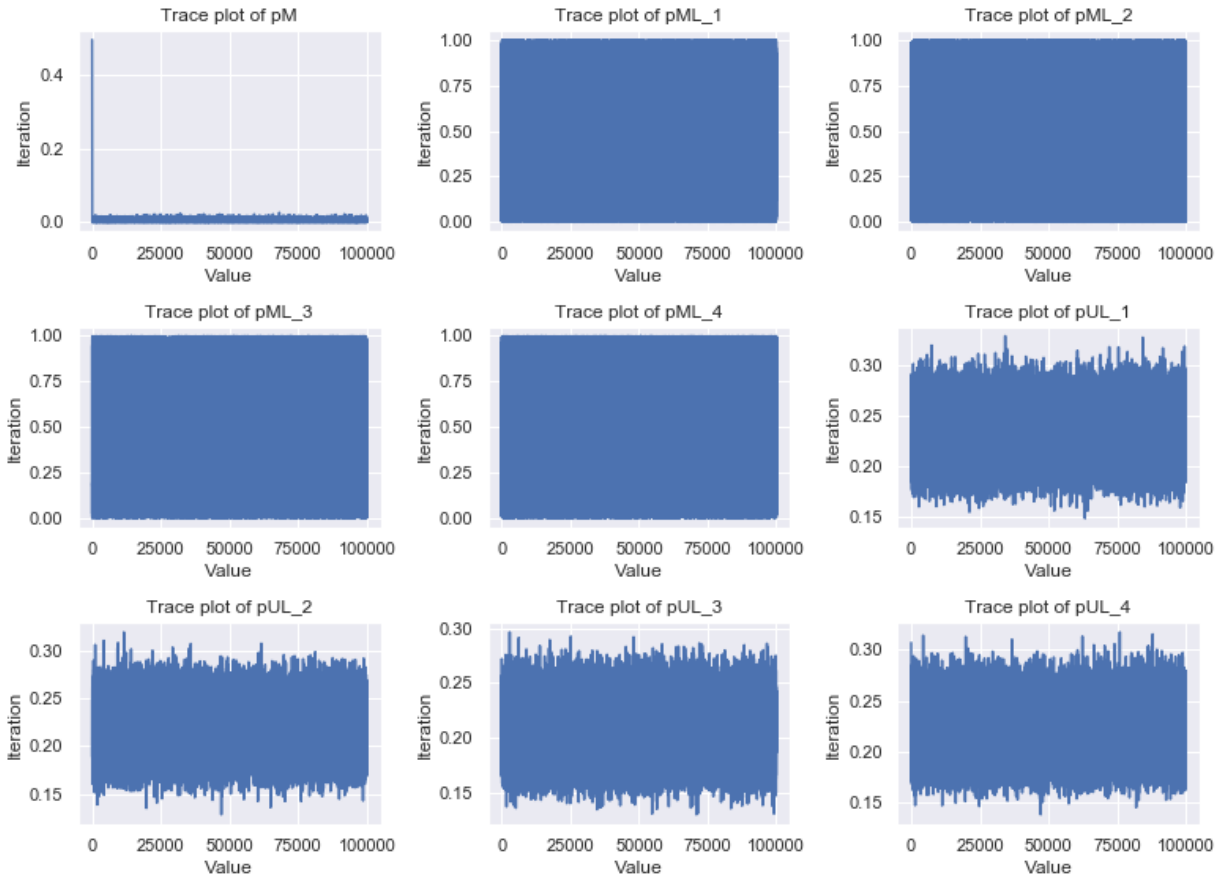

FIGURE 1. Trace plots for 100,000 draws of $(p_M, p_{M\ell}, p_{U\ell})$ with true values of $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (4, 20, 20, 0.5, 0.9, 0.2)$ and no burn-in

6.1.2. *Beta Record Linkage.* Judging by the trace plots for $(p_M, p_{M\ell}, p_{U\ell})$ and the likelihood of Z from the last 15,000 draws of the chain plotted in Figure 2, it appears that $\mathbf{p_{U\ell}}$ mixes well, but that there are slow-moving trends in $p_M, \mathbf{p_{M\ell}}$ and the likelihood of Z. These results suggest that the chain has not yet converged. Most likely this is due to the fact that the Gibbs Sampler in Appendix A.2 changes $\mathbf{Z}$ incrementally, adding, deleting, or swapping one record pair at a time. With just

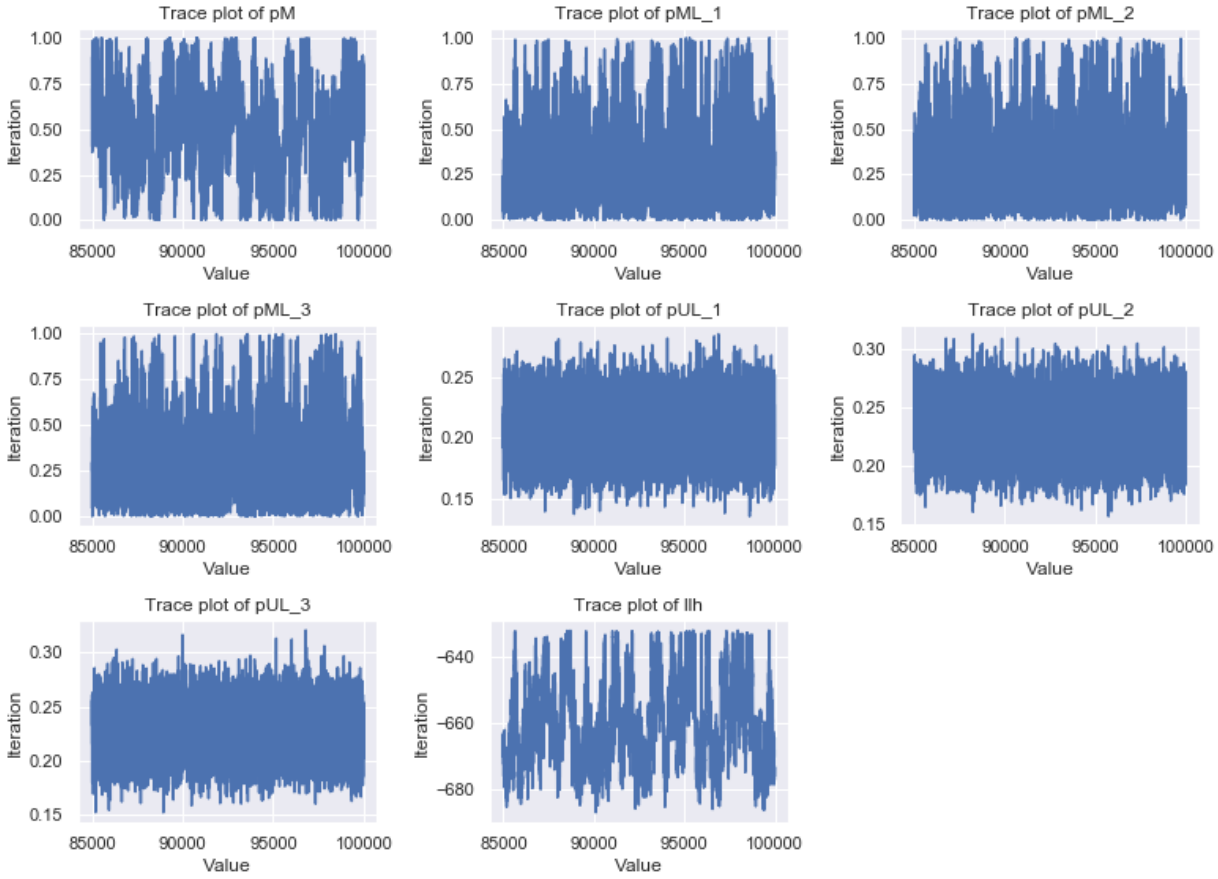

FIGURE 2. Trace plot of draws 85,000-100,000 for $(p_M, p_{M\ell}, p_{U\ell})$ and the likelihood of Z from a Beta Record Linkage model with true parameters $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (5, 20, 20, 0.5, 0.9, 0.2)$

$n_1 = n_2 = 10$, there are $234,662,231$ possible configurations of $\mathbf{Z}$. In Figure 2, $n_1 = n_2 = 20$, so it is not surprising that after only 100,000 draws, the chain has not yet converged.

6.2. **Posterior Densities.** Figures 3 and 4 plot the simulated posterior densities for both models using the same dataset with parameters $(L, n_1, n_2, p_M^*, p_M, p_{M\ell}, p_{U\ell}) = (5, 20, 20, 0.5, 0.025, 0.9, 0.2)$. The posterior distributions for $\mathbf{p_{U\ell}}$ are virtually identical for both models. This is true for all combinations of parameters (although, as mentioned previously, sometimes the U and M is flipped in the mixture model estimates).

As previously mentioned, the implied parameters for p_M are different for the two models – in this case, the true $p_M^{MM} = 0.025$ and $p_M^{BRL} = 0.5$. Whereas the histogram of p_M in Figure 3 shows that the posterior values are close to the truth, the distribution of p_M draws from the beta record linkage model in Figure 4 still look like the prior.

This is likely because p_M is sampled from the posterior according to

$$p_M \mid \mathbf{Z} \sim \text{Beta}(\alpha_M + n_M(\mathbf{Z}),\ \beta_M + n_2 - n_M(\mathbf{Z})) \tag{11}$$

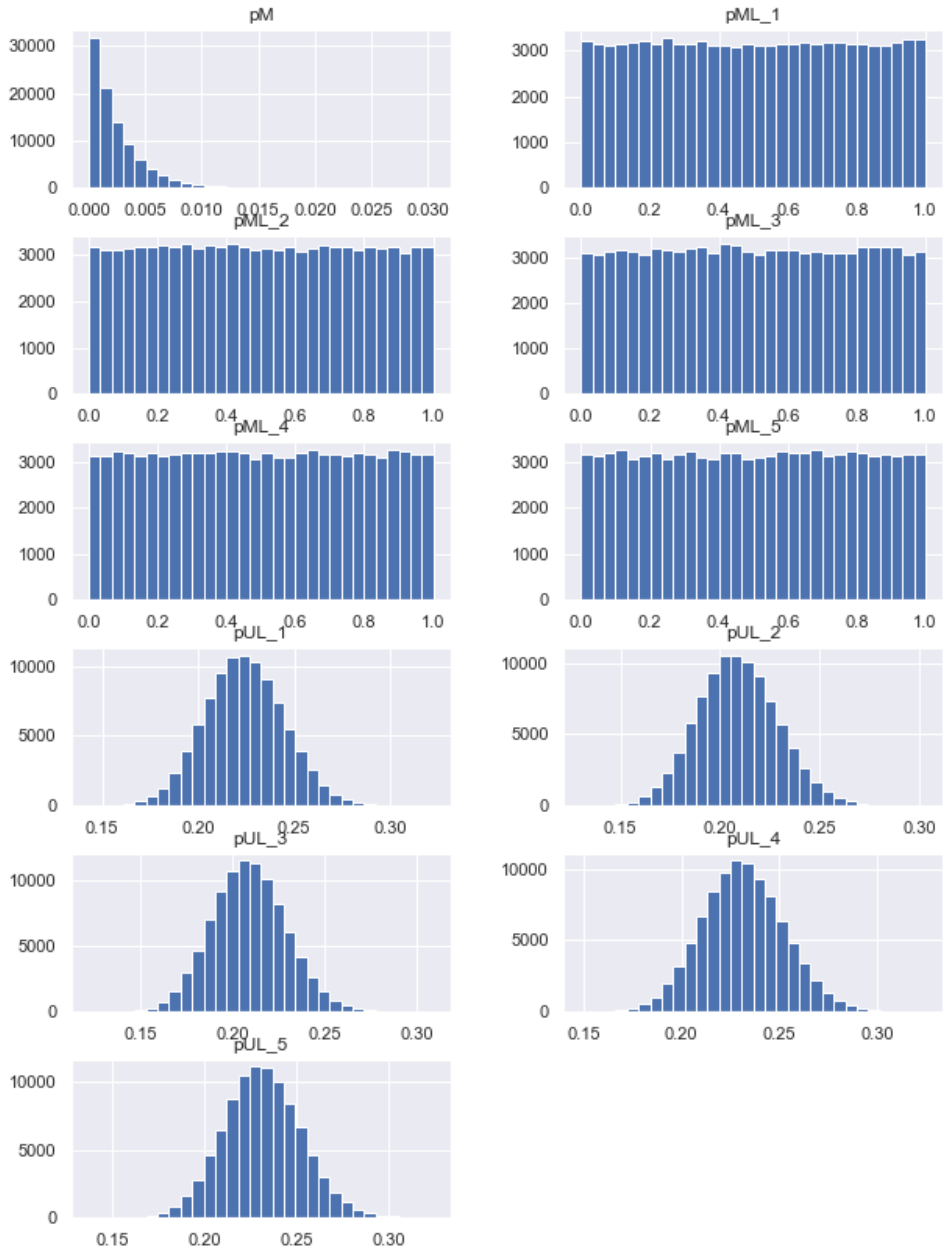

FIGURE 3. Histogram of 100,000 draws (burn-in = 5,000) from mixture model for all parameters with true parameter values $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (5, 20, 20, 0.5, 0.9, 0.2)$ and independent, flat priors

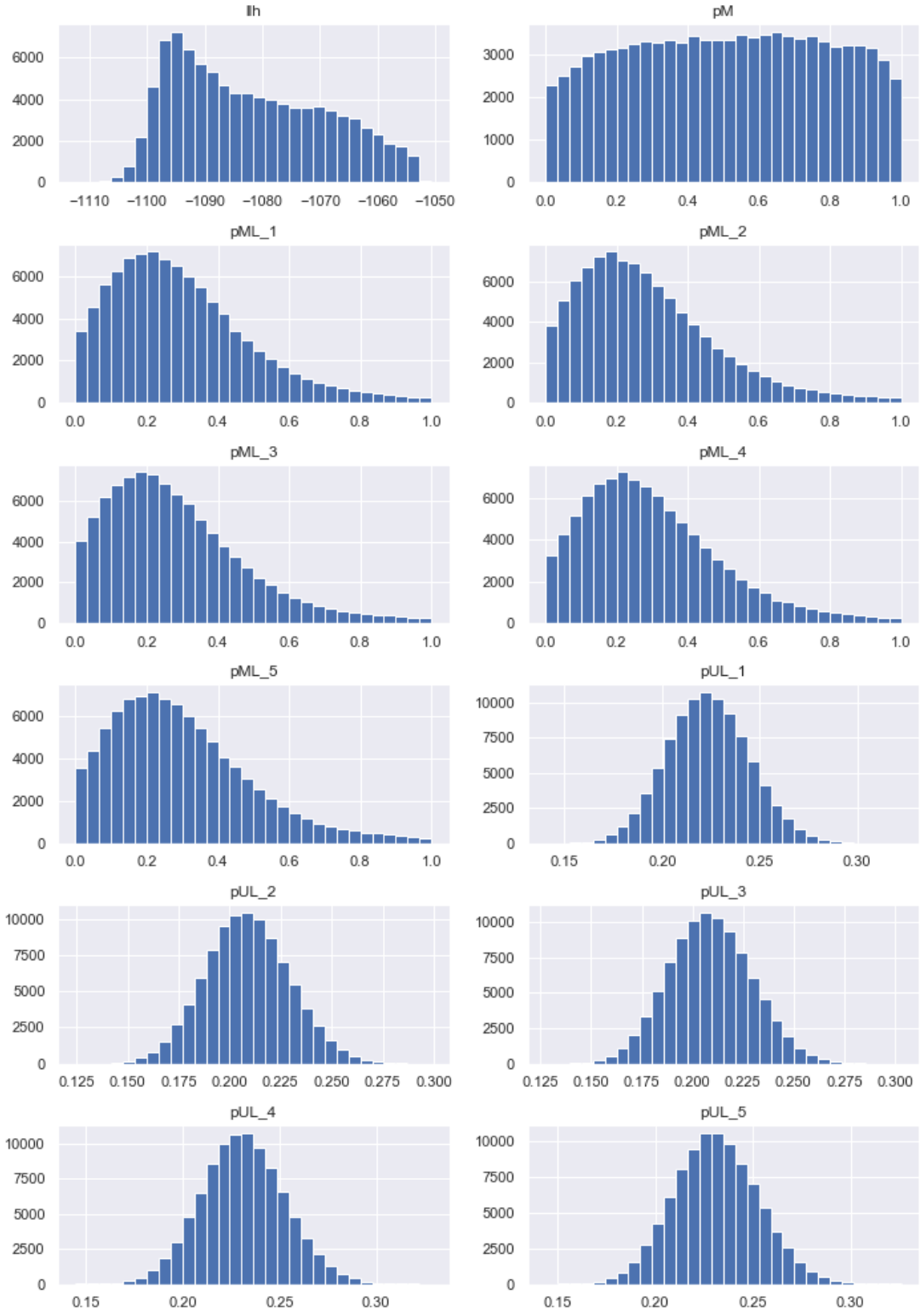

FIGURE 4. Histogram of 100,000 draws (burn-in = 5,000) from Beta Record Linkage model for all parameters and likelihood of Z with true parameter values $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (5, 20, 20, 0.5, 0.9, 0.2)$ and independent, flat priors

in the beta record linkage model. Since $n_M(\mathbf{Z})$ is bounded between $[0, 20]$ across applications, the parameters of the Beta distribution are bounded between $[1, 21]$ given the flat prior. By comparison, the mixture model samples p_M from 7, where $\sum_{(i,j)} I_{ij} \in [0, 400]$ depending on the parameters tested. As a result, the p_M draws from the mixture model converge to either 0 or 1 extremely fast.

Interestingly, the shapes for $p_{M\ell}$ are different not In the Beta Record Linkage model, the posterior density of p_M remains rather flat, and the posterior densities for $p_{M\ell}$ look like a χ^2 distribution.

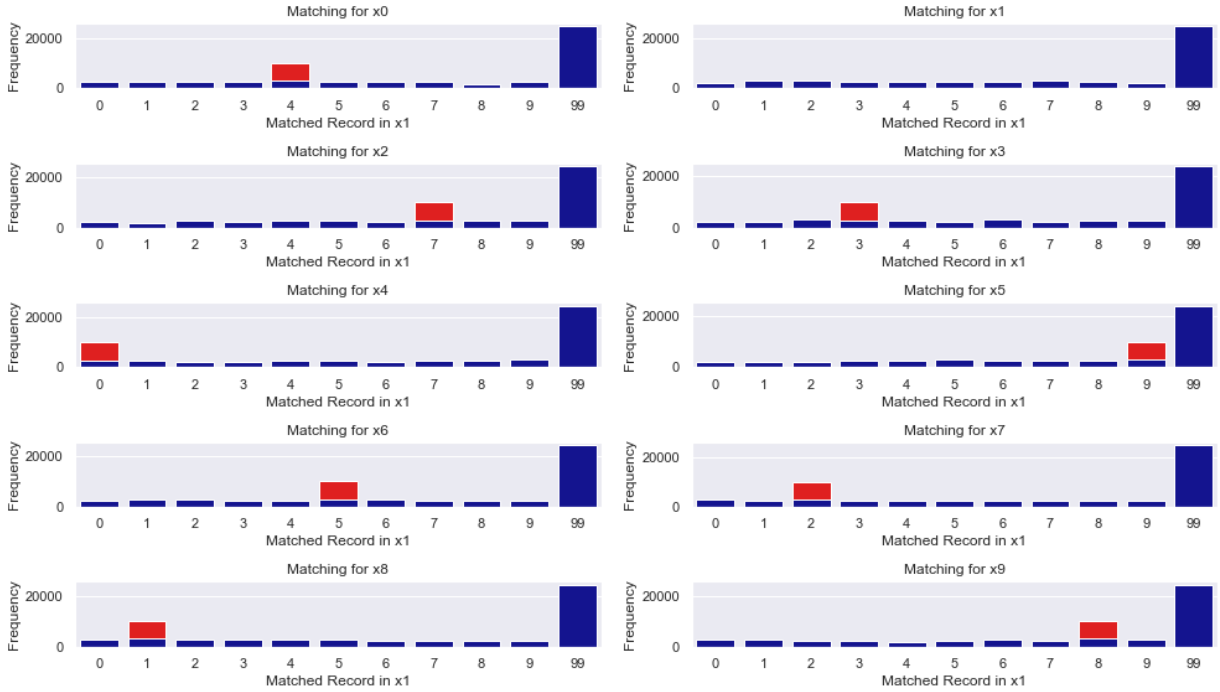

FIGURE 5. Histogram of Z draws for bipartite matching from a Beta Record Linkage model with true parameters $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (3, 10, 10, 0.9, 0.9, 0.2)$

6.3. **Posterior match probabilities.** The estimated posterior probability of $(i, j) \in M$ is:

$$\frac{\#\text{ iterations that } Z_j = i}{\text{total } \#\text{ of iterations}}$$

for all $(i, j) \in X_1 \times X_2$. Figure 5 shows that even when almost all records have a match ($p_M = 0.9$), the posterior probability that any record is assigned a match is low. Conditional on being assigned a match in a draw, however, the posterior seems to slightly favor the correct match (Figure 6) so that if we used a decision rule to assign matches, some of the matches may be correctly assigned. Given that the chain has not converged, this seems to be promising, albeit not ideal.

7. DISCUSSION

7.1. **Mixture Models.** A significant limitation of the mixture model approach is that there is no guarantee that the clusters will correspond to matches and non-matches. Although this is a

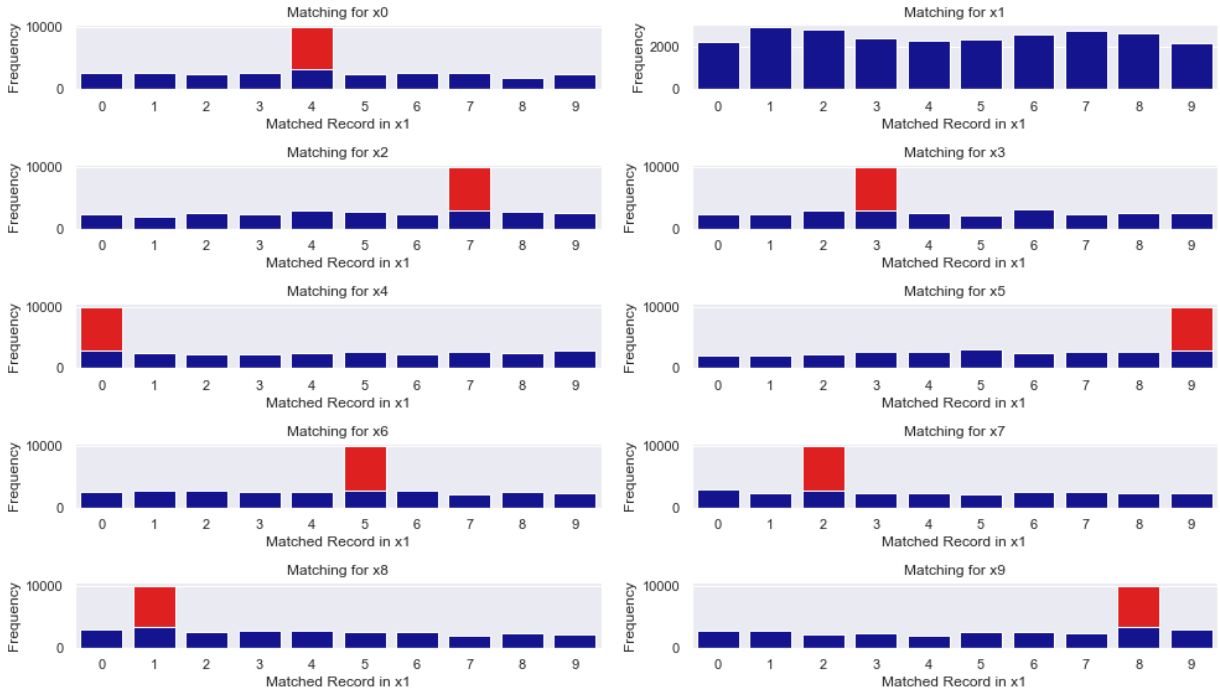

FIGURE 6. Histogram of Z draws for bipartite matching (conditional on being matched) from a Beta Record Linkage model with true parameters $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (3, 10, 10, 0.9, 0.9, 0.2)$

criticism about mixture models in general, this proved to be an issue in even the most simple datasets I generated.

In practice, 3-component mixtures tend to work better, even though theoretically we would like two. Winkler (2002) mentioned conditions for the mixture model to give good results: the proportion of matches should be greater than 5%, the classes of matches and non-matches should be well-separated, typographical errors must be relatively low, there must be redundant fields that overcome errors in other fields, among others.

Many-to-one matches can still happen unless the linear sum assignment is used.

Another issue with mixture models is that, even if mixture model is fitted with the one-to-one constraint, the FS decision rule alone may lead to many-to-many assignments. The linkage decision for the pair (i, j) not only depends on γ_{ij} but on the other pairs.

7.2. **Beta Record Linkage.** Sadinle (2017) uses beta record linkage to match files with sizes $n_1 = 4,430, n_2 = 1,324$ which gives $5,852,080$ record pairs. He claims that his chains converged after just 2,000 iterations of the Gibbs Sampler based on Geweke's convergence diagnostic as implemented in `R` package `coda`. Based on my own preliminary results, I am skeptical of this claim.

In future versions of this paper, I will explore how to apply similar functions to my output, or write a custom module that calculates the appropriate convergence criteria from Brooks and Gelman (1998) and Gelman and Rubin (1992).

I would like to explore the possibility of using Dirichlet distributions to model to the comparison vectors, or by using some sort of machine learning.

7.3. **Interpreting the results.** For a candidate pair (i, j), the posterior probability of a match is $P(I_{ij} = 1|p_M, p_{M\ell}, p_{U\ell}) = \frac{1}{K}\sum_k I_{ij}^{(k)}$. Options for designating matches are to (1) designate all candidate pairs exceeding a cutoff as matches, or (2) use a linear program to enforce one-to-one matching.

7.4. **Enforcing one-to-one assignment.** The FS decision rule does not enforce the maximum one-to-one assignment that is desirable in many economic applications. In practice, the optimal assignment of record pairs is obtained by solving the linear sum assignment problem:

$$\max_{\Delta} \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} w_{ij}\Delta_{ij}$$
$$\text{subject to } \Delta_{ij} \in \{0,1\}; \sum_{i=1}^{n_1} \Delta_{ij} \leq 1,\ j = 1, \ldots, n_2$$
$$\text{and } \sum_{j=1}^{n_2} \Delta_{ij} \leq 1,\ i = 1, \ldots, n_1$$

where the constraints ensure that Δ represents a bipartite matching. The output of this step is a bipartite matching that maximizes the sum of the weights w_{ij} among matched pairs, and the pairs that are not matched. Sadinle (2017) shows that this can be thought of as the MLE under the assumption that the comparison vectors are conditionally independent given the bipartite matching.

7.5. **Extensions.**

7.5.1. *Machine learning.* Other option is to use training data to obtain the weights. Another is to apply mixture models to the comparison vectors directly. This latter method is favorable because in many contexts training data is not available, or creating a sufficiently large, representative training data set is costly.

7.5.2. *Looking at bias.* Scheuren and Winkler (1993) and Lahiri and Larsen (2005) propose bias-corrected estimators of coefficients in a linear regression model given data from a probabilistically linked file.

7.5.3. *missing data.*

7.5.4. *Real data.* Now that I have verified that the simple gibbs sampler works, perhaps the most obvious extension will be to extend these techniques to real data. The techniques are tested first using a synthetic data generator developed by Christen and Pudjijono (2009) and Christen and Vatsalan (2013), and then with two real datasets from Enamorado (2018) and Enamorado, Fifield and Imai (2018).

(1) In the first empirical application I merge two datasets on local-level candidates for the 2012 and 2016 municipal elections in Brazil. Each dataset contains more than 450,000 observations with a perfectly-recorded unique identifier, the Brazilian individual taxpayer registration identification number (called the *Cadastro de Pessoas Físicas*). All other common identifiers are manually entered into the database, so they may contain errors.

(2) In the second application, I merge the 2016 American National Election Study(ANES) with a nationwide voter file containing over 160 million voter records.

8. Conclusion

Referee for Belin (93): "every gain which is achieved by a superior record linkage procedure must be justified by the cost of implementing that procedure".

References

Brooks, S. P. and A. Gelman (1998). General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics 7*(4), 434–455.

Fellegi, I. P. and A. B. Sunter (1969). A theory for record linkage. *Journal of the American Statistical Association 64*(328), 1183–1210.

Fortini, M., B. Liseo, A. Nuccitelli, and M. Scanu (2001). On bayesian record linkage. *Research in Official Statistics 4*(1), 185–198.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2014). *Bayesian data analysis*. CRC Press, Taylor Francis Group, an informa business.

Gelman, A. and D. B. Rubin (1992). Inference from iterative simulation using multiple sequences. *Statistical Science 7*(4), 457–472.

Herzog, T. N., F. J. Scheuren, and W. E. Winkler (2007). *Data Quality and Record Linkage Techniques* (1st ed.). Springer Publishing Company, Incorporated.

Lahiri, P. and M. D. Larsen (2005). Regression analysis with linked data. *Journal of the American Statistical Association 100*(469), 222–230.

Larsen, M. (2005, 10). Hierarchical bayesian record linkage theory.

Larsen, M. D. and D. B. Rubin (2001). Iterative automated record linkage using mixture models. *Journal of the American Statistical Association 96*(453), 32?41.

Ridder, G. and R. Moffitt (2007). Chapter 75 the econometrics of data combination. *Handbook of Econometrics*, 5469?5547.

Sadinle, M. (2017). Bayesian estimation of bipartite matchings for record linkage. *Journal of the American Statistical Association 112*(518), 600?612.

Tancredi, A. and B. Liseo (2011). A hierarchical bayesian approach to record linkage and population size problems. *The Annals of Applied Statistics 5*(2B), 1553?1585.

Tancredi, A. and B. Liseo (2015). Regression analysis with linked data: problems and possible solutions. *Statistica 75*(1), 19–35.

Winkler, W. (1990, 01). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *Proceedings of the Section on Survey Research Methods*.

Appendix A. Appendix

A.1. **Gibbs Sampler (Larsen, 2005).**

(1) Specify parameters for the prior distributions. Choose initial values of $\left(p_M^{(0)}, p_{M\ell}^{(0)}, p_{U\ell}^{(0)}\right)$ for $\ell = 1, \dots L$.

(2) Repeat the following steps numerous times until the distribution of draws has converged to the posterior distribution of interest:

(a) Using the current values of $\left(p_M^{(k)}, p_{M\ell}^{(k)}, p_{U\ell}^{(k)}\right)$, draw $I_{ij}^{(k+1)}$ for each (i, j) candidate pair as an independent draw from a Bernoulli distribution with parameter

$$Pr\left(I_{ij}^{(k+1)} = 1 \middle| \gamma_{ij}\right) = Pr\left(M|\gamma_{ij}\right) = \frac{p_M^{(k)} Pr\left(\gamma_{ij}|M, p_{M\ell}^{(k)}\right)}{Pr\left(\gamma_{ij}\middle|p_M^{(k)}, p_{M\ell}^{(k)}, p_{U\ell}^{(k)}\right)} \tag{12}$$

where

$$Pr\left(\gamma_{ij}|M, p_{M\ell}^{(k)}\right) = \prod_{\ell=1}^{L} \left(p_{M\ell}^{(k)}\right)^{\gamma_{ij}^{\ell}} \left(1 - p_{M\ell}^{(k)}\right)^{1-\gamma_{ij}^{\ell}}$$

and the denominator is calculated according to (1) above.

(b) Draw a value of $p_M^{(k+1)}$ from (4).

(c) Draw values of $\{p_{M\ell}^{(k+1)}\}_{\ell=1}^{L}$ independently from (5).

(d) Draw values of $\{p_{U\ell}^{(k+1)}\}_{\ell=1}^{L}$ independently from (6).

(3) Stop once the algorithm has converged. Criteria for Convergence ARE X Y Z.

A.2. **Gibbs sampler for bipartite matching (Larsen, 2005).**

(1) Pick an initial values of $p_M, p_{M\ell}, p_{U\ell}$ and a valid configuration of Z. Repeat the following until convergence:

(a) Draw p_M from

$$p_M \mid Z \sim \text{Beta}(\alpha_M + n_M(Z),\ \beta_M + n_2 - n_M(Z))$$

(b) Draw $p_{M\ell}$ and $p_{U\ell}$ from their conditional distributions (same as before).

(c) Use Metropolis-Hastings algorithm to draw values of Z and $n_M(Z)$ from their full conditional distributions.

(2) Stop when converged.

I implement the incremental method for modifying n_M and Z via the Metropolis-Hastings steps outlined in the appendix of Larsen (2005). As Larsen notes, there are various Gibbs and Metropolis-Hastings sampling procedures that could generate draws from the target distribution, however most are computationally infeasible. The following procedure is designed to cover the space of possible configurations and to produce higher probabilities of change across iterations.

The full conditional distribution of (n_m, Z) is:

$$Pr\left(n_m, Z|\gamma, p_{M\ell}, p_{U\ell}, p_M, \alpha, \beta\right) \propto Pr(n_m|p_M)Pr(Z|n_M)Pr\left(\gamma|Z, \{p_M\ell, p_{U\ell}\}_{\ell=1}^{L}, p_M, \alpha, \beta\right) \quad (13)$$

This is used to calculate the jump probabilities $P(n_M, Z)$ in the moves below. For all types of moves, there are more clever ways to select which pairs to add, drop, or switch, however these are computationally expensive. Future steps may include optimizing these steps, although ACCEPTANCE PROBABILITY IS PRETTY GOOD

Move 1: $n_M^* = n_M - 1$

A pair (i, j) is picked at random from the set of matched record pairs according to the current configuration of Z with equal probabilities. The probability of picking pair (i, j) is $(n_M)^{-1}$.

The inverse move is to add the deleted pair of records to the set of designated matches. If a non-matching pair is selected at random with equal probabilities, the probability of selecting the dropped match is $((n_1 - n_M + 1)(n_2 - n_M + 1))^{-1}$. Hence the acceptance probability for the Metropolis-Hastings algorithm is:

$$\min\left\{1, \frac{Pr\left(n_M^*, Z^*|\text{ current parameter values}\right) n_M}{Pr\left(n_M, Z|\text{ current parameter values}\right)(n_1 - n_M + 1)(n_2 - n_M + 1)}\right\}$$

Move 2: $n_M^* = n_M + 1$

A pair (i, j) is selected at random from the set of non-matches according to the current configuration of Z with probability $((n_1 - n_M)(n_2 - n_M))^{-1}$.

The inverse move is to delete a pair of records from the set of designated matches with equal probability n_M^{-1} for each pair. This implies the acceptance probability for the Metropolis-Hastings algorithm:

$$\min\left\{1, \frac{Pr\left(n_M^*, Z^*|\text{ current parameter values}\right)(n_1 - n_M)(n_2 - n_M)}{Pr\left(n_M, Z|\text{ current parameter values}\right)(n_M + 1)}\right\}$$

Move 3: $n_M^* = n_M$**, but** Z **changes**

There are three variations of this move to consider.

Move 3.1: Two matches switch pairings

Two matched pairs (i, j) and (k, l) are selected at random from the set of matched pairs according to the current configuration of Z. Then the new pairs (i, l) and (k, j) are assigned as matches, and the old pairs (i, j) and (k, l) are assigned non-match status.

The reverse move is to undo the switch, so that the acceptance probability of the M-H algorithm is

$$\min\left\{1, \frac{Pr(\gamma_{il}|M)Pr(\gamma_{kj}|M)Pr(\gamma_{ij}|U)Pr(\gamma_{kl}|U)}{Pr(\gamma_{ij}|M)Pr(\gamma_{kl}|M)Pr(\gamma_{il}|U)Pr(\gamma_{kj}|U)}\right\}$$

Move 3.2: A matched pair replaces one of its matching records with a non-matching record

A matched pair (i, j) is chosen with uniform probability $(n_M)^{-1}$ from the set of matches. With probability $1/2$, $i \in X_1$ is assigned a new match at random; otherwise, $j \in X_2$ is assigned a new match at random.

The new match is selected at random from the corresponding set of unmatched records in X_1 or X_2, according to whether i or j is dropped. If the record $i \in X_1$ is replaced through random selection with $k \in X_1$ then the M-H acceptance probability is

$$\min\left\{1, \frac{Pr(\gamma_{kj}|M)Pr(\gamma_{ij}|U)}{Pr(\gamma_{ij}|M)Pr(\gamma_{kj}|U)}\right\}$$

If a record $j \in X_2$ is replaced through random selection with $l \in X_2$, then the M-H acceptance probability is

$$\min\left\{1, \frac{Pr(\gamma_{il}|M)Pr(\gamma_{ij}|U)}{Pr(\gamma_{ij}|M)Pr(\gamma_{il}|U)}\right\}$$

Move 3.3 A matched pair is deleted and two unmatched records are paired

A matched pair (i, j) is selected at random from the set of matched records according to current configuration Z with equal probability. An unmatched pair (k, l) is selected at random from the set of unmatched candidate pairs with equal probability. The acceptance probability for the M-H algorithm is

$$\min\left\{1, \frac{Pr(\gamma_{kl}|M)Pr(\gamma_{ij}|U)n_M}{Pr(\gamma_{ij}|M)Pr(\gamma_{kl}|U)(n_1 - n_M)(n_2 - n_M)}\right\}$$

A.3. **Notes on computation/implementation.** Following the advice of (Gelman ref), all probabilities for the algorithm above are computed first in logs and later exponentiated to calculate the jump probability to avoid numerical overflow or underflow.

Unless otherwise noted, simulations were conducted on Adroit computing, with this machine.