

# 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.

The “merging” problem appears in many fields – statistics, computer science, operations research, and epidemiology – and under many names – record linkage, data linkage, entity resolution, instance identification, de-duplication, merge/purge processing, and entity reconciliation. In econometrics, research on merging belongs to the field of “data combination,” which is becoming increasingly important as the availability and use of large administrative datasets rises across the profession (?).

While there are many record linkage techniques, there is little understanding of how merging decisions bias inferences, nor advice to economists on how they should handle data combination issues in practice. In this paper I study hierarchical Bayesian record linkage models to learn how to incorporate multiple sources of uncertainty that may arise at different stages of an econometric analysis.

In the following sections, I introduce two models based on ? and ? – the Mixture Model (MM) and Beta Record Linkage (BRL) approaches. I then compare the performance of the two methods using a variety of simulated datasets. My preliminary results suggest that although mixture models are faster to converge, beta record linkage is superior if a one-to-one matching is desired. Extensions to this project will aim to apply these techniques to real data, and explore how to incorporate the posterior estimates of uncertainty about matches into subsequent analyses.

## 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 ?, 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* ( $U$ ):

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

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*,

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

where each comparison field  $\gamma_{ij}^\ell$  may be binary-valued, as in “ $i$  and  $j$  have the same birthday,” or use levels to account for partial agreement between strings (see ?, 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 ?.

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

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

where  $P(\gamma_{ij}|M)$  and  $P(\gamma_{ij}|U)$  are the probabilities of observing the pattern  $\gamma_{ij}$  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  $\gamma_{ij}$ ,

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

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 ?, 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 (?). Given the usefulness of the quantities in (2), the next sections will introduce two methods for estimating them.

### 3. SIMPLIFYING ASSUMPTIONS

Let  $\mathbf{\Gamma} \equiv \{\gamma_{ij} : (i, j) \in X_1 \times X_2\}$  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 (?). 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,

$$\begin{aligned}\gamma_{\mathbf{ij}} &| M \sim \text{Dirichlet}(\delta_{\mathbf{M}}) \\ \gamma_{\mathbf{ij}} &| U \sim \text{Dirichlet}(\delta_{\mathbf{U}})\end{aligned}$$

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  $\ell$  are defined so that  $\gamma_{ij}^\ell$  are independent across  $\ell$  conditional on match status. This implies:

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

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

? 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  $\ell$  (and global parameters that do not vary by block, if using blocked records), a convenient prior distribution is the product of independent Beta distributions,

$$\begin{aligned}p_M &\sim \text{Beta}(\alpha_M, \beta_M) \\ p_{M\ell} &\sim \text{Beta}(\alpha_{M\ell}, \beta_{M\ell}), \ell = 1, \dots, L \\ p_{U\ell} &\sim \text{Beta}(\alpha_{U\ell}, \beta_{U\ell}), \ell = 1, \dots, L\end{aligned}$$

For  $i = 1, \dots, n_1 \in X_1, j = 1, \dots, n_2 \in X_2$  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:

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

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

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

Based on these ideas, ? proposed a Bayesian version of record linkage for the mixture model approach, that uses a Gibbs Sampling scheme<sup>1</sup> for simulating the posterior distribution of  $I, (p_M, p_{M\ell}, p_{U\ell})$ .

**4.2. Beta Record Linkage.** As noted by ?, 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 ?? certainly violates this, since the components of  $\mathbf{I}$  are sampled independently in every iteration.

When the goal is one-to-one matching, ? 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  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_{n_2})$ , 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

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

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

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

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

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

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

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

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

To estimate the model, I use the Gibbs Sampling scheme outlined in ?, which I have reproduced in Appendix ??. It is very similar to the Gibbs Sampler for estimating the mixture model, however

---

<sup>1</sup>See Appendix ??

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 ?? and ?. 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.

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

**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. For the  $n_M$  matching record pairs, I generated  $\gamma_{ij}$  according to

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

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

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

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

I must emphasize that  $p_M^*$  controls the amount of overlap between the two files,  $X_1$  and  $X_2$ , so that  $p_M^*$  may be different from  $p_M$  defined in (1). For example, when  $p_M^* = 0.9, n_1 = n_2 = 10$ , I generated a dataset where 9 out of 10 files in  $X_2$  has a match in  $X_1$ . In the mixture model framework, these parameter values imply  $p_M = 0.09$ , since there are 9 record pairs that refer to matches out of 100 possible candidate pairs. In the beta record linkage model,  $p_M^*$  and  $p_M$  both refer to the proportion of records in  $X_2$  that have a match. This bad parameterization makes comparison across posterior estimates of  $p_M$  between models challenging.<sup>2</sup>

<sup>2</sup>Admittedly, this is a significant mistake on my part that I realized only after analyzing the results.

**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 ?:

- (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  $p_M \leq \frac{\min\{n_1, n_2\}}{n_1 n_2} = \frac{1}{n_1}$ , 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  $\ell$ . 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 of the mixture model and beta record linkage Gibbs samplers. Prior to submitting these jobs to the cluster, I 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  $(L = 3, n_1 = n_2 = 20)$ .

## 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  $p_{U\ell}$ .

**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 ?? changes  $\mathbf{Z}$  incrementally, adding, deleting, or swapping one record pair at a time. With just

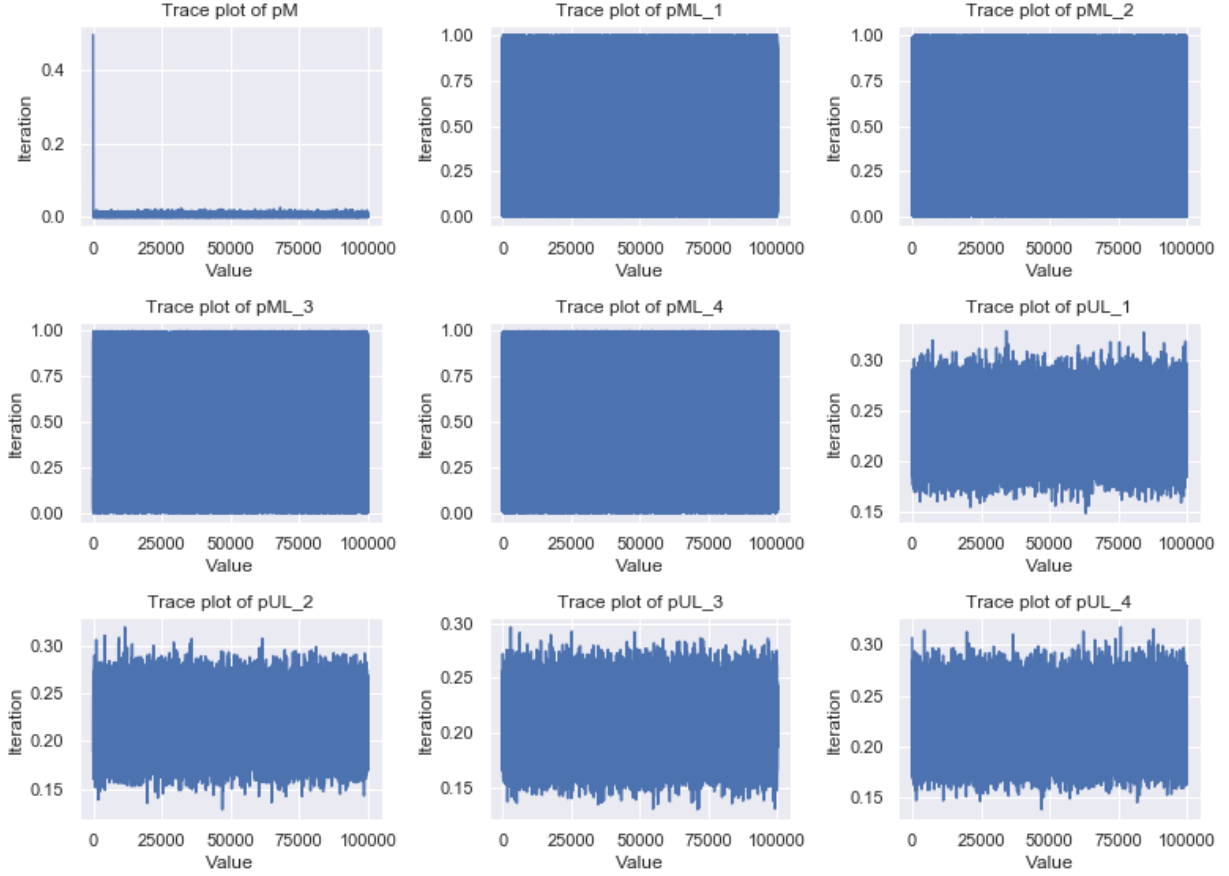


FIGURE 1. Trace plots for 100,000 draws of  $(p_M, p_{M\ell}, p_{U\ell})$  with true values of  $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (4, 20, 20, 0.5, 0.9, 0.2)$  and no burn-in

$n_1 = n_2 = 10$ , 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

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

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,

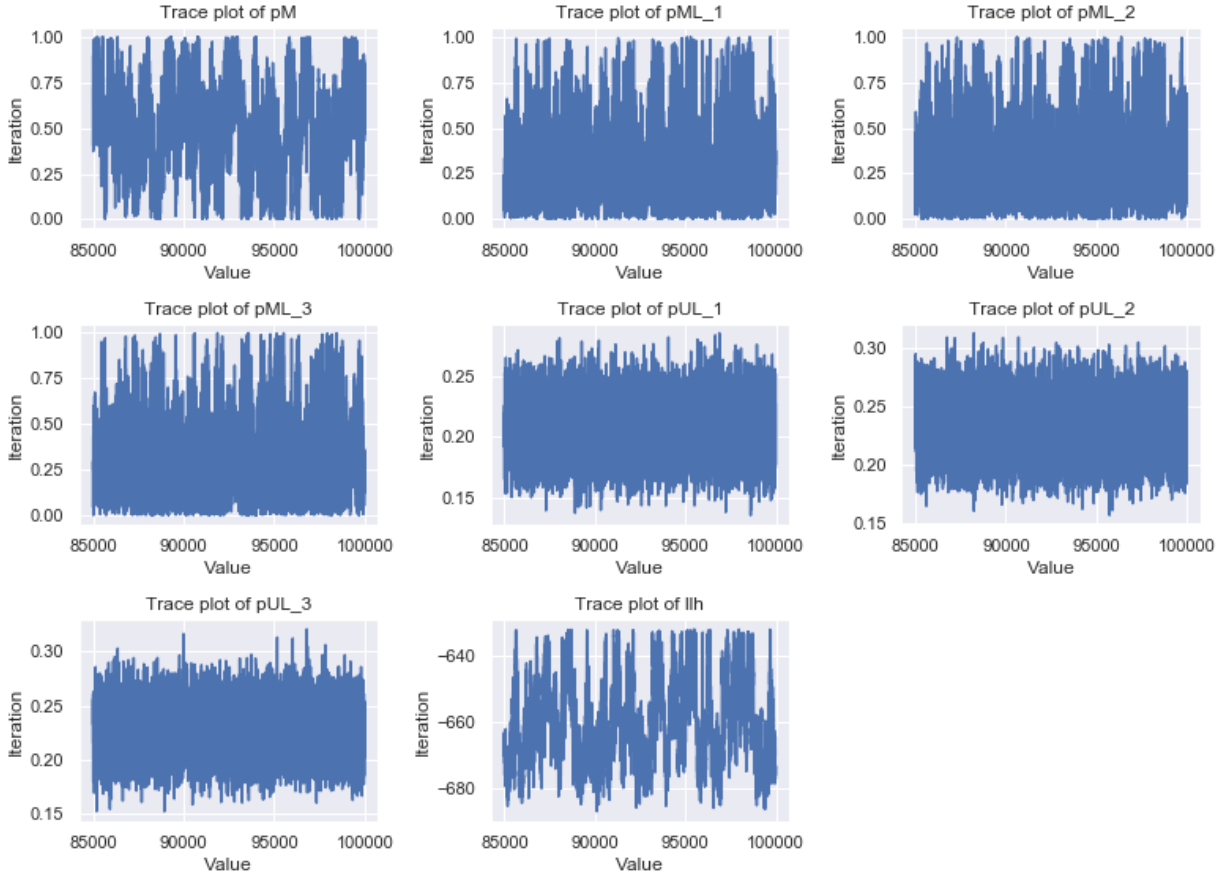


FIGURE 2. Trace plot of draws 85,000-100,000 for  $(p_M, p_{M\ell}, p_{U\ell})$  and the likelihood of  $Z$  from a Beta Record Linkage model with true parameters  $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (5, 20, 20, 0.5, 0.9, 0.2)$

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 posterior distributions for  $\mathbf{p}_{M\ell}$  are different between models, even though they are updated according to the same conditional distribution. The only way that the posterior of  $p_{M\ell}$  changes is via the comparison fields  $\gamma_{ij}^\ell$  of the record pairs currently assigned as a match. Since the mixture model almost never assigns record pairs as matches (because  $p_M \approx 0$ ), the posterior distribution of  $p_{M\ell}$  is updated only a very small amount. The posterior distribution of  $p_M$  does not collapse to 0 in the beta record linkage model, and so  $p_{M\ell}$  does get updated, however the distribution skewed significantly toward 0. I think this may be an artifact of my Gibbs sampler, which proposes moves in  $\mathbf{Z}$  at random, but future investigation is necessary to be sure.

### 6.3. Posterior match probabilities.



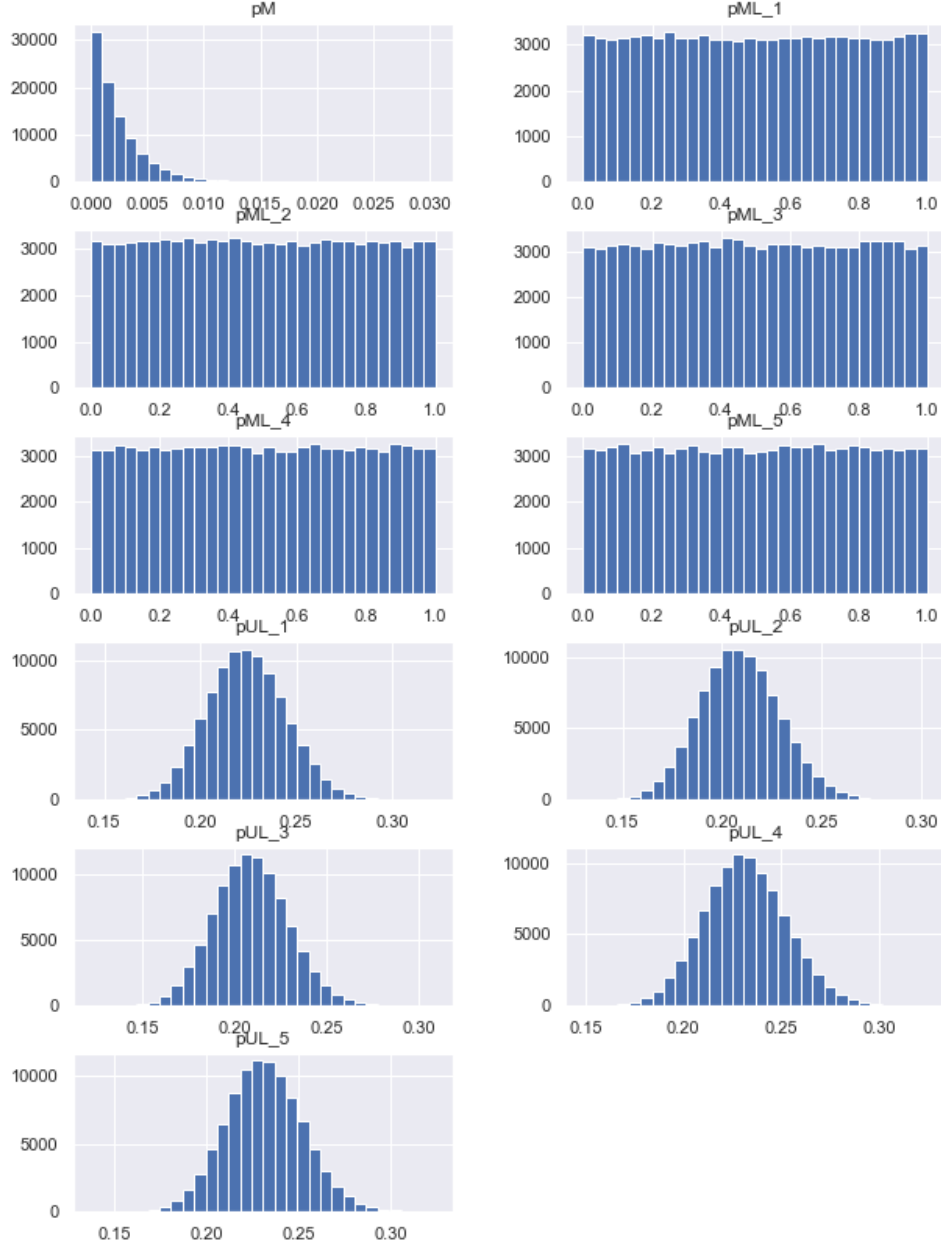


FIGURE 3. Histogram of 100,000 draws (burn-in = 5,000) from mixture model for all parameters with true parameter values  $(L, n_1, n_2, p_M, p_{ML}, p_{UL}) = (5, 20, 20, 0.5, 0.9, 0.2)$  and independent, flat priors

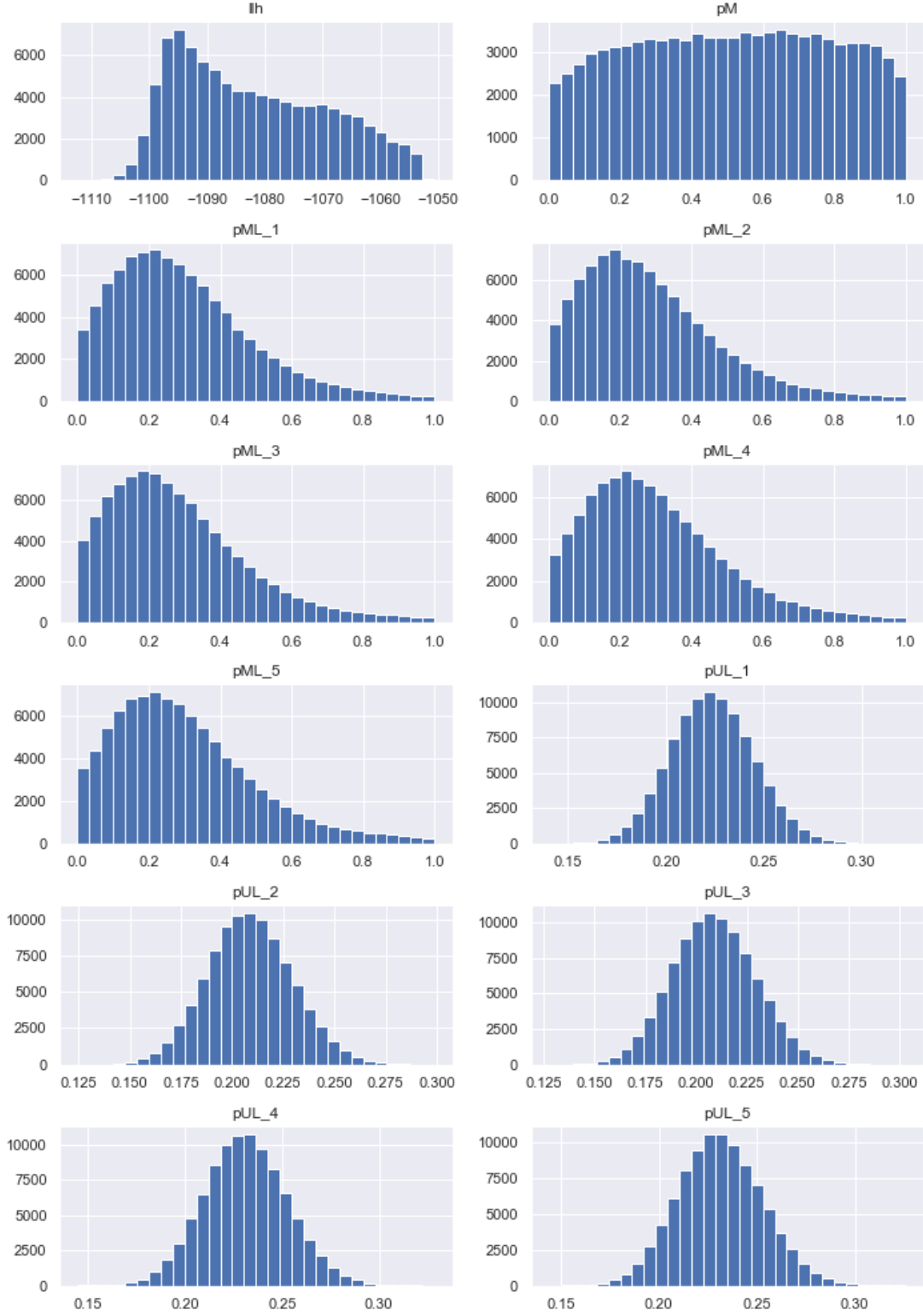


FIGURE 4. Histogram of 100,000 draws (burn-in = 5,000) from Beta Record Linkage model for all parameters and likelihood of  $Z$  with true parameter values  $(L, n_1, n_2, p_M, p_{ML}, p_{UL}) = (5, 20, 20, 0.5, 0.9, 0.2)$  and independent, flat priors

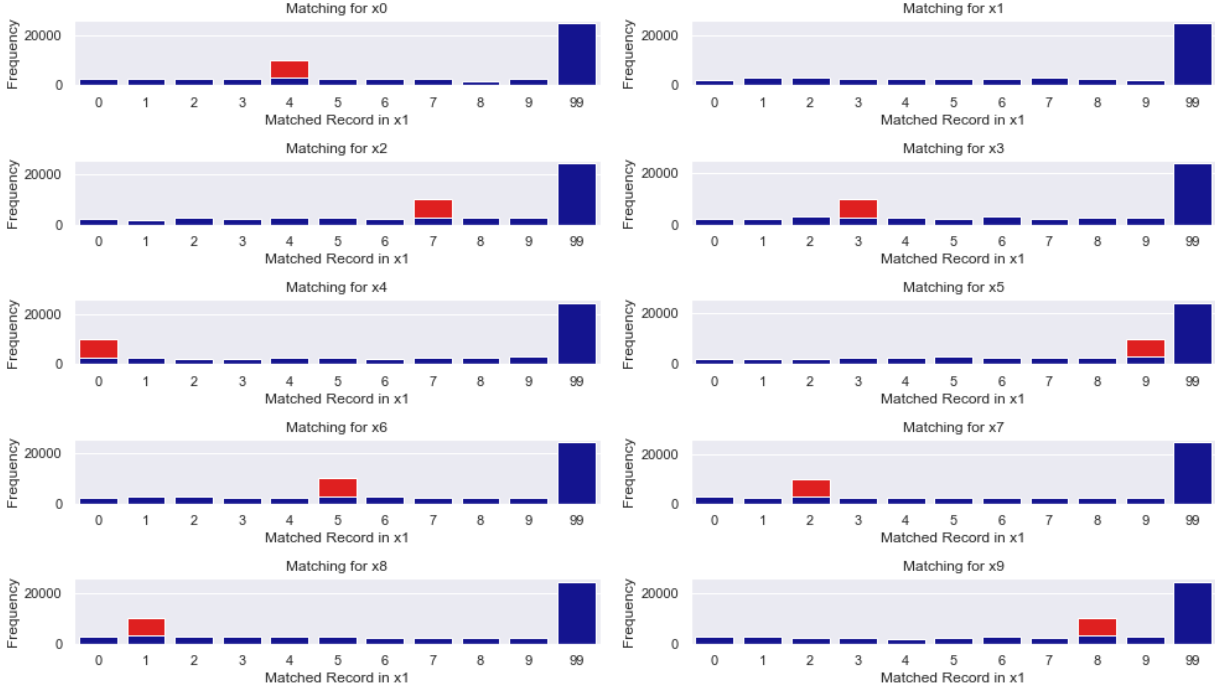


FIGURE 5. Histogram of  $Z$  draws for bipartite matching from a Beta Record Linkage model with true parameters  $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (3, 10, 10, 0.9, 0.9, 0.2)$

6.3.1. *Mixture Model.* For a candidate pair  $(i, j)$ , the posterior probability of  $(i, j) \in M$  is

$$P(I_{ij} = 1 | p_M, p_{M\ell}, p_{U\ell}) = \frac{1}{K} \sum_{k=1}^K I_{ij}^{(k)} \quad (12)$$

where  $k$  is the iteration number from the Gibbs sampler (post-convergence). If one-to-one matches are desired, one must either (1) designate all candidate pairs whose posterior match probability exceeds a certain threshold as matches, or (2) use a linear program to enforce one-to-one matching.

Across all applications, the values of  $I_{ij}$  are virtually identical for all record pairs, and so I do not calculate the posterior match probabilities based on mixture models.

6.3.2. *Beta record linkage.* The estimated posterior probability of  $(i, j) \in M$  according to the beta record linkage model 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 ??). Given that the chain has not converged, I am hopeful that with more draws (or a better Gibbs sampler), the chains would converge to the correct values.

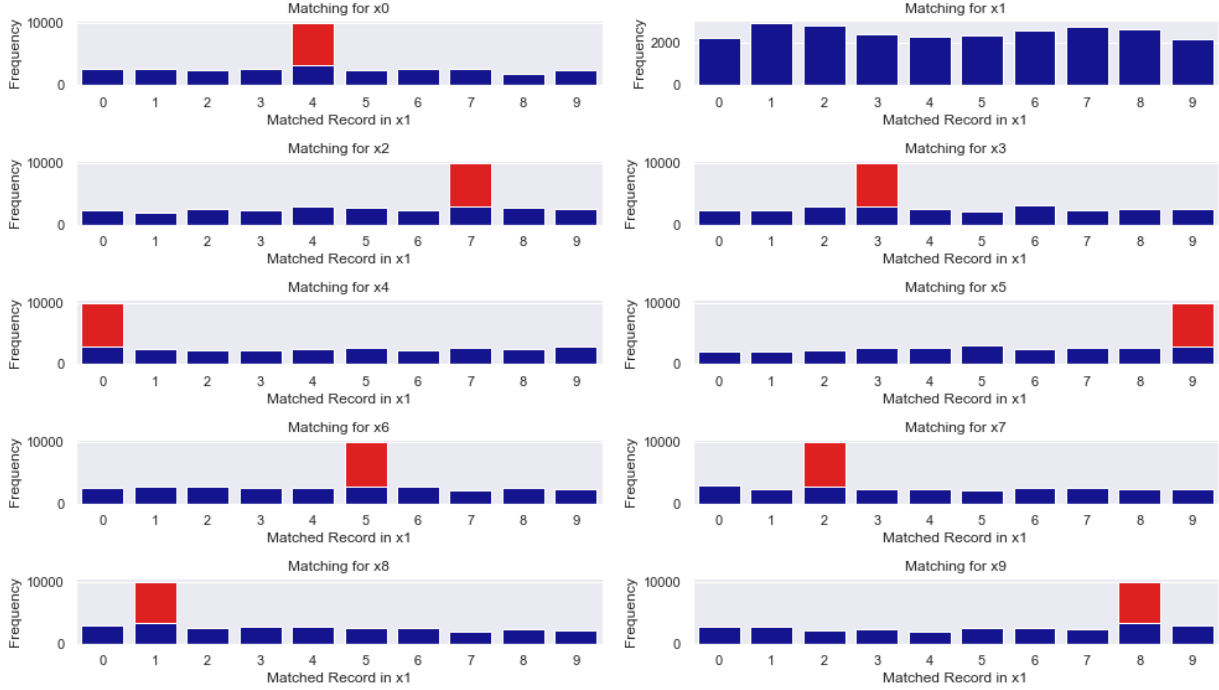


FIGURE 6. Histogram of  $Z$  draws for bipartite matching (conditional on being matched) from a Beta Record Linkage model with true parameters  $(L, n_1, n_2, p_M, p_{M\ell}, p_{U\ell}) = (3, 10, 10, 0.9, 0.9, 0.2)$

## 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 criticism about mixture models and clustering techniques in general, this label degeneracy produced confusing results for even the most simple datasets I tested.

Another disadvantage of the mixture model approach presented here is that  $\mathbf{I}$  potentially matches a file  $i \in X_1$  with multiple files in  $X_2$  and vice versa. Even if the mixture model is fitted with a one-to-one constraint, using the “optimal” decision rule from ? may lead to many-to-many assignments (?). One-to-one matches must be assigned post-processing using a linear sum assignment program that uses the estimated log-likelihood ratios  $w_{ij} = \log \left( \frac{P(\gamma_{ij}|M)}{P(\gamma_{ij}|U)} \right)$  as weights, which is challenging when the weights themselves are uncertain.

These issues aside, results from my paper show that the mixture model chains converged quickly (both in terms of number of iterations and in time spent running the Gibbs sampler), and that the posterior distributions for the identified variables were reasonably accurate. Given these advantages, the mixture model approach may serve as an easy-to-implement counterpart to deterministic record linkage methods.

**7.2. Beta Record Linkage.** ? uses beta record linkage to match files with sizes  $n_1 = 4,430, n_2 = 1,324$  which gives a total 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 about this claim, however I am optimistic that if I improve my Gibbs sampler I will achieve better results. For computational simplicity, I implemented a Gibbs sampler which chooses at random whether to add, drop, or switch a matching record pair in  $\mathbf{Z}$  each iteration. I think that implementing some of the more computationally intensive steps recommended in ? will help the chains converge after fewer iterations.

## 8. CONCLUSION

My motivation for studying Bayesian record linkage techniques was to learn how to incorporate uncertainty produced by merging files with imperfect identifiers into subsequent econometric analyses. While I have not yet begun to explore this issue directly, I studied and implemented two Bayesian approaches to record linkage, which sharpened my computational skills and deepened my understanding of MCMC techniques and hierarchical models. I also gained a greater appreciation for applying economic reasoning to statistical analyses, that was spurred by a quote that came up in my research:

“Every gain which is achieved by a superior record linkage procedure must be justified by the cost of implementing that procedure.” – Anonymous Referee

In other words, economics applies even to econometric modeling – the marginal benefit should exceed its marginal cost.

## 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.
- 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.
- 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.
- 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 (?)

- (1) Specify parameters for the prior distributions. Choose initial values of  $(p_M^{(0)}, p_{M\ell}^{(0)}, p_{U\ell}^{(0)})$  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  $(p_M^{(k)}, p_{M\ell}^{(k)}, p_{U\ell}^{(k)})$ , draw  $I_{ij}^{(k+1)}$  for each  $(i, j)$  candidate pair as an independent draw from a Bernoulli distribution with parameter

$$Pr(I_{ij}^{(k+1)} = 1 | \gamma_{ij}) = Pr(M | \gamma_{ij}) = \frac{p_M^{(k)} Pr(\gamma_{ij} | M, p_{M\ell}^{(k)})}{Pr(\gamma_{ij} | p_M^{(k)}, p_{M\ell}^{(k)}, p_{U\ell}^{(k)})} \quad (13)$$

where

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

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. Convergence of the algorithm can be monitored using the appropriate convergence criteria from ? and ?.

## A.2. Gibbs sampler for bipartite matching (?)

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

$$Pr(n_m, Z | \gamma, p_{M\ell}, p_{U\ell}, p_M, \alpha, \beta) \propto Pr(n_m | p_M) Pr(Z | n_M) Pr(\gamma | Z, \{p_{M\ell}, p_{U\ell}\}_{\ell=1}^L, p_M, \alpha, \beta) \quad (14)$$

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.

**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:

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

**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:

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

**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

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