Scaling Bayesian Probabilistic Record Linkage with Post-Hoc Blocking: An Application to the California Great Registers

Brendan S. McVeigh, Bradley Spahn, and Jared S. Murray

July 2018

Abstract

Probabilistic record linkage (PRL) is the process of determining which records in two databases correspond to the same underlying entity in the absence of a unique identifier. Bayesian solutions to this problem provide a powerful mechanism for propagating uncertainty due to uncertain links between records (via the posterior distribution). However, computational considerations severely limit the practical applicability of existing Bayesian approaches. We propose a new computational approach, providing both a fast algorithm for deriving point estimates of the linkage structure that properly account for one-to-one matching and a restricted MCMC algorithm that samples from an approximate posterior distribution. Our advances make it possible to perform Bayesian PRL for larger problems, and to assess the sensitivity of results to varying prior specifications. We demonstrate the methods on a subset of an OCR'd dataset, the California Great Registers, a collection of 57 million voter registrations from 1900 to 1968 that comprise the only panel data set of party registration collected before the advent of scientific surveys.

1 Introduction

Probabilistic record linkage (PRL) is the task of merging two or more databases that have entities in common but no unique identifier. In this setting linking records typically must be done based on incomplete information – attributes of the records may be incorrectly or inconsistently recorded, or may be missing altogether. Early development of PRL methods was motivated by applications in merging vital records and linking files from surveys and censuses to provide estimates of population totals (Newcombe et al., 1959; Fellegi and Sunter, 1969; Jaro, 1989; Copas and Hilton, 1990; Winkler and Thibaudeau, 1991). Recent applications of PRL cover a wide range of problems, from linking health care data across providers (Dusetzina et al., 2014; Sauleau et al., 2005; Hof et al., 2017) and following students across schools (Mackay et al., 2015; Alicandro et al., 2017) to estimating casualty counts in civil wars (Betancourt et al., 2016; Sadinle, 2017). In all these applications record linkage is uncertain (probabilistic) and this uncertainty should be propagated through to subsequent statistical analyses.

Bayesian approaches to PRL provide an appealing framework for uncertainty quantification and propagation via posterior sampling of the unknown links between records. Bayesian methods have been deployed in a similar range of applied problems: Capture-recapture or multiple systems estimation, where the quantity of interest is a total population size (Liseo and Tancredi, 2011; Tancredi et al., 2011; Steorts et al., 2016; Sadinle et al., 2014; Sadinle, 2017), linking healthcare databases to estimate costs (Gutman et al., 2013), and merging educational databases to study student outcomes (Dalzell and Reiter, 2016).

One drawback of Bayesian approaches to PRL is their computational burden – in the best of circumstances Bayesian inference can be computationally demanding, and making inference over a large combinatorial structure (the unobserved links between records) is particularly difficult. Computational considerations have largely limited Bayesian inference for PRL to small problems, or large problems that can be made small using clean quasi-identifiers through pre-processing steps known as indexing or blocking. For example, we may only consider sets of records as potential matches if they agree on one or more quasi-identifiers, such as a geographic area or a partial name match.

However, these pre-processing steps can lead to increased false non-match rates if the quasi-identifiers are subject to error. Many datasets lack the requisite clean quasi-identifying variables to make aggressive indexing or blocking feasible. Our application in this paper is one such example: Using Bayesian PRL we link extracts of the California Great Registers (historical voter registration rolls from the early 20th century). Like many historical datasets, the Great Registers contain few attributes available to perform linkage, all of which are subject to nontrivial amounts of error.

In this paper we introduce an approach to approximate Bayesian inference for PRL that is model agnostic and can provide samples from posterior distributions over record links between databases of hundreds of thousands of records in under two days, even in settings where traditional indexing or blocking is difficult to implement. We realize these gains using a data-driven approach we call "post-hoc blocking" to limit attention to distinct sets of record pairs where there is significant ambiguity about true match status. Unlike methods for calibrating traditional indexing or blocking schemes, post-hoc blocking requires no known sets of "true" links and non-links to implement (although our approach can utilize this data if available).

The paper proceeds as follows: Section 2 collects background material and reviews model-based approaches to record linkage. Section 3 introduces post-hoc blocking and describes the approach in generality. Section 4 introduces a new estimation technique for generating point estimates of links and efficiently obtaining inputs used in post-hoc blocking. Section 5 examines the performance of our methods in real and simulated data. Section 6 applies our methods to an extract from the Great Registers, comparing it to a recent proposal in the political science literature (Enamorado et al., 2017). Section 7 concludes with some discussion and directions for future work.

2 Approaches to Model-Based Probabilistic Record Linkage

Early approaches to probabilistic record linkage (such as the seminal work of Fellegi and Sunter (1969), described below) take what might be called a model-based clustering approach to record linkage were they developed today. This remains a popular approach. While the models and modes of inference have changed, the basic idea is the same: Each record pair corresponds to either a true match or a true non-match, and the goal in PRL is to use observed data to infer the latent match status for each pair of records under consideration. In this paper we consider matching two files which have no duplicates, a special but important use case of probabilistic record linkage. In this case a record from one file can match at most one record from the other. The one-to-one matching assumption is often useful even when the input files are not exactly deduplicated, as it provides important regularization during modeling.

To fix notation, suppose we have two collections of records, denoted A and B, containing n_A and n_B records respectively. Records $a \in A$ and $b \in B$ are said to be "matched" or "linked" if they refer to the same underlying entity. In this case the latent true match status for each record pair can be represented by

ID	Surname	Age	County
a_1	Williams	33	Alameda
a_2	Smith	24	Alameda
a_3	Jones	47	Santa Cruz

(a) Data A

ID	Surname	Age	County
b_1	Jonnes	45	Santa Cruz
b_2	Sauter	22	Napa
b_3	Wiliams	35	Alameda

(b) Data B

Record Pair	Surname	Age	County
(a_1, b_1)	7	12	1
(a_1, b_2)	8	11	1
(a_1, b_3)	1	2	0

(c) Record Comparisons

Figure 1: In (a) and (b) we show simple examples records to which PRL can be applied. (c) then shows comparisons.

an $n_A \times n_B$ binary matrix C, where

$$C_{ab} = \begin{cases} 1 & \text{if record } a \text{ matches record } b \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Since there are no duplicated records in A or B, the matching between the two files must be one-to-one and the row and column sums of C must all be less than or equal to one. Our goal is to infer C using observable attributes of the records.

2.1 Comparing records for PRL

The matrix C describing the linkage structure between the two files is unobserved. However, for each record we obtain a set of attributes that partially identify the individual to whom the record corresponds. Common examples include names, addresses, or other demographic information. These features serve to weakly identify the individual to whom the record belongs – intuitively, pairs of records that appear similar are more likely correspond to true matches ($C_{ab} = 1$).

A popular approach to model-based PRL is to begin by generating comparisons between record pairs on these attributes. These record comparisons then constitute the observed "data". The specific comparisons used can be tailored to the specific features available – for example, we might use a similarity score or the edit distance between two names, or the absolute difference between two dates of birth. Figure 1 provides a simple example of the reduction of record pairs to comparison vectors. The first two tables (a and b) are records from file A and file B, respectively. Panel (c) shows the constructed comparisons between the first entry in (a) and each entry in (b). Here surnames are compared using a Levenshtein (edit) distance, ages are compared using the absolute difference between the values, and counties are compared using a strict matching criterion (1 if the counties are identical and 0 otherwise). Many specialized comparison metrics have been developed, such as the Jaro-Winkler similarity score which was designed for comparing names in the presence of typographical error (Winkler, 1990). See (Christen, 2012, Chapter 5) for a detailed account of generating comparisons for different data types.

Suppose d separate comparisons are generated for each record pair. We collect these in a vector:

$$\gamma_{ab} = (\gamma_{ab}^1, \gamma_{ab}^2, \ldots, \gamma_{ab}^d). \tag{2}$$

and denote the collection of comparison vectors for all record pairs as Γ. These comparison vectors constitute the observed data in our model-based approach to PRL. Throughout we will assume that each γ_{ab}^j is a categorical measure of similarity or agreement, with higher levels corresponding to higher similarity. This is not the only approach to model-based PRL; one could alternatively model the error generation process and work directly with the observed field values (as in e.g. Tancredi et al. (2011, 2013); Gutman et al. (2013);

Steorts et al. (2015, 2016); Dalzell et al. (2017)). While this strategy has some advantages over working with comparison vectors it becomes unwieldy with more complicated fields like names and addresses.

2.2 The Fellegi-Sunter Framework

Fellegi and Sunter (1969) provide an early approach to PRL using comparison vectors. The basic idea is to model comparison vectors as arising from two distributions, one corresponding to true matching pairs and one corresponding to true *non*-matching pairs. Recalling that each element of the comparison vector takes on a discrete set of values, define

$$\begin{aligned} m(g) &= \Pr\left(\gamma_{ab} = g \mid C_{ab} = 1\right) \\ u(g) &= \Pr\left(\gamma_{ab} = g \mid C_{ab} = 0\right), \end{aligned} \tag{3}$$

for g ranging over the possible values of the comparison vector. These parameters are often referred to as "$m-$probabilities" and "$u-$probabilities" in the literature, a convention we adopt here.

2.2.1 The Fellegi-Sunter Decision Rule

Fellegi and Sunter (1969) provided a procedure for estimating C using the values of m and u, or estimates thereof. Define a weight for each record pair:

$$w_{ab} = \log\left(\frac{m(\gamma_{ab})}{u(\gamma_{ab})}\right). \tag{4}$$

This weight summarizes information about the relative likelihood of a record pair being a link versus non-link. Informally we can think of w_{ab} as a log-likelihood ratio statistic for testing whether γ_{ab} was generated by comparing matching or non-matching records. Intuitively, when a comparison vector g indicates significant agreement between the fields of two records we would expect $m(g) >> u(g)$, so if $\gamma_{ab} = g$ then $w_{ab} >> 0$. To generate a point estimate $\hat{C}$ of C Fellegi and Sunter (1969) provide a simple decision rule: A record pair (a, b) with $w_{ab} > T_\mu$ is called a match ($\hat{C}_{ab} = 1$), and a record pair with $w_{ab} < T_\lambda$ is called a non-match ($C_{ab} = 0$). Any remaining pairs have "indeterminate" match status and are evaluated manually. The thresholds T_μ and T_λ are set to simultaneously control the false positive rate μ (the probability a non-matching pair is called a match) and false negative rate λ (the probably a matching pair is called a non-match). Fellegi and Sunter (1969) prove that this procedure minimizes the size of the indeterminate set for given values of m and u.

2.2.2 Parameter estimation

Fellegi and Sunter (1969) proposed computing m and u from known population values for some special cases, or estimating m and u via the method of moments. However, it has become more common to estimate these parameters using the EM algorithm to maximize

$$L(m, u, \pi; \Gamma) = \prod_{(a,b) \in A \times B} \pi m(\gamma_{ab}) + (1 - \pi) u(\gamma_{ab}), \tag{5}$$

the likelihood under a simple mixture model:

$$\begin{gathered} \Pr(C_{ab} = 1) = \pi \\ \Pr(\gamma_{ab} = g \mid C_{ab} = 1) = m(g), \quad \Pr(\gamma_{ab} = g \mid C_{ab} = 0) = u(g) \end{gathered} \tag{6}$$

where each constructed comparison vector is treated as an independent observation (Winkler, 1988).

Regardless of the estimation strategy employed it is generally necessary to impose further structure on the $m-$ and $u-$probabilities. (A simple parameter counting argument shows that the model with unrestricted component probabilities is not identifiable.) A common assumption originating with Fellegi and Sunter (1969) is conditional independence of each comparison given the true match status, in which case the model in (6) reduces to a latent class model with two classes. Define

$$\begin{aligned} m_{jh} &= \Pr\left(\gamma_{ab}^{j} = h | C_{ab} = 1\right) \\ u_{jh} &= \Pr\left(\gamma_{ab}^{j} = h | C_{ab} = 0\right), \end{aligned} \tag{7}$$

for $1 \leq j \leq d$ and $1 \leq h \leq k_j$, where comparison j has k_j possible levels. Under conditional independence we have

$$\begin{aligned} m(g) &= \prod_{j=1}^{d} \prod_{h=1}^{k_j} m_{jh}^{\mathbb{1}(g_j = h)} \\ u(g) &= \prod_{j=1}^{d} \prod_{h=1}^{k_j} u_{jh}^{\mathbb{1}(g_j = h)}. \end{aligned} \tag{8}$$

Less restrictive models impose log-linear or other constraints on the $m-$ and $u-$ probabilities (Thibaudeau, 1993; Winkler, 1993; Larsen and Rubin, 2001).

2.2.3 One-to-one matching in the Fellegi-Sunter Framework

As originally constructed, neither the methods for inferring m and u nor the decision rule for generating an estimate of the matching structure C respect one-to-one matching constraints. In particular, in the Fellegi-Sunter decision rule, if $w_{ab} > T_\mu$ and $w_{ab'} > T_\mu$ then both would be declared links even though this would violate one-to-one matching. Jaro (1989) proposed a three-stage approach for adapting the Fellegi-Sunter decision rule to respect one-to-one matching. The first stage generates estimates of $\hat{m}$ and $\hat{u}$ by maximizing (5). The second stage generates C^*, an estimate of C that satisfies the following assignment problem:

$$\begin{aligned} C^* = \max_{C} & \sum_{a,b \in A \times B} C_{ab} \hat{w}_{ab} \\ \text{subject to} \quad & C_{ab} \in \{0, 1\} \\ & \sum_{b \in B} C_{ab} = 1 \qquad \forall a \in A \\ & \sum_{a \in A} C_{ab} \leq 1 \qquad \forall b \in B, \end{aligned} \tag{9}$$

where we assume that $n_A \leq n_B$. Despite its combinatorial nature this optimization problem can be solved relatively efficiently with standard linear programming techniques (solving assignment problems is discussed in Section 4.2.1). In the final stage, the matching estimate $\widehat{C}$ is obtained from C^* by setting $\hat{C}_{ab} = C^*_{ab} \mathbb{1}(\hat{w}_{ab} > \lambda)$, where λ plays the same role as in the FS decision rule.

While this procedure leads to an estimate of C that respects one-to-one matching, it is critically dependent upon good estimates of m and u. However, it has been observed empirically that failing to enforce the one-to-one constraint during estimation can lead to poor estimates of m and u (Tancredi et al., 2011; Sadinle, 2017). One-to-one matching could be enforced using a more realistic model for C that incorporates one-to-one constraints. However, the relatively simple and scalable EM algorithm for estimating m and u probabilities marginalizing over C no longer applies. In Section 4 we introduce a new estimation algorithm that could be

used for jointly estimating m, u, and C, but to date the most common solution to this problem appears to be full Bayesian modeling.

2.3 Bayesian Modeling for Probabilistic Record Linkage

Bayesian models for PRL naturally enforce one-to-one matching via support constraints in the prior distribution over C. Prior information on the matching parameters or the total number of linked records can be incorporated as well. But perhaps the strongest advantage of employing Bayesian modeling is that it naturally provides uncertainty over the linkage structure (through posterior samples of C) that can be propagated to subsequent inference.

Early approaches to Bayesian PRL utilized the same comparison-vector based model as in (6), typically replacing the independent Bernoulli prior distributions on the elements of C with priors that respect one-to-one matching constraints (e.g. (Fortini et al., 2001; Larsen, 2005)). Other Bayesian approaches avoid the reduction to comparisons by modeling population distributions of fields and error-generating processes directly (Tancredi et al., 2011, 2013; Steorts et al., 2015, 2016) or specify joint models for C and the ultimate analysis of interest, such as a regression model where the response variable is only available on one of the two files (Gutman et al., 2013; Dalzell et al., 2017). While we focus on comparison based modeling here, our post-hoc blocking procedure is model-agnostic and could be utilized in any of these models.

Most implementations of Bayesian PRL under one-to-one matching update C using local Metropolis-Hastings moves. At each step the algorithm proposes to either add or drop individual links, or swap the links between two record pairs, as described in e.g. Fortini et al. (2002); Larsen (2005); Green and Mardia (2006). A notable exception is Zanella (2017), in which the current values of matching parameters are used to make more efficient local MCMC proposals (often at significant computational cost). This is particularly effective when the records in both files can be partitioned or "blocked" such that links between records can only occur within elements of the partition (the blocks). With high-quality blocking variables some of these blocks can be small enough to enumerate, which admits simpler Gibbs sampling updates of the corresponding submatrices of C (Gutman et al., 2013; Dalzell and Reiter, 2016).

The MCMC steps for other model parameters are generally standard Gibbs or Metropolis-Hastings updates conditional on C. For all the Bayesian PRL models of which we are aware the most computationally expensive operations by far are the updates of the matrix C, due to its size and the local nature of the proposals. In the next section we propose a strategy for scaling Bayesian inference to much larger problems.

3 Scaling Probabilistic Record Linkage with Post-Hoc Blocking

Probabilistic record linkage is inherently computationally expensive; with files of size n_A and n_B there are $n_A n_B$ record comparisons to be made, which is prohibitively large for even seemingly modest file sizes. Comparisons such as string similarity metrics commonly used for names or addresses are much slower to compute than simple agreement measures and further add to the computational complexity of PRL. Regardless of the approach taken to PRL it is generally necessary to reduce number of record pairs under consideration during data pre-processing, a process known as *indexing* or *blocking*. We provides a short overview of traditional pre-processing steps before introducing our new strategy of *post-hoc blocking* for scaling Bayesian PRL.

3.1 Traditional approaches: Indexing, blocking and filtering

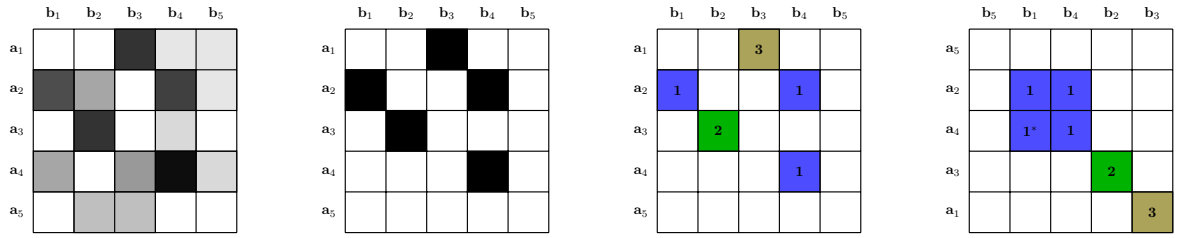
Traditional approaches to reducing the number of potentially matching record pairs can be separated into three categories: indexing, blocking, and filtering (Murray, 2016). Indexing refers to any technique for excluding a record pair from consideration before performing a complete comparison; for example, we might exclude record pairs from different counties, or any record pairs with years of birth that differ by more than five. A blocking scheme is an indexing scheme that requires candidate record pairs to agree on a single derived comparison, known as a *blocking key*. This induces a nontrivial partition of the records such that all links must occur within elements of the partition (the blocks). For example, indexing by requiring that matching record pairs agree on a county defines a blocking scheme, since matches can only occur within a county. Filtering refers to excluding record pairs *after* a complete comparison has been made. Little reference is made to filtering in the literature, but it is featured in many implementations of PRL (e.g. the U.S Census Bureau's BigMatch software (Yancey, 2002)).

Reducing the number of comparisons by blocking is particularly attractive since it effectively leads to a collection of smaller PRL problems that can be solved in parallel. However, it is relatively rare in applications to have a single blocking key that leads to an effective reduction in the number of candidate pairs without excluding many true matches in the process. In applications it is more common to combine the results of multiple blocking passes (see e.g. Winkler et al. (2010) for a high-profile example), which Murray (2016) calls indexing by disjunctions of blocking keys. For example, we might include all record pairs that come from the same county *or* match on the first three characters of their given name. In general indexing by disjunctions of blocking keys does not itself yield a blocking scheme.

3.2 Post-Hoc Blocking for Bayesian PRL

Given their computational complexity, Bayesian implentations of PRL tend to require stricter blocking or indexing than competing methods, such as the simpler Fellegi-Sunter approach. Stricter indexing increases the risk of false non-matches. Yet Bayesian approaches lead to better estimates of C under one-to-one matching when the records contain limited information, and provide a meaningful characterization of uncertainty in the linkage structure. To scale Bayesian inference we would be like to construct a high-quality blocking key from the available fields which excludes most of the obviously non-matching pairs. Critically, to obtain low false non-match rates and meaningful estimates of uncertainty we need to construct this blocking key without splitting sets of records across blocks when there is significant uncertainty about whether or not they are true matches. Given such a blocking key the MCMC algorithm could efficiently target the "interesting" areas of C, i.e. those areas with significant posterior variability, by fixing the elements of C outside of the blocks to zero.

We propose constructing such a blocking key using *post-hoc blocking*. The procedure is straightforward: First, if necessary, perform traditional blocking or indexing only to the extent necessary to make computing the comparison vectors feasible. Second, estimate matching weights or probabilities for each record pair. We refer to these generically as post-hoc blocking weights. The only criteria for these weights is that they reliably give high weight to true matching pairs and low weight to true non-matching pairs. Third, conduct one more blocking pass using the estimated weights to define the blocking key, reducing the number of record pairs just enough to make running an MCMC algorithm feasible. Using the weights to construct a blocking key incorporates all the information available in the comparison vector, with the goal of obtaining relatively compact blocks containing most of the pairs that are plausible matches.



(a) (b) (c) (d)

Figure 2: An example of *post-hoc blocking* (a) shows an example of estimated weights with dark cells corresponding to larger weights. (b) We construct a binary matrix where ones indicate weights above the threshold; this is the adjacency matrix of a bipartite graph. (c) We number and color the connected components of the graph; these are the basis of the post-hoc blocks. (d) We reorder the records to group them into the post-hoc blocks. Note that record pair (a_4, b_1) (labelled 1^*) is included to complete post-hoc block one, even though its weight was below w_0.

Figures 2a - 2d illustrate the process of post-hoc block generation. The rows and columns of the heatmaps correspond to records from file A and file B, respectively. Figure 2a shows a heatmap of the post-hoc blocking weights for each pair, with darker squares signifying larger weights. To generate a set of post-hoc blocks we begin by thresholding the matrix of weights at some value w_0. Figure 2b shows the thresholded matrix, where black boxes correspond to the record pairs with weights over the threshold.

At this point we have defined a bipartite graph between the records in files A and B; an edge is present between records a_i and b_j the weight for the record pair exceeds w_0. The sets of records corresponding to the nodes in each connected component are the *post-hoc blocks*; these are labelled in Figures 2c and 2d (the latter merely reorders the records to emphasize the block structure, and adds links below the threshold to complete the block structure). In this example, post-hoc blocking reduced the number of candidate pairs by nearly 80% while identifying a block of records that appear to have multiple plausible configurations (post-hoc block 1, the record pairs in blue).

The procedure for sampling from an approximate posterior distribution for C employing post-hoc blocking is summarized in Algorithm 1 below; implementation details follow.

Algorithm 1 Post-hoc Blocking with Restricted MCMC

Input: Comparison vectors Γ for a set of record pairs, weight threshold w_0

Output: Approximate posterior distribution for C and other model parameters

1. Estimate post-hoc blocking weights $\hat{w}_{ab}$.
2. Compute the matrix E where $e_{ab} = \mathbb{1}(\hat{w}_{ab} > w_0)$
3. Find the connected components of G, where G is defined as the bipartite graph with adjacency matrix E. The set of records corresponding to the nodes in each connected component are the post-hoc blocks
4. Run a standard MCMC algorithm, fixing $C_{ab} = 0$ for the record pairs outside the post-hoc blocks.

Step 1: Weight estimation. Clearly the performance of post-hoc blocking will depend on the quality of the weights. However for the purposes of post-hoc blocking they can be somewhat inaccurate, provided we

can set a low threshold w_0. But at a minimum these weights must avoid giving low weight to truly matching and ambiguous record pairs, while giving low weight to clearly non-matching pairs.

If labelled true matching and non-matching record pairs are available we could use these to predict the probability that the remaining pairs are a match using standard classification methods. These predicted probabilities will often fail to be calibrated; for example, when a record in file A has multiple plausible candidates in file B they may all receive high matching probabilities despite the one-to-one constraint. However, these records will be gathered into the same post-hoc block and the uncertainty in the matching structure will be represented in the posterior distribution.

Alternatively, in the absence of labelled record pairs we could use EM estimates of the Fellegi-Sunter weights in (4) as post-hoc blocking weights. This can work well in settings where there are several attributes available for matching and where most of the records in A also appear in B. When the two files have few fields in common, or there is significant error in the fields available, or when there is limited overlap between the files, EM-estimated weights can perform quite poorly (Tancredi et al., 2011; Sadinle, 2017). More reliable weights can be obtained by estimating the $m-$ and $u-$ probabilities while accounting for the one-to-one matching constraint, which we explore in Section 4.2 below.

Step 2: Threshold selection. Choosing the threshold w_0 requires balancing statistical accuracy (the quality of our posterior approximation) against computational efficiency. Larger values of w_0 are more likely to exclude true matching pairs, increasing the false non-match rate, and even excluding truly non-matching pairs which are not *obviously* non-matches risks misrepresenting posterior uncertainty. Increasing w_0 tends to increase bias. Decreasing w_0 decreases bias by admitting more record pairs, which tends to yield a smaller number of larger post-hoc blocks. Larger post-hoc blocks and more record pairs lead to increased computation time during MCMC; the most significant computational gains from post-hoc blocking accrue when the post-hoc blocks are small.

Given these considerations we should choose the smallest w_0 that leads to computationally feasible MCMC. What constitutes a "feasible" problem will naturally be context dependent, but a straightforward approach setting the threshold is to choose a maximum post-hoc block size and solve for the corresponding value of w_0.

Step 3: Finding post-hoc blocks. For a given threshold w_0, finding the post-hoc blocks is equivalent to finding the connected components of a bipartite graph. This is a well-studied problem with efficient solutions (Tarjan, 1972; Gazit, 1986). Post-hoc blocks at multiple levels of w_0 can be obtained recursively: As the threshold is increased we need only repeat the post-hoc blocking procedure within each current post-hoc block.

Step 4: Restricted MCMC Post-hoc blocking typically achieves massive reductions in scale relative to traditional blocking schemes for even moderate threshold values. Generally, a large number of small or singleton blocks are produced, in addition to a smaller number of larger blocks. This distribution of block sizes makes it possible design a restricted MCMC algorithm which mixes much more efficiently than standard approaches.

For very small blocks it is possible to enumerate all possible values of the corresponding submatrix of C and compute the corresponding unnormalized posteriors (conditional on the rest of C), allowing us to perform a Gibbs sampling update instead of a Metropolis step. Balancing increased computational time against decreased mixing time, this only makes sense for very small blocks. For example, there are only 7 possible linkage structures within a block of size 2×2 and 34 for a block of size 3×3, but for a 5×5 block there are 1,546 possible structures. Other implementations of Bayesian PRL have taken advantage of this

enumerability when a large number of high-quality traditional blocking fields are available (Gutman et al., 2013). However, post-hoc blocking is by design much more likely to produce a large number of small blocks than traditional blocking.

For moderately sized blocks, informative locally balanced Metropolis-Hastings proposals can be used instead of simple add/drop/swap proposals (Zanella, 2017). Zanella (2017) showed that locally balanced proposals can dramatically improve mixing over standard Metropolis-Hastings proposals in Bayesian PRL models. However, locally balanced proposals also become prohibitively costly for large blocks: For a $k_A \times k_B$ block contining L links at one iteration, the likelihood (up to a constant) must be computed $2(k_A k_B - L(L-1))$ times to perform a single locally balanced update. Zanella (2017) mitigated this issue by including random sub-block generation as part of the locally balanced proposal. But as the file sizes increase these completely random sub-blocks are increasingly unlikely to capture all or even many of the plausible candidates for each record in the block. In contrast, the post-hoc blocks are specifically constructed to capture all the plausible candidates for a given record in the same block.

The integration of post-hoc blocking with locally balanced moves and Gibbs updates produces an MCMC algorithm which mixes substantially faster for large problems than standard approaches. However, the posterior distribution obtained under post-hoc blocking is only an approximation, as the posterior probability of links between record pairs outside of the post-hoc blocks is artificially set to zero. In small problems where we can check against the full posterior the practical effect of this approximation seems to be limited (Section 5). In large problems, an approximation of this sort seems unavoidable – it is infeasible to run any current MCMC algorithm over datasets with hundreds of thousands of records generating hundreds of millions of candidate record pairs, even after indexing. However, post-hoc blocking and restricted MCMC can function well in this setting (Section 6).

3.3 Post-hoc blocking versus traditional blocking, indexing, and filtering

Post-hoc blocking combines ideas from indexing (specifically blocking) and filtering. However, it is not a special case of either. In traditional indexing and blocking, the goal is to avoid a complete comparison of the record pairs. As a result, the record pairs excluded by indexing are simply ignored and have no impact on model fitting. The same is typically true when filtering is employed – the record pairs that are filtered *after* a complete comparison have been made are ignored during model fitting, even though their comparison vectors have been generated.

In post-hoc blocking we use of all the generated comparison vectors by fixing $C_{ab} = 0$ for record pairs outside the post-hoc blocks. Even though they cannot be matched, data from these record pairs will be used to estimate the model parameters (for example, the $u-$parameters in Bayesian variants of the basic Fellegi-Sunter mixture model in (6). This makes efficient use of the comparison data, and avoids some of the more pernicious effects of filtering on subsequent parameter estimation described by Murray (2016).

4 Post-Hoc Blocking Weights under One-to-One Constraints

High-quality weights are important for efficient implementation of the post-hoc blocking algorithm. In applications of PRL to historical data we often have relatively few fields available to perform matching, many or all of which are subject to error. At the same time we know that the constituent files are at least approximately de-duplicated, so imposing a one-to-one matching constraint makes sense. We also have

limited or no labelled matching and non-matching record pairs with which to construct weights or validate results, suggesting the use of EM-estimated Fellegi-Sunter weights in post-hoc blocking.

However, we have observed that in this setting (one-to-one matching with a small number of noisy fields) the Fellegi-Sunter weights can be unreliable. We provide some evidence of this in Section 5. Similar observations have been made by Tancredi et al. (2011); Sadinle (2017). In this section we propose a new method for estimating post-hoc blocking weights under one-to-one matching constraints by enforcing the constraints during estimation.

4.1 Penalized Maximum Likelihood Weight Estimates

Jaro (1989)'s three-stage method for producing estimates of C (summarized in Section 2.2.3) involved three steps: Estimating the weights by maximum likelihood ignoring the one-to-one matching constraint, obtaining an optimal complete matching, and discarding matches with weights under a threshold to obtain a partial matching between the two files. Better estimates of the weights can be obtained by incorporating all three steps in a single-stage estimation procedure, simultaneously maximizing a joint likelihood in C, m, and u while penalizing the total number of matches.

The penalized likelihood takes the following form, where the last term in (10) is the penalty and the leading terms are the same complete data log likelihood corresponding to the standard two-component mixture model in (6):

$$
\begin{aligned}
l(C, m, u; \Gamma) &= \sum_{ab} [C_{ab} \log(m(\gamma_{ab})) + (1 - C_{ab}) \log(u(\gamma_{ab}))] - \theta \sum_{ab} C_{ab} && (10) \\
&= \sum_{ab} \log(u(\gamma_{ab})) + \sum_{ab} C_{ab} [\log(m(\gamma_{ab})) - \log(u(\gamma_{ab}))] - \theta \sum_{ab} C_{ab} \\
&= \sum_{ab} \log(u(\gamma_{ab})) + \sum_{ab} C_{ab} [w_{ab} - \theta] && (11)
\end{aligned}
$$

The form of the penalized likelihood in (11) shows that θ plays a similar role to T_μ in the FS decision rule; only pairs with $w_{ab} > \theta$ can be linked without decreasing the log-likelihood. This is also the unnormalized log posterior for C, m, and u under the a prior for C introduced in Green and Mardia (2006); the penalized likelihood estimate corresponds to a maximum a posteriori estimate under a particular Bayesian model.

Finding a local mode of (10) is straightforward via alternating maximization steps, which are iterated until the change in (10) is negligble:

1. Maximize C, given values of m and u. To maximize the penalized likelihood in C we need to solve the following assignment problem:

$$
\begin{aligned}
\max_{C} \quad & \sum_{a,b \in A \times B} C_{ab} \tilde{w}_{ab} \\
\text{subject to} \quad & C_{ab} \in \{0, 1\} \\
& C_{ab} = 0 \text{ if } \tilde{w}_{ab} = 0 \\
& \sum_{b \in B} C_{ab} \leq 1 \qquad \forall a \in A \\
& \sum_{a \in A} C_{ab} \leq 1 \qquad \forall b \in B.
\end{aligned} \tag{12}
$$

   where

$$
\tilde{w}_{ab} = \begin{cases} w_{ab} - \theta & w_{ab} \geq \theta \\ 0 & w_{ab} < \theta \end{cases} \tag{13}
$$

We discuss how to efficiently solve these thresholded assignment problems in Section 4.2.1.

2. Maximize m and u probabilities, given a value of C. These updates are available in closed form under the conditional independence model (Equation 8):

$$m_{jh} = \frac{n_{mjh} + \sum_{ab} C_{ab} \mathbb{1}(\gamma^j_{ab} = h)}{\sum_h n_{mjh} + \sum_{ab} C_{ab}} \tag{14}$$

$$u_{jh} = \frac{n_{ujh} + \sum_{ab} (1 - C_{ab}) \mathbb{1}(\gamma^j_{ab} = h)}{\sum_h n_{ujh} + \sum_{ab} (1 - C_{ab})}. \tag{15}$$

where the n's are optional pseudocounts used to regularize the estimates. (These terms correspond to an additional penalty, omitted from (10)-(11) for clarity.) We suggest their use in practice to avoid degenerate probabilities of zero or one. They are easy to calibrate as "prior counts" – i.e., n_{mjh} is the prior count of truly matching record pairs with level h on comparison j, and the strength of regularization is determined by $\sum_h n_{mjh}$ (with larger values implying stronger regularization).

Conceptually this optimization procedure is straightforward, but iteration to a global mode is not guaranteed. In general a global mode in all the parameters need not exist – for example, if a record in A has two exact matches in B then the penalized likelihood function will have at least two modes with the highest possible value. However, the values of the $m-$ and $u-$probabilities will be the same in both modes and these are the only objects of interest for defining weights. Of course it is also possible for an alternating maximization approach to get trapped in sub-optimal local modes. Our experience running multiple starts from different initializations suggests that this not common – that is, when we iterate to distinct local modes they tend to have similar values for the $m-$ and $u-$ probabilities.

4.2 Maximal Weights for Post-Hoc Blocking

The estimated $m-$ and $u-$ probabilities obtained via penalized likelihood maximum estimation can depend strongly on the value of the penalty parameter θ. In general higher values of θ correspond to lower numbers of matches, and one could potentially try to calibrate this parameter based on subject matter knowledge and prior expectations. However, rather than banking on our prior expectations we propose a more conservative approach: Rather than fixing a value of θ and obtaining weights for each pair, we vary θ over a range of values, obtain estimated weights for every value of θ, and take the maximum observed weight for each record pair as the post-hoc blocking weight. This obviates the need to calibrate θ and assigns relatively high weight to any record pair that is a plausible match candidate for *some* value of θ.

To define the sequence of values we suggest starting with $\theta = 0$ and then selecting successively larger penalty values. The actual sequence of penalty values can be chosen via a variety of different rules. A useful rule of thumb is that the next penalty in the sequence should be larger than the smallest weight in the previous solution, to ensure a change in the solution to the assignment problem. Specifying a minimum gap size between successive values of θ provides further control over computation time.

A naive implementation of maximal weight estimation is computationally intensive – standard algorithms for solving the assignment problems (such as the Hungarian algorithm, Kuhn (1955)) have worst-case complexity that is cubic in the larger of the two file sizes. Each step of the penalized likelihood maximization involves solving multiple assignment problems, and this must be repeated for each distinct value of θ. However, there are three features of our assignment problems that make them dramatically easier to

solve: The weight matrices involved are usually extremely sparse, exact solutions are often not necessary, and assignments from previous iterations can be used to effectively initialize the next iteration.

4.2.1 Solving Sparse Thresholded Assignment Problems

Given a set of estimated weights, Jaro (1989) suggested solving (16) by constructing a canonical linear sum assignment problem, which assumes that each record in A will be matched to some record in B. If $n_A < n_B$ Jaro (1989) does this by constructing an $n_B \times n_B$ augmented square matrix of weights $\check{w}$ and an augmented assignment matrix C and solving the following canonical *linear sum assignment problem* (LSAP):

$$\begin{aligned} \max_{C} & \sum_{a=1}^{n_B}\sum_{b=1}^{n_B} C_{ab}\check{w}_{ab} \\ \text{subject to} \quad & C_{ab} \in \{0,1\} \\ & \sum_{b=1}^{n_B} C_{ab} = 1 \qquad \forall a \in A \\ & \sum_{a=1}^{n_A} C_{ab} = 1 \qquad \forall b \in B, \end{aligned} \tag{16}$$

where $\check{w}_{ab} = \hat{w}_{ab}$ (the estimated weight) if $a \leq n_A$ and is otherwise set to the smallest observed weight or another extreme negative value. In a final step any matches with weights under a threshold are dropped, which necessarily removes any matches that correspond to augmented entries in C.

Unfortunately, in general this procedure will not lead to the estimate of C with the highest total weight assigned to the matched pairs. Figure 3 provides a simple counterexample. Figure 3a shows an example of estimated weights. Since the LSAP above makes a complete assignment, there are two feasible values of C that could be returned: Either a_1 matches b_1 and a_2 matches b_2, or a_1 matches b_2 and a_2 matches b_1. The latter matching (Figure 3b) provides the solution to the assignment problem above, because of the relatively large negative weight on the pair (a_2, b_2). But inspecting the weight matrix shows that the best *overall* matching – accounting for the subsequent thresholding – is obtained by linking a_1 and b_1, leaving a_2 and b_2 unmatched.

The solution to this problem is to incorporate the threshold into the maximization problem by setting any weights below the threshold to zero (and adding a constant to make the remaining weights positive if necessary). In the final step, any entry of C with a corresponding zero weight is dropped. Figure 3c shows the thresholded weight matrix, which leads to the correct solution (Figure 3d). This is how we construct the weight matrices during penalized likelihood estimation; see (13).

Adopting the formulation in (12) has the added benefit of making the assignment problem easier to solve. While relatively efficient algorithms exist for solving dense LSAPs, (e.g. the Hungarian algorithm (Kuhn, 1955)), they have a worst case complexity of $O(n^3)$ where $n = \max(n_A, n_B)$ (Jonker and Volgenant, 1986; Lawler, 1976). However, after thresholding our weight matrix will be very sparse. Indeed, depending on the degree of overlap between the two files there may be entire rows and columns of zeros – effectively reducing n and yielding an easier optimization problem.

Even greater benefits can be realized by partitioning the sparse weight matrix to derive many small optimization problems to be solved in parallel. Similar to our procedure for obtaining post-hoc blocks, we can employ graph clustering to separate the records into blocks (defined by the connected components of the weighted graph defined by the thresholded post-hoc blocking weights) such that links are only possible

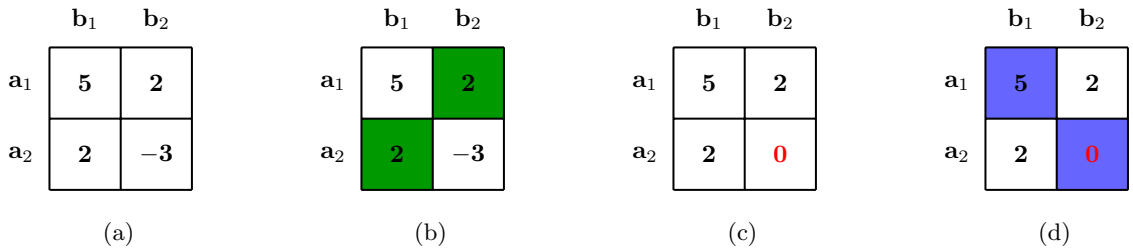


Figure 3: Simple assignment problem with and without threshold (a) shows an example of estimated weights. (b) Highlights the assignment that maximizes the assigned weights for the costs given in (a). (c) Adjusts the costs shown in (a) by soft-threshold at 0. (d) the maximal assignment solution if the soft-thresholded costs are used and zero cost assignments are then deleted. We note that the resulting assignment has a higher weight than the one given in (b).

within and not across blocks. This allows us to decompose the full assignment problem into a set of smaller problems that can be solved in parallel, as summarized in Algorithm 2.

Algorithm 2 Connected-Component Based Assignment Problem

Input: Soft-thresholded weight matrix $\widetilde{W}$ (from Eq (13))
Output: Estimate $\widehat{C}$ partial assignment with highest total weight

1. Find the connected components of the bipartite graph G which has edges between nodes a and b where $\tilde{w}_{ab} > 0$.
2. Solve the assignment problem for each component separately.
3. Merge assignment solutions.

Finding the connected components of a bipartite graph has computational complexity of $O(|E|+n_A+n_B)$ (linear time with respect to the number of edges in the graph, i.e. the number of nonzero weights after thresholding) (Tarjan, 1972). This is loosely bounded by $n_A n_B$ above. After partitioning the graph, computational demands are driven primarily by solving the LSAP corresponding to the largest connected component. Since the computational complexity of this step is at worst $O(k^3)$, with k being the maximum number of records from either file appearing in the component, we can obtain dramatic reductions in computational complexity by partitioning the original problem.

Practical performance is often much better than these worst-case complexity results might suggest. Many of the sub-matrices of $\tilde{W}$ corresponding to connected components will remain sparse. In computational studies many algorithms for solving LSAPs show substantially faster results on sparse problems (Carpaneto and Toth, 1983; Jonker and Volgenant, 1987; Orlin and Lee, 1993; Hong et al., 2016). In fact previous work suggests, but does not prove, that it may be possible to solve sparse assignment problems in linear time with respect to the number of edges (Orlin and Lee, 1993). The only case of proven complexity improvements that we are aware of is for auction algorithms (Bertsekas and Eckstein, 1988; Bertsekas and Tsitsiklis, 1989).

We adopt the auction algorithm for solving sparse LSAPs in all of our algorithms. In addition to its performance guarantees, the auction algorithm allows us to use previous solutions to specify initial values for new problems. This is useful in the iterative maximization problems in both the penalized maximum likelihood and the maximal weight procedure. Further, we have the option to stop the auction algorithm

Last	Sex	Edu	Count	EM Weight	Maximum Weight
1	1	1	25	5.31	6.09
1	0	1	8	3.71	3.36
1	1	0	13	-0.43	2.11
0	1	1	126	2.61	0.22
1	0	0	21	-2.02	-0.69
0	0	1	78	1.02	-2.63
0	1	0	601	-3.12	-3.84
0	0	0	658	-4.72	-6.70

Table 1: Maximum weights used for post-hoc blocking, and EM weights for comparison

early to save on computation time. This is useful in penalized likelihood estimation, where we need not necessarily find the optimal assignment in order to improve the objective function at each step. For a more complete overview of auction algorithms see Bertsekas and Tsitsiklis (1989); Bertsekas (1998, 1992).

5 Examples and Illustrations of Post-Hoc Blocking and Restricted MCMC

We consider two examples from the existing literature to illustrate the performance of post-hoc blocking with maximal weights. First, we re-analyze a small real dataset from the 2001 Italian census and a post-enumeration survey presented in (Tancredi et al., 2011). Second, we consider performance in a set of simulated datasets provided by Sadinle (2017) where the error rates and number of true matches vary substantially.

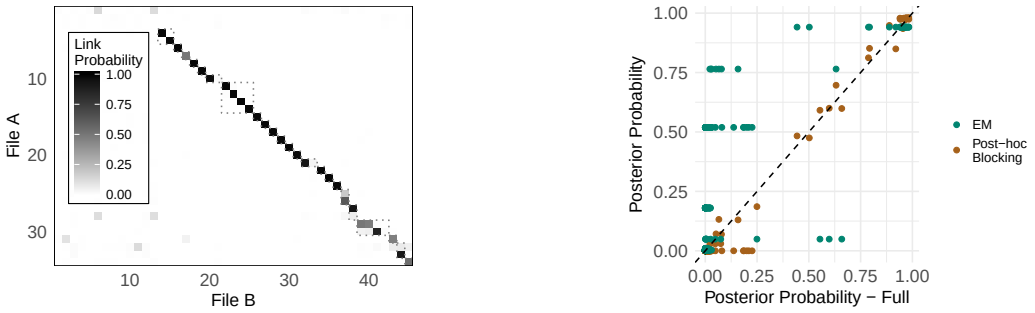
In both examples we use the Beta-bipartite prior with a uniform prior over the expected proportion of matches (Fortini et al., 2001, 2002; Larsen, 2005, 2010; Sadinle, 2017). We assume a conditional independence model for $m-$ and $u-$probabilities as in (8). Each vector of conditional probabilities is assigned a Dirichlet prior distribution; specific prior parameter values are noted below.

5.1 Italian Census Data (Tancredi et al., 2011)

The data in this example come from a small geographic area; there are 34 records from the census (file A) and 45 records from the post-enumeration survey (file B). The goal is to identify the number of overlapping records to obtain an estimate of the number of people missed by the census count using capture-recapture methods. This small scale example allows us to compare the results of estimation performed employing post-hoc blocking with the results from considering the full set of record pairs.

Each record includes three categorical variables: the first two consonants of the family name (339 categories), sex (2 categories), and education level (17 categories). We generate comparison vectors as binary indicators of an exact match between each field. We assume that $m_j \sim \text{Dir}(20, 3)$ and $u_j \sim \text{Dir}(3, 20)$ for $j = 1, 2, 3$ independently. These priors where chosen to contain modes near 0.9 and 0.1 respectively, with a reasonable degree of dispersion.

We estimated post-hoc blocking weights using the maximal weight procedure in Section 4.2. The resulting weights for each possible comparison vector are shown in Table 1, along with EM weights for comparison.



(a) (b)

Figure 4: (a) Post-hoc blocks overlayed on posterior link probabilities estimated via MCMC using all record pairs (b) Posterior probabilities from EM and restricted MCMC versus posterior link probability considering all record pairs.

Notable discrepancies are in gray. We see that the EM weights consider a record pair agreeing on sex and education alone to be a more probable match than a record pair agreeing on last name and sex. While we have no ground truth here, this seems unlikely. More striking is the EM weight assigned to record pairs that agree solely on education – its value of 1.02 under this model would suggest that these such a pair is more likely than not a true match. Overall the maximum weights seem to provide a more reasonable rank ordering of the comparison vectors.

To implement restricted MCMC we selected the post-hoc blocking threshold w_0 to restrict the size of the largest post-hoc block to fewer than 400 record pairs. This is achieved for any $w_0 \in (0.22, 2.11)$. The resulting post-hoc blocks contain only 53 of the 1530 possible record pairs. These are spread across 23 separate post-hoc blocks. Of the 23 post-hoc blocks, 15 contain only a single record pair, 5 contain 2 record pairs, the remaining three contain 4, 8, and 16 record pairs respectively.

We then run both an MCMC algorithm containing all 1530 record pairs and our restricted MCMC under identical model specifications. Results from both models are displayed in Figure 4a, with the post-hoc blocks overlayed. Nearly all of the posterior link density is contained within the post-hoc blocks, but a few pairs with modest posterior probability are omitted from the post-hoc blocks. (Lowering the threshold to capture these would have resulted in a single large block.)

In Figure 4b we compare the posterior link probability estimated by the full MCMC, our post-hoc blocking restricted MCMC, and posterior probability estimates as computed from the EM output. The full and restricted MCMC probabilities are quite similar, except the small cluster of points on the x-axis near the origin. These are points that had modest posterior probability – less than 0.25 – in the full MCMC but were excluded from the post-hoc blocks and assigned zero probability in the approximate posterior. Even in this small example we obtain a significant improvement in runtimes: Using identical implementations posterior sampling takes 18.7 seconds for the full MCMC algorithm versus 0.9 seconds when employing post-hoc blocks. This factor of 20 is almost certainly an understatement if we also consider the mixing time of the two chains – the restricted chain targets its moves carefully and tends to mix much faster.

The EM fit provides estimates of posterior probabilities, albeit posterior probablities that do not respect one-to-one matching constraints. These estimates do not align well with the MCMC output. This is in part due to the problematic weight estimates in Table 1. But the failure to account for one-to-one matching seems to play a larger role – in general we would expect ommitting the constraint to lead the posterior probability

estimates to be too high, which is what we see here – nearly all the EM posterior probabilities exceed the Bayesian estimates.

5.2 Simulated datasets (Sadinle, 2017)

To study the behavior of these algorithms with a known ground truth we utilize synthetic data generated by Sadinle (2017) for a simulation study. Each synthetic dataset comprises two files of 500 records ($n_A = 500$, $n_B = 500$) subsampled from one of 100 larger datasets. Each record contains four fields: given name, family name, age category and occupation category. Error rates were maniupated such that between one and three of the four fields for contains an error for each record. The degree of overlap also varied, (100%, 50% or 10%). Following Sadinle (2017) we generated comparisons as follows: Given name and family name are compared based on a discretized Levenshtein distance measure with four levels: exact agreement (distance of zero); mild disagreement, (0, .25]; moderate disagreement, (.25, .50]; and extreme disagreement, (.50, 1.0]. Age and occupation categories were compared using exact agreement.

For given name and family name we assign the $m-$vectors $\text{Dir}(4, 3, 2, 1)$ priors. For the two exact comparisons (age and occupation categories) we assign $\text{Dir}(2, 1)$ prior distributions. For all comparisons we use uniform priors on the $u-$probabilities. We implemented restricted MCMC by performing post-hoc blocking with maximal weights. The threshold w_0 was selected to be the smallest value for which the largest post-hoc block contains less than 2,000 record pairs. We then ran the MCMC algorithm for 10,000 steps – here a "step" consists of a Metropolis-Hastings proposal or Gibbs sampling update within each post-hoc block, so this rather conservative. The average runtime of our MCMC algorithm was around 30 seconds for each of the 900 simulations.

We compared the various methods via their point estimates of the linkage structure. To construct a point estimate of the linkage structure $\widehat{C}$, we set $\widehat{C}_{ab} = 1$ if the posterior probability is larger than 0.5 and 0 otherwise. This is the Bayes estimate under a variety of loss functions (Tancredi et al., 2011); Bayes estimates under more sophisticated loss functions that allow fo0r indeterminate decisions are given by Sadinle (2017). For each simulation we also estimate C by modifying Jaro (1989)'s procedure, estimating the weights via EM but solving the correct thresholded assignment problem (as discussed in Section 4).

In an attempt to mitigate the possibility of EM converging to poor local optima, we initialized the $m-$ and $u-$ probabilities in the EM algorithm in two ways. First with what seemed to be sensible values: $m-$ probabities of $(.35, .35, .25, 0.05)$ for given name and family name and $(.25, 0.75)$ for age and occupation categories and $u-$ probabilities of $(.01, .02, .07, 0.9)$ for given name and family name and $(.01, .99)$ for age and occupation categories. Second, in an attempt to initialize near a "good" mode we used the true match status of each record pair to derive the conditional probabilities directly. Results from both initializations were similar. When computing the EM point estimate we used $T_\mu = 0.0$ as in Sadinle (2017). This is probably a lower value than would typically be used in practice, but we maintain it here so our results will be comparable to Sadinle (2017).

Precision (proportion of the estimated matches that are false) and recall (proportion of true matches that are recovered) for the various approaches are shown in Figure 5. Overall the MCMC based estimate has somewhat lower recall but produces an estimate with much higher precision, particularly in scenarios where the percent overlap between record sets is low. This is due in part to the low threshold T_μ, but also to a general tendency of the EM + LSAP based estimate to link a large number of record pairs by failing to appropriately account for the one-to-one constraint. If a record in A has multiple plausible candidates in B, each will have a high EM-estimated weight. The EM-based approach will then tend to simply match the

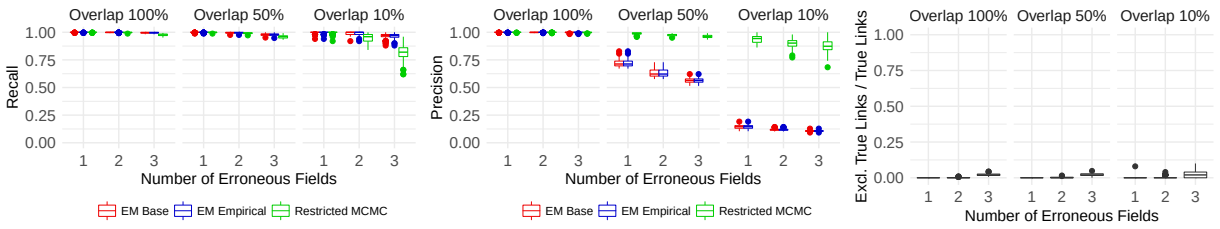


Figure 5: Recall (left), precision (center), portion of true links excluded from the post-hoc blocks (right)

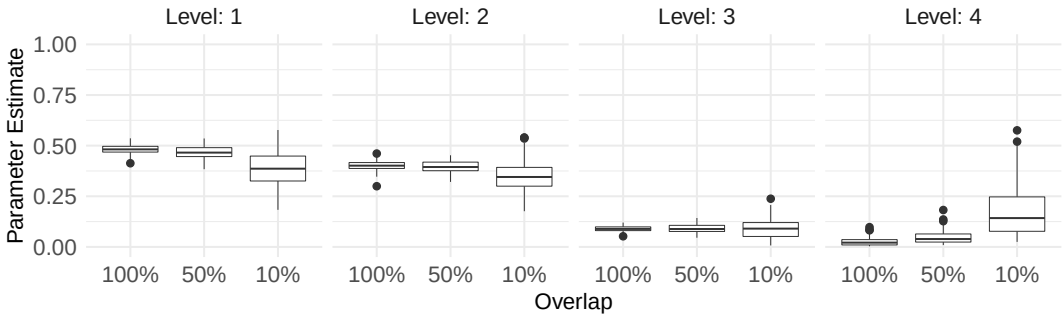


Figure 6: EM estimates of $m-$parameters for the given name field across simulated datasets with two errors. Level 1 is strong agreement, and level 4 is strong disagreement.

record pair with the largest weight. In contrast, since the Bayesian model enforces the one-to-one posterior matching constraint it will "spread" probability across the candidates, often dragging all their posterior probabilities below 0.5.

Many of the low-overlap simulated datasets yielded relatively poor estimates of the $m-$probabilities, similar to the Italian census dataset. Figure 6 shows the EM parameter estimates for the $m-$probability vector for given name (level 1 indicating strong agreement and level 4 indicating strong disagreement) across the simulated datasets with two errors per record. The estimates are highly variable in the low overlap setting. More troubling is that in the low overlap datasets, strong disagreement has a *higher* $m-$probability than moderate agreement on average, which is clear evidence of bias in the estimated parameters.

6 Linking the Great Registers: Alameda County Case Study

Beginning in 1900, each California county was required to publish a typeset copy of their voter registers in each election year. The California Great Registers, which contained the name, address, party registration and occupation of every registered voter, served as a record of the county's voters and as poll books on election day. The Great Registers provide a fine-grained tool for measuring the dynamics of partisan change over an especially interesting period of American history, the New Deal realignment. From 1928 to 1936, a substantial number of Americans switched their partisan allegiance from the Republicans (the party of Herbert Hoover) to the Democrats (the party of Franklin Roosevelt). While this change is known to have taken place at the macro-level, the Great Registers are the first individual-level dataset that follows this change. Though every California county published a register, we focus on Alameda county, where Oakland is located, as a case study.

Though the structure of the data is simple, transferring it from the printed page into digital format is challenging. Ancestry.com scanned and performed optical character recognition (OCR) the Great Registers, enabling use of the data by their subscribers for genealogical research. This process is only partially successful because the quality of the scan as well as the original organization of the page can make the OCR fail to produce recognizable text or can mistranscribe words and letters.

One quantity of interest to historians and political scientists is the frequency with which voters changed between the parties from 1932 to 1936, during Roosevelt's first term as president. Though voters rarely switch parties, this period featured the most dramatic and rapid change in partisanship in the twentieth century, making individual-level panel data from this period especially interesting. To make such a panel, we link records from the 1932 voter register to the 1936 register using name, address and occupation. Though party might be an informative field in making a match, it is withheld from the matching process so that our estimate of the key quantity of interest, the party switching rate, is not biased toward stability by the matching process. Because erroneous matches will inflate the match rate (a randomly selected voter from 1932 will share a party affiliation with a randomly selected voter from 1936 51.4% of the time), making quality matches, and correctly estimating the probability of a match, is essential to estimating the party-switching rate successfully.[1]

6.1 Preprocessing, Post-hoc Blocking, and Bayesian Model Specification

6.1.1 Construction of Comparison vectors

Before constructing comparison vectors we undertook a number of pre-processing steps. The suffix field is coded as missing so frequently that we were forced to discard it entirely. Prefix is also largely missing but is useful in that the vast majority of non-missing entries are either "mrs", "ms", or "miss" indicating that the individual is a woman, a feature which is not coded explicitly in the original data. We constructed an indicator variable for probable females if one of these prefixes appears, or if the occupation is recorded as "housewife" or a variant thereof. We then coded the occupation variable as missing for housewives; as chance agreement on occupation is very common Finally, we split the address field into three parts: street number, street name, and street type. Street number is coded as missing in cases where the street number is not included in the address. The street type was recoded (e.g. mapping both "rd" and "road" to "road") to standardize common abbreviations. The street name contains the remains of the original address field after removing the street number and street type from the original address string.

The files comprised 259,635 records from 1932 and 288,252 from 1936, yielding almost 75 billion record pairs. Before generating comparison vectors, we reduced the set of record pairs under consideration by employing indexing by disjunctions of blocking keys. A record pair was included if the first three characters of the given name or the first three characters of the surname matched exactly. Because many women's given name begins with "mar", a pairing required that either the first four characters of the given name match or the first three characters of the surname match. The result was a total of 822,444,349 record pairs, for which comparison vectors were computed.

To generate comparison vectors we employed a Jaro-Winkler string similarity score with a scaling factor of $p = 0.1$ for the given name, surname, occupation, and street name fields. We compared the street number field with a Levenshtein distance, using zero-padding to make them the same length before comparison. The

[1] To simplify the presentation of results, the 14.7% of voters that were registered as neither Democrat nor Republican in either of the two elections are excluded.

Level	Similarity Range
1	[1]
2	[0.85, 1)
3	[0.6, 0.85)
4	[0.45, 0.6)
5	[0.25, 0.45)
6	[0.0, 0.25)

Level	Similarity Range
1	[1]
2	[0.75, 1)
3	[0.5, 0.75)
4	[0.25, 0.5)
5	[0.0, 0.25)

Table 2: String similarity to ordinal mapping. Jaro-Winkler string similarity (left) and zero-padded Levenshtein string similarity (right).

string similarities were then binned, with the specific bins listed in Table 2. We compared our constructed female indicator and the street type using strict matching, assigning a 2 for an exact match and a 1 otherwise.

6.1.2 Post-Hoc Blocking

After computing the comparison vectors we estimated maximal weights using a sequence of penalized likelihood estimators as described in Section 4. Estimating the weights took about two hours on a desktop machine. We then performed a sensitivity analysis on the post-hoc blocks to select an appropriate threshold. Figure 7 shows the changes in the number of record pairs contained in the post-hoc blocks as well as the number of distinct post-hoc blocks generated as a function of the cutoff w_0. To ensure that an MCMC algorithm would be able to mix sufficiently we selected the smallest penalty for which the largest post-hoc block contained no more than 250,000 record pairs. The result was $w_0 = 7.9$, indicated by the vertical line in Figure 7. This turned out to be a conservative setting, as a slightly higher threshold would have yielded a precipitous drop in the total number of record pairs.

In the right panel of Figure 7 we also show the number of post-hoc blocks created as the threshold varies. We obtain a total of 65,792 post-hoc blocks with $w_0 = 7.9$. Of these, 23,364 are of size 1×1 meaning they contain only a single record pair. A further 36,553 blocks contain more than one pair but are 3×3 or smaller. Many of these correspond to housholds. As the vast majority of post-hoc blocks are small the restricted MCMC algorithm mixes rapidly.

6.1.3 Prior Distributions and Restricted MCMC

For given name, surname, occupation, and street name the prior distribution over $m-$parameters was set to $m_j \sim \text{Dir}(10, 6, 2, 1, 1, 1)$, while for street number we used $m_j \sim \text{Dir}(10, 6, 2, 1, 1)$ and for female and occupation we used $m_j \sim \text{Dir}(5, 1)$. We considered two prior distributions for C: The Beta-bipartite prior (Beta-bipartite) with $\alpha = 1.0$ and $\beta = 1.0$ as in Section 5, which is uniform over the proportion of matched records, and Green and Mardia (2006)'s prior distribution:

$$p(C) \propto \exp\left(\theta \sum_{ab} C_{ab}\right).$$

The mode under this prior corresponds to the maximum penalized likelihood estimator from 4.1, and we refer to it as Link-Penalty in the plots below. We chose $\theta = 5$, which puts higher probability on a larger number of links than the Beta-bipartite prior.

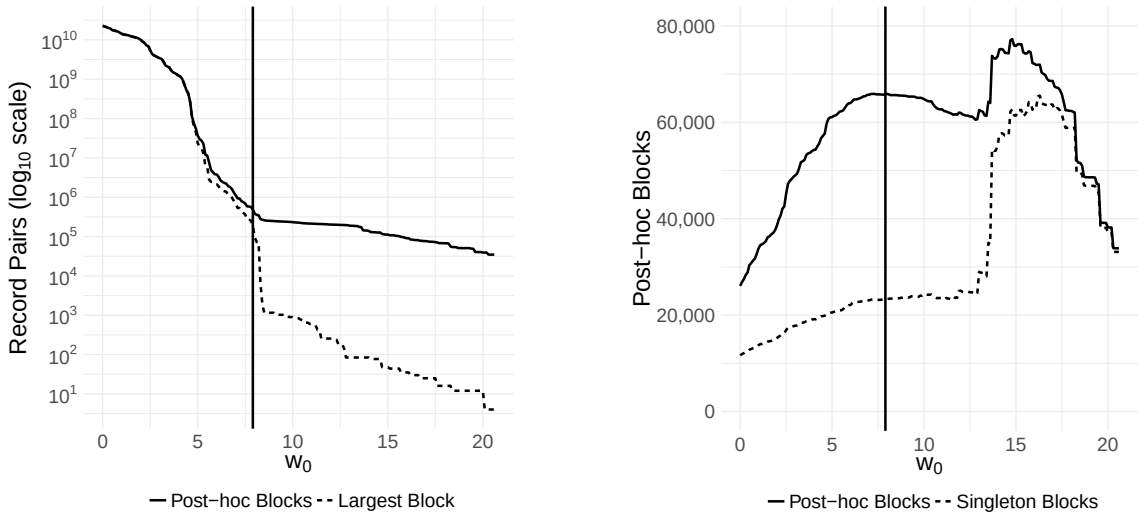


Figure 7: Record pairs in post-hoc blocks and in the largest post-hoc block (left). Largest dimension of largest post-hoc block (center). Number of post-hoc blocks and post-hoc blocks containing a single record pair (right).

We ran the MCMC algorithm for a total of 10,000 steps under each prior specification. Each "step" comprises an MCMC move within each of the post-hoc blocks. The runtimes were similar: 36.6 hours and 33.6 hours for the Beta-bipartite and Link-Penalty 5 priors respectively. Standard diagnostics indicate that the MCMC algorithm mixes well under both priors. After discarding the first 2,500 steps as burn-in we computed a posterior distribution over the number of links (Figure 8) and computed the number of record pairs for which link probabilities are above a given threshold (Figure 9). The Beta-bipartite prior is clearly more restrictive, resulting in both a lower posterior mean on the number of links and a smaller number of record pairs above any given posterior probability value.

The total number of links is quite sensitive to the prior. We examine changes in estimated posterior probabilities at the record pair level to get a better sense of what is driving the differences – for example, is there a small shift in all the record pair matching posterior probabilities, or a large shift for a small number of record pairs? The left panel of Figure 10 seems to indicate that it is a little of both, but there are a significant number of record pairs – over 2,000 – which have posterior probability near one under the more permissive Link-Penalty prior and posterior probability near zero in the restrictive Beta-bipartite prior. We examined a sample of these record pairs, and while he patterns are somewhat difficult to summarize we did observe that record pairs linked under Link-Penalty prior often agreed on all fields except for a modest disagreement on street name or street number.

6.2 Comparing Bayesian Models to fastLink

We compared the results of our Bayesian modeling to links obtained using the fastLink R package (Enamorado et al., 2017). We could not match all pre-processing and modeling choices for fastLink and our preferred choices for our Bayesian model. When it was necessary to modify our modeling or pre-processing decisions we deferred to fastLink defaults whenever possible. so we consider two different comparisons: Between our preferred Bayesian modeling approach and preprocessing steps, and a version of the Bayesian model that matches fastLink's data preprocessing and modeling approach (specifically the choice of blocking scheme and

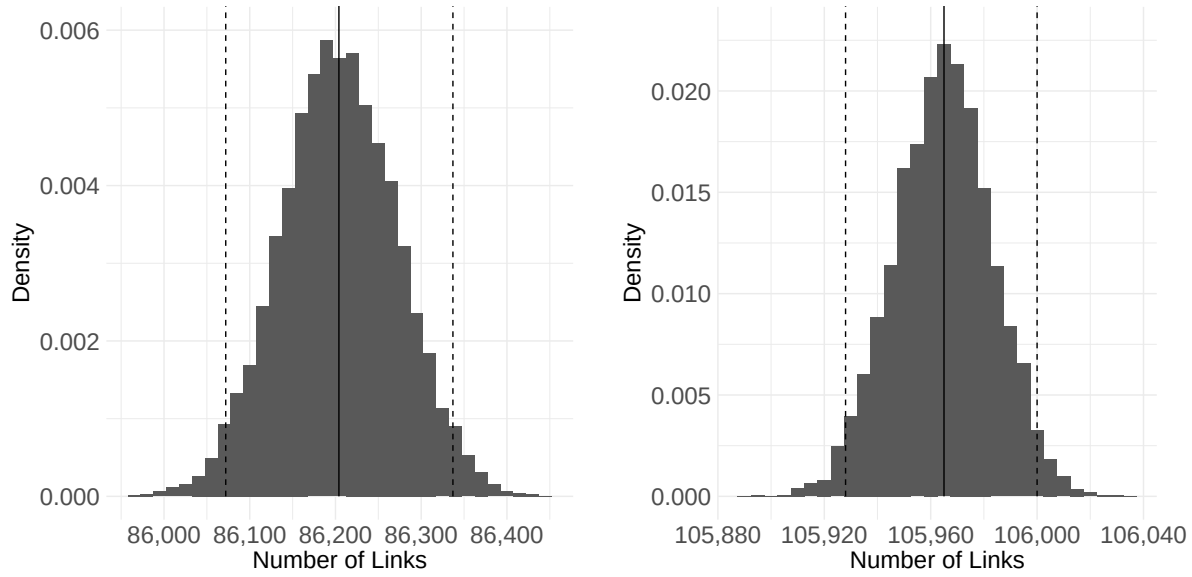


Figure 8: Posterior distribution of number of links under different priors: Beta-bipartite($\alpha = 1.0, \beta = 1.0$) (left) and Link-Penalty($\theta = 5.0$) (right)

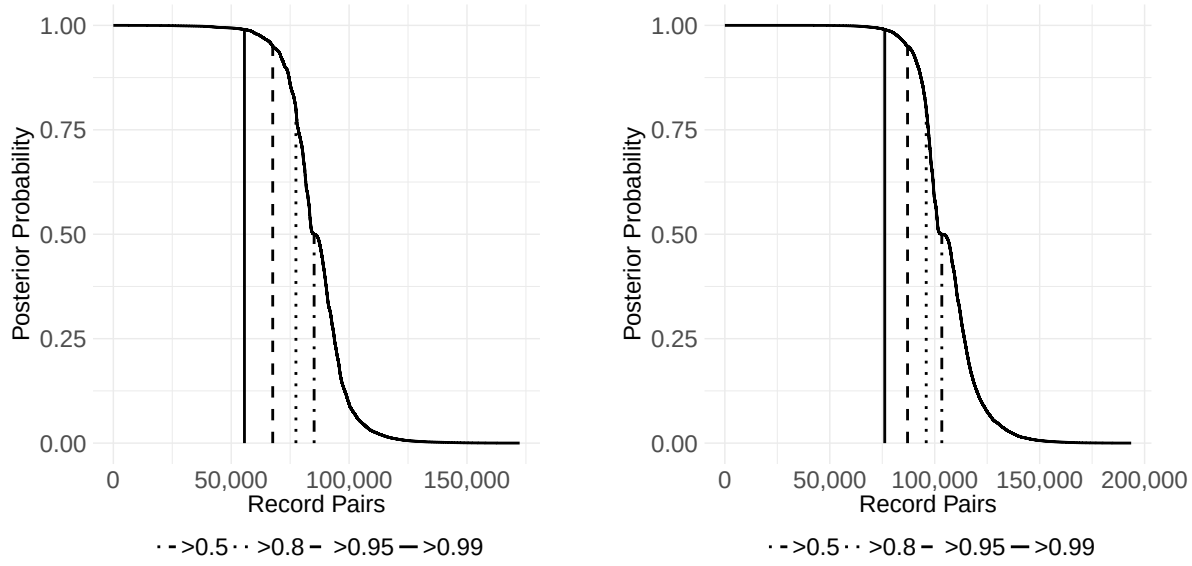


Figure 9: Number of record pairs classified as links varying posterior probability threshold under different priors: Beta-bipartite($\alpha = 1.0, \beta = 1.0$) (left) and Link-Penalty($\theta = 5.0$) (right)

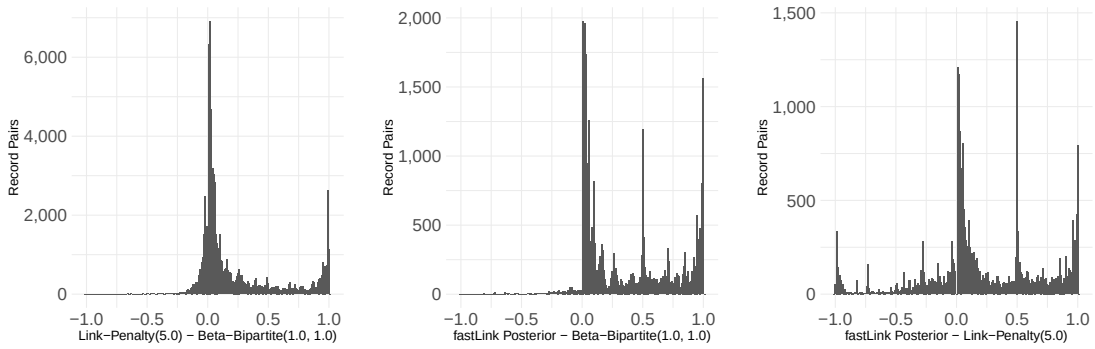


Figure 10: Difference in estimated pairwise posterior probabilities from prior specification (left) and comparison with fastLink posterior probabilities (center and right). We exclude the large proportion of points for which the estimated parameters are within 0.01 of each other. This excludes 124,276 pairs (62.7%) from the left panel, 34,160 pairs (54.4%) from the center panel, and 39,159 pairs (62.4%) from the right panel.

thresholds for similarity measures).

We were unable to implement indexing by disjunctions within the fasLink package, so we relied on internal functions to block on given name only. Using the built-in clustering function to generate blocks we elected to generate 100 separate blocks as this resulted in block sizes of no more than 10,000 by 10,000 (much larger than our post-hoc blocks). Since the blocking scheme is different, the Bayesian models and fastLink did not consider the same set of record pairs. However, there is some degree of overlap: Specifically, the fastLink blocking scheme considers a total of 1,038,427,954 record pairs, compared to the 822,444,349 record pairs included using our indexing scheme. A total of 558,813,418 pairs were included in both indexing schemes. Of the 85,165 record pairs for which the Beta-bipartite model estimates a posterior link probability of at least 0.5, 75,137 (88.2%) are included in the fastLink blocking scheme.

To generate comparisons with fastLink we used a Jaro-Winkler similarity metric with $p = 0.1$ on all fields except for female and street type for which we relied on exact matching. We were limited to three categories for the string comparisons by the fastLink package. To set thresholds we let string similarity above 0.92 correspond to an "exact" match, between 0.88 and 0.92 a partial match, and below 0.88 a non-match.

After running the fastLink matching algorithm we compared the estimated posterior match probabilities for all record pairs with estimated posterior probabilities provided by both of our Bayesian models. This comparison is somewhat imperfect as it is limited to the record pairs common to fastLink's blocking and our indexing scheme. However, it gives us some sense of how often the two approaches generate similar posterior probabilities. The center and right panels of Figure 10 show the differences between estimated posterior probabilities. The majority of probabilities are quite close, but there are some significant differences. Relative to both of our MCMC models fastLink tends to estimate larger posterior probabilities overall, particularly relative to the conservative Beta-biparatite prior. Examining record pairs in the bump around 0.5 in each of the figures suggests that these are cases were duplicates or near-duplicates were contained in the voter rolls. Examining a sample of record pairs for which fastLink predicts a near certain match and the MCMC posteriors assign close to 0 weight we find that these often correspond to record pairs which match on address and given name but disagree strongly on surname and occupation; these generally appear to be false matches.

One additional difference between the modeling approach taken in our MCMC algorithm and that of fastLink is that our algorithm shares parameters across blocks while fastLink estimates matches parameters separately within each block. We show the distribution of parameters estimated by fastLink across the blocks

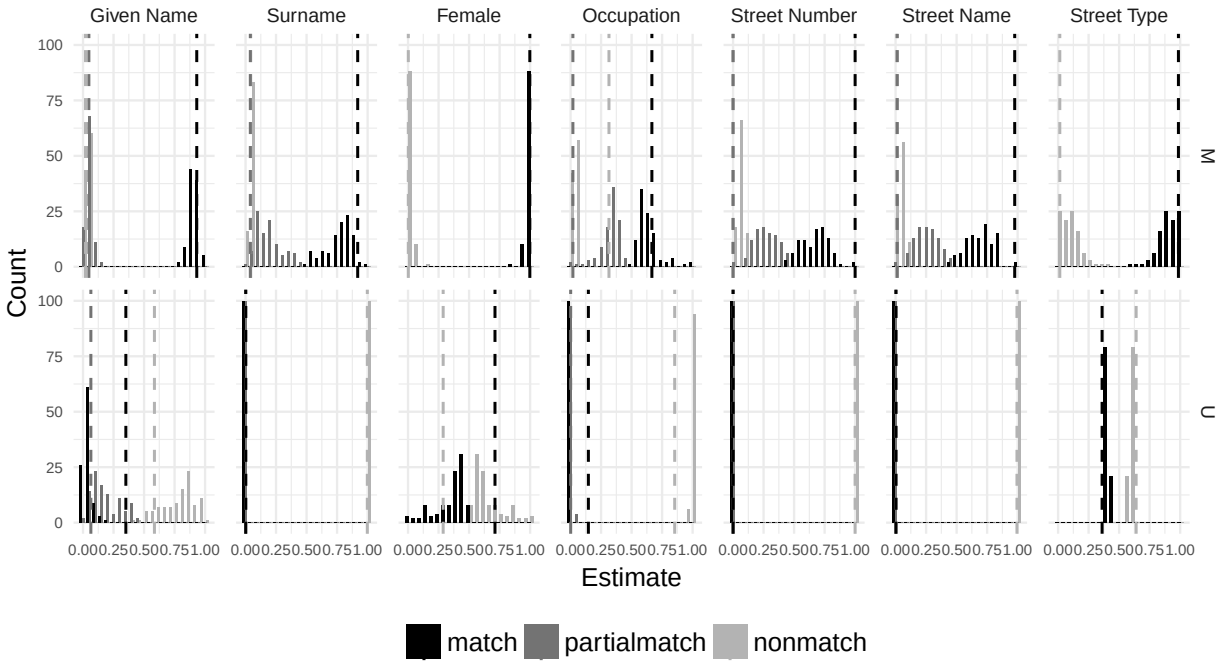


Figure 11: Distribution of parameters estimated across fastLink blocks. Dashed lines shown posterior mean of Bayesian model parameter estimates.

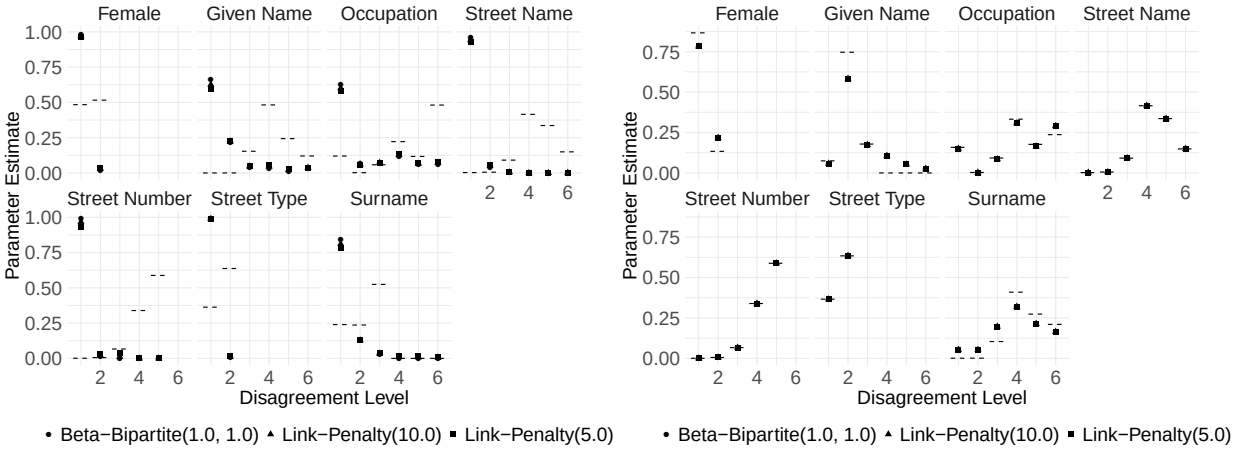


Figure 12: Posterior means of m parameters (left) and u parameters (right). Values estimated by our EM algorithm are shown as dashed horizontal lines.

in Figure 11. Some of the differences in posterior probabilities may be attributed to fastLink's blockwise approach, as there is significant heterogeneity in estimated parameters across blocks. On the whole parameter estimates seem reasonable for most of the blocks. The variability in $u-$ probabilities for given name and gender is due to the blocking scheme, as described in Murray (2016). The $m-$ parameters are also highly variable across blocks. This is a little harder to explain in some cases – it is unclear why the probability of an exact match on street name would depend so strongly on the given name block of the records in question.

As a final comparison between our MCMC approach and an EM based approach we estimate the m and u parameters using our own EM algorithm which allowed us to implement shared parameters and an identical indexing scheme. We compared posterior means of the estimated parameters across the three priors with the parameters estimated by the EM algorithm in Figure 12. The EM algorithm performs quite poorly; we observe cases of parameter inversions, where a high probability is estimated for high levels of disagreement conditional on a record pair being a true match matching. In contrast, the posterior means of the parameters under the Bayesian models are consistent with our expectations and nearly identical across prior specifications. This indicates that divergent results for posterior distributions over the number of matches are primarily a function of the prior.

6.3 Bayesian Model with fastLink inputs

As a further comparison between the Bayesian modeling approach and the results obtained from fastLink we fit the Bayesian model on the set of record pairs and comparisons vectors used by fastLink. As described in the previous section we block on given name using internal fastLink functions and compute comparision vectors using the "exact" match, partial match, and non-match bins defined by default values from fastLink. We do however continue to share the model parameter estimates across blocks in the Bayesian model. We employ the Beta-bipartite prior with $alpha = 1.0$ and $\beta = 1.0$ over the link structure. The prior over the m-parameters was set to $m_j \sim$Dir(10,5,1) for given name, surname, occupation, street number, and street name, the fields for which partial matches are computed. For female and street type, where only exact matches and non-matches are computed the prior was to $m_j \sim$Dir(10,1).

Estimating the same matching model allows a direct comparison between the parameters estimated by the Bayesian model and those estimated by fastLink as shown in Figure 11. In general we find that the Bayesian model produces $m-$parameter estimates that tend to be closer to 0 (non-matches and partial matches) and 1 (exact matches) than fastLink estimates. For street number and street name the partial match and non-match parameter estimates are both so close to zero that only one of the estimates is visible on the graph. We also notice large discrepancies in the estimated $u-$parameters for the female and given name fields. This appears to be driven by the constraint imposed in fastLink that $u_{exact-match} \leq u_{partial-match} \leq u_{non-match}$. In the case of these two parameters this assumption is at odds with the empirical distribution of computed comparisons, which we generally expect to closely match the estimated $u-$parameters. In the case of given name, because it is used for blocking, we observe more comparison vectors with an exact match on given name than comparison vectors with a partial match. Similarly, for female most blocks contain more comparison vectors which match on female than comparison vectors which record a non-match. In contrast the Bayesian model estimates for the $u-$parameters closely match the empirical proportions.

We next examine the differences in estimated pairwise posteriors match probabilites for the two models. We first exclude all record pairs for which neither model estimates a posterior probability above 0.05 to focus on differences in plausible matches. As shown in Figure 13 the estimated posterior probabilites for the vast majority of record pairs are fairly similar but there exists a substantial minority of record pairs for

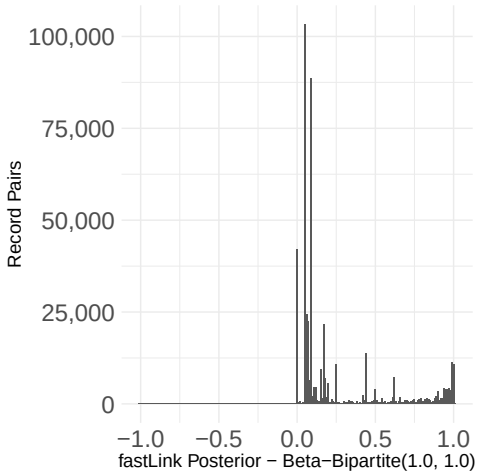


Figure 13: Difference in estimated pairwise posterior probabilities between fastLink and Bayesian model.

which fastLink estimates a posterior match probability near 1 while the Bayesian model estimates a posterior match probability near 0. There record pairs appear to fall into two primary categories address changes or "moves" and record pairs involving records with multiple strong match candidates.

We take a restrictive definition of movers, defined as record pairs for which we observe a match on given name, surname, female, and occupation, and a non-match on street name and street number (street type is ignored as it frequently matches by chance). We observe a total 14,135 distinct record pairs meeting this criteria for which fastLink estimates a posterior match probability of greater than 90%. However, many of the records included in this set If we further restrict ourselves to record pairs where the mover record pair has the highest estimated posterior probability across all record pairs containing either of the two records we retain 10,280 (72.7%) of the mover candidates. A hand examination of a sample of these record pairs suggests that they are highly likely to correspond to true matches. In contrast the MCMC algorithm assigns a posterior match probability of less than 1% to all record pairs which potentially correspond to movers.

6.4 Estimating Party Switching Rates

When using a dataset comprising roughly 86,000 to 106,000 linked record pairs, uncertainty due to sampling is small. Even in politically important subgroups like working women or white-collar men, there is enough data to measure party switching rates with reasonable precision if one assumes a fixed set of links. However, because we know that there is uncertainty in the link structure, it's necessary to account for that uncertainty in inferences. Probabilistic record linkage allows for two kinds of uncertainty to be captured: uncertainty in linkages conditional on a model, and uncertainty in linkages over a range of models. To illustrate these differences, we compare four models: the beta-bipartite model (hereafter "strict"), the link-penalty model with $\theta = 5.0$ (hereafter "loose"), the fastLink implementation of the EM model and the same run of fastLink with links with posterior probability $< .9$ excluded.

Politial scientists have postulated that the conversion of Republicans into Democrats was led by working class voters and women, arguing that more members of these groups switched parties than other segments of the electorate (Corder and Wolbrecht, 2016; Sundquist, 1983). That such a change occurred is readily apparent in the Great Registers. Because occupation and gender are identified for each individual in the Great Registers, it's straightforward to estimate the cross-sectional partisan composition and the party-

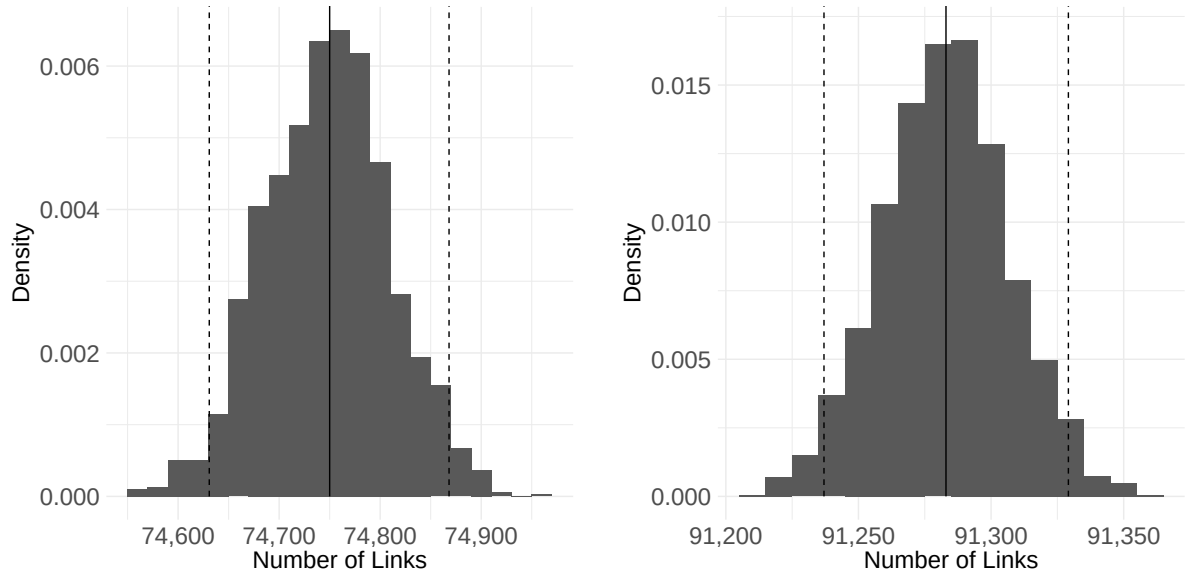


Figure 14: Posterior distribution of number of links where both record pairs record a known party under different priors: Beta-bipartite($\alpha = 1.0, \beta = 1.0$) (left) and Link-Penalty($\theta = 5.0$) (right)

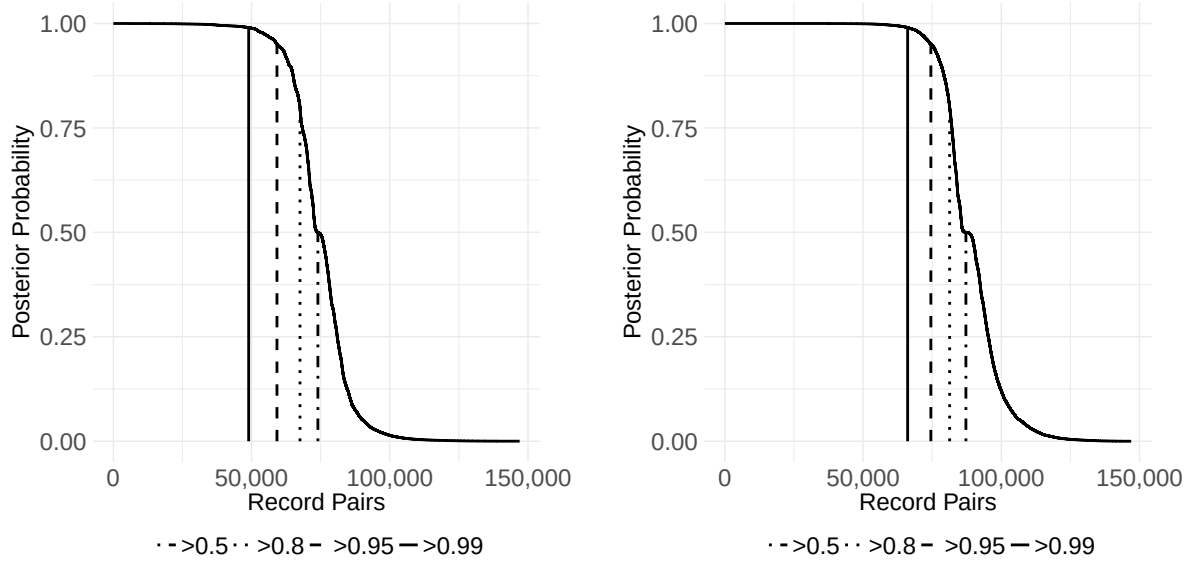


Figure 15: Number of record pairs classified as links varying posterior probability threshold where both record pairs reocrd a known party under different priors: Beta-bipartite($\alpha = 1.0, \beta = 1.0$) (left) and Link-Penalty($\theta = 5.0$) (right)

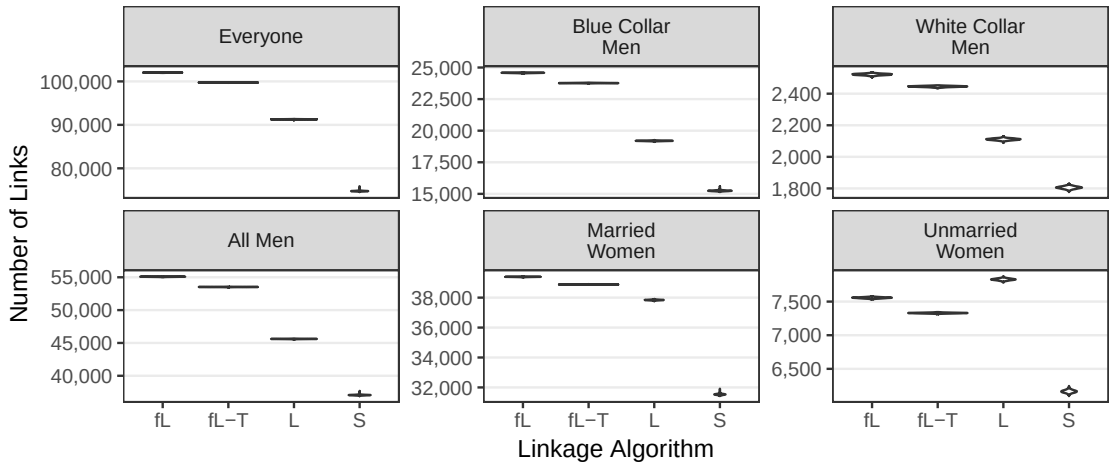


Figure 16: Violin plots of the distribution of the number of matches with observed party affiliation for interesting subgroups across samples of record-pairs. The four sets of links are for, respectively, fastLink (fL), thresholded fastLink (fL-T), the relatively loose Link-Penalty prior with $\theta = 5$ (L), and the strict Beta-Bipartite prior (S). Note that the totals in Fig 8 include matches without observed party affiliation and are therefore higher than the totals in the top-left panel here.

switching rates for individuals and aggregate up to groups to shed light on these theories. Before the realignment, all demographic groups had a Democratic registration rate of about 20%, though this rate rose significantly during the realignment period, indicating that most party switches were from the Republicans to the Democrats. Among men registered with one of the two major parties, blue collar men were 25 percentage points more likely to register as a Democrat in 1936 than in 1932. Among white collar men, the change was just 18 points. In Alameda county, women and men moved about the same amount, each increasing their support for the Demographics by a bit less than 25 percentage points. We focus here on the overall party-switching rate because it highlights differences in the linkage methods, but analyses that confront the politics of the period will be explored in another venue.

To characterize its uncertainty, the party-switching rate is computed for every fourth MCMC iteration of the Bayesian models to generate a posterior distribution of linkages. Because FastLink returns pairwise link probabilities rather than sampling sets of links, 1000 sets of links were sampled proportional to their fastLink estimated probability to provide some notion of uncertainty. In this procedure, links that are estimated to be certainly correct would be included in every set, while links with a posterior probabiliy of .5 would only be included in about half the iterations. For the thresholded version, only pairs with probability greater than $> .9$ were resampled, to avoid potential bias from poor links. This allows for the fastLink results to be analyzed analogously to the posterior draws from the MCMC algorithms. It is admittedly a crude approximation, but appropriate estimators and standard errors for the party switching estimand that account for linkage uncertainty do not appear to exist.

Figure 16 shows the distribution of the number of matches for each algorithm. FastLink returns the most matches, with an average of 102,000 pairs returned for each run. Restricting to matches with posterior probability $> .9$ reduces the total number of matches slightly, to about 100,000 on average. The link-penalty and beta-bipartite priors return an average of 91,000 and 75,000 records,respectively. The composition of the algorithms' sets of matches is broadly similar, though the bayesian algorithms are relatively more likely

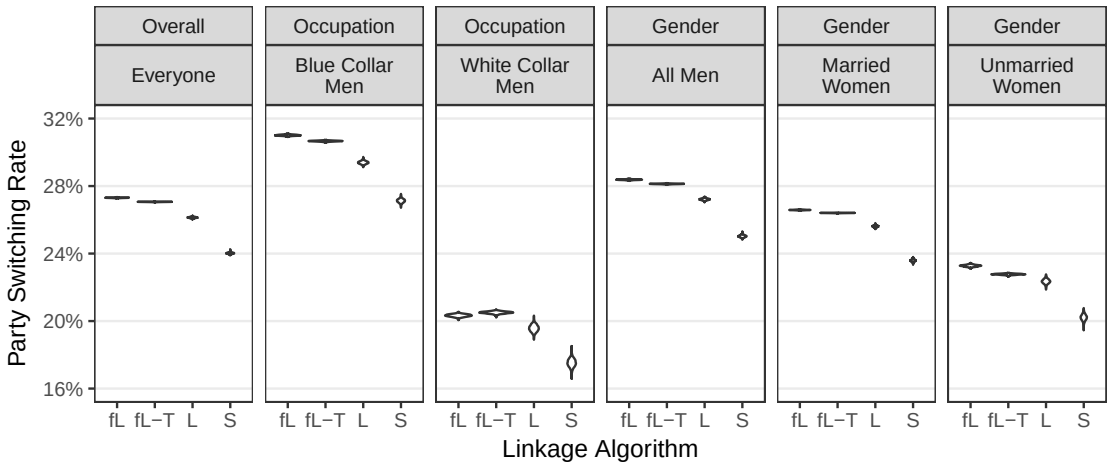


Figure 17: Violin plots of the distribution of the mean party switching rate for interesting subgroups across samples of record-pairs. The four sets of links are for, respectively, fastLink (fL), thresholded fastLink (fL-T), the relatively loose Link-Penalty prior with $\theta = 5$ (L), and the strict Beta-Bipartite prior (S).

to match women than fastlink, leading their returned set of matches to be about 4% more female. If the algorithms that return a higher number of matches are returning correct matches (or at least, correctly calibrated probabilities), then the number of data points available for analysis is increased without any bias. But if erroneous matches are returned (or matches with miscalibrated probabilities) we should expect a positive bias in the party switching rate.

The distribution of the mean party-switching rate overall and for interesting subgroups is displayed as a violin plot by linkage algorithm in figure 17. One notable pattern is that fastLink consistently shows higher rates of party-switching than the two Bayesian models. Similarly, the looser of the two Bayesian models, the one using the Link-Penalty prior, shows a higher switching rate than the strict model. The higher switching rate could be the result of substantive differences in the types of people being linked. For example, Figure 16 shows that the Link-Penalty prior actually matches more unmarried women than fastLink, despite linking fewer records overall. However, a more concerning possibility is that the high rate is attributable to a higher incidence of mismatches. That is, because most voters don't switch parties, but random pairings of different people will show a party switch about half the time, sets of links with more incorrect matches will show higher rates of party switching.

One way to identify which of the two sources is leading to the conflicting switching rates is to look at groups one would expect to have particularly high or low switching rates, thereby controlling for differences in composition between the sets of matches. It's widely accepted that blue collar workers swung much more to the Democrats during the realignment than did white collar workers, so one would expect higher switching rates for the former group. Indeed, figure 17 shows that white collar workers are much more likely to remain with their party than their blue collar counterparts. However, within both groups the pattern of relatively high rates of switching among fastLink matches and low rates of switching among the strictest Bayesian model's matches persists. This suggests that the higher rates of switching observed in the fastLink matches (and perhaps also in the Bayesian model with the link-penalty prior) may be spurious.

Because women were less likely to switch parties than men, it's possible that the overall higher rate of party-switching measured by the fastlink algorithms could be in part due to compositional differences in the

matches returned. However, even if the individuals returned by each algorithm were assigned the switching rate measured for their demographic group by the "S" algorithm, the overall estimate of the switching rate would only decline by less than .2 percentage points. This means that the differences in switching rates can not be explained by observable differences in the composition of the two sets of matches.

Given that the

One other area of interest is uncertainty propagation. In this setting, there are three important sources of uncertainty: uncertainty in the correct set of links conditional on the model, uncertainty due to sampling error (of the typical frequentist kind) and uncertainty in the linkage model. The first two sources of uncertainty are quite small here. Differences between models, however, are substantial. For this reason, choosing a reasonable model that minimizes incorrect linkages is essential.

Though the absolute estimates of switching rates differ between the three models, their relative variation across categories is consistent, allowing for an unpacking of important questions about which voters drove the realignment. The high switching rates among blue collar voters confirm what's long been known: that blue collar workers led the realignment towards Roosevelt's Democratic party. This fact is well-established because blue collar and white collar workers tend to be geographically separated, allowing ecological inference methods to tease apart the voting behavior of different kinds of workers.

Separating the political attitudes of men and women is considerably harder because they tend to be clustered together in space, voting in the same places. Though Corder & Wolbrecht (2016) use ecological inference methods to try to separate the political behavior of men and women, such approaches will always prove difficult because of low variation in the gender ratio. Individual-level data is much better suited to the task. Figure 17 shows that men switched parties at a considerably higher rate than women, contrary to what one would expect from Corder & Wolbrecht's analysis. Indeed, the realignment forged a partisan gender gap where none existed before, with married women lagging men by 5 points in Democratic party affiliation, with unmarried women lagging by a further 4. Though it's hard to say for sure why unmarried women (who are presumably younger) would have realigned less than their married women counterparts, one possibility is that they were more influenced by the parents who are older (and, on average, more Republican) than married women's spouses, who are closer to their same age. The clarity that panel data can bring to questions of individual behavior demonstrates the promise that new data and improved linkage methods bring to social science.

7 Discussion

Bayesian probabilistic record linkage models provide an appealing framework for performing record linkage: They can provide accurate point estimates of links between records, and they allow for uncertainty in the links between to be quantified and propagated through to subsequent inference. The main barrier to their adoption in practice has been computational. Post-hoc blocking and restricted MCMC make Bayesian modeling for PRL feasible for much larger problems, as demonstrated in our Great Registers case study. Our case study also showed that results can be quite sensitive to prior specifications; we are not the first to observe this phenomenon (see e.g.. Steorts et al. (2016)). A serious Bayesian analysis is obliged to consider sensitivity to the prior and model specification. Sensitivity analysis at this scale is only possible because of post-hoc blocking and restricted MCMC. While our development of post-hoc blocking focused on merging two files under one-to-one matching constraints, this general approach can be adapted for record linkage and de-duplication with more than two files. We expect this will be a fruitful line of research moving forward,

and bring Bayesian PRL to bear on a host of new and important scientific problems.

References

Alicandro, G., Frova, L., Sebastiani, G., Boffetta, P., and La Vecchia, C. (2017). Differences in education and premature mortality: a record linkage study of over 35 million italians. *European Journal of Public Health*.

Bertsekas, D. P. (1992). Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, 1(1):7–66.

Bertsekas, D. P. (1998). *Network optimization: continuous and discrete models*. Citeseer.

Bertsekas, D. P. and Eckstein, J. (1988). Dual coordinate step methods for linear network flow problems. *Mathematical Programming*, 42(1-3):203–243.

Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ.

Betancourt, B., Zanella, G., Miller, J. W., Wallach, H., Zaidi, A., and Steorts, R. C. (2016). Flexible models for microclustering with application to entity resolution. In *Advances in Neural Information Processing Systems*, pages 1417–1425.

Carpaneto, G. and Toth, P. (1983). Algorithm for the solution of the assignment problem for sparse matrices. *Computing*, 31(1):83–94.

Christen, P. (2012). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media.

Copas, J. and Hilton, F. (1990). Record linkage: statistical models for matching computer records. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, pages 287–320.

Corder, J. K. and Wolbrecht, C. (2016). *Counting Women's Ballots : Female Voters from Suffrage through the New Deal*. Cambridge University Press, New York, NY.

Dalzell, N. M. and Reiter, J. P. (2016). Regression modeling and file matching using possibly erroneous matching variables. *arXiv preprint arXiv:1608.06309*.

Dalzell, N. M., Reiter, J. P., and Boyd, G. (2017). File matching with faulty continuous matching variables. Technical report.

Dusetzina, S. B., Tyree, S., Meyer, A.-M., Meyer, A., Green, L., and Carpenter, W. R. (2014). *Linking data for health services research: a framework and instructional guide*. Agency for Healthcare Research and Quality (US), Rockville (MD).

Enamorado, T., Fifield, B., and Imai, K. (2017). Using a probabilistic model to assist merging of large-scale administrative records.

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

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

Fortini, M., Nuccitelli, A., Liseo, B., and Scanu, M. (2002). Modelling issues in record linkage: a bayesian perspective. In *Proceedings of the American Statistical Association, Survey Research Methods Section*, pages 1008–1013.

Gazit, H. (1986). An optimal randomized parallel algorithm for finding connected components in a graph. In *Foundations of Computer Science, 1986., 27th Annual Symposium on*, pages 492–501. IEEE.

Green, P. J. and Mardia, K. V. (2006). Bayesian alignment using hierarchical models, with applications in protein bioinformatics. *Biometrika*, 93(2):235–254.

Gutman, R., Afendulis, C. C., and Zaslavsky, A. M. (2013). A bayesian procedure for file linking to analyze end-of-life medical costs. *Journal of the American Statistical Association*, 108(501):34–47.

Hof, M. H., Ravelli, A. C., and Zwinderman, A. H. (2017). A probabilistic record linkage model for survival data. *Journal of the American Statistical Association*, 112(520):1504–1515.

Hong, C., Zhang, J., Chungfeng, C., and Qinyu, C. (2016). Solving large-scale assignment problems by kuhn-munkres algorithm. In *2nd Int. Conf. Adv. Mech. Eng. Ind. Informatics (AMEII 2016), no. Ameii*, pages 822–827.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, 84(406):414–420.

Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340.

Jonker, R. and Volgenant, T. (1986). Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97.

Larsen, M. D. (2005). Advances in record linkage theory: Hierarchical bayesian record linkage theory. In *Proceedings of the American Statistical Association, Survey Research Methods Section*, pages 3277–3284.

Larsen, M. D. (2010). Record linkage modeling in federal statistical databases. In *FCSM Research Conference, Washington, DC. Federal Committee on Statistical Methodology*. Citeseer.

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

Lawler, E. L. (1976). *Combinatorial optimization: networks and matroids*. Courier Corporation.

Liseo, B. and Tancredi, A. (2011). Bayesian estimation of population size via linkage of multivariate normal data sets. *Journal of Official Statistics*, 27(3):491–505.

Mackay, D. F., Wood, R., King, A., Clark, D. N., Cooper, S.-A., Smith, G. C., and Pell, J. P. (2015). Educational outcomes following breech delivery: a record-linkage study of 456 947 children. *International journal of epidemiology*, 44(1):209–217.

Murray, J. S. (2016). Probabilistic record linkage and deduplication after indexing, blocking, and filtering. *arXiv preprint arXiv:1603.07816*.

Newcombe, H. B., Kennedy, J. M., Axford, S., and James, A. P. (1959). Automatic linkage of vital records. *Science*, 130(3381):954–959.

Orlin, J. B. and Lee, Y. (1993). Quickmatch–a very fast algorithm for the assignment problem.

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

Sadinle, M. et al. (2014). Detecting duplicates in a homicide registry using a bayesian partitioning approach. *The Annals of Applied Statistics*, 8(4):2404–2434.

Sauleau, E. A., Paumier, J.-P., and Buemi, A. (2005). Medical record linkage in health information systems by approximate string matching and clustering. *BMC Medical Informatics and Decision Making*, 5(1):32.

Steorts, R. C. et al. (2015). Entity resolution with empirically motivated priors. *Bayesian Analysis*, 10(4):849–875.

Steorts, R. C., Hall, R., and Fienberg, S. E. (2016). A bayesian approach to graphical record linkage and deduplication. *Journal of the American Statistical Association*, 111(516):1660–1672.

Sundquist, J. L. (1983). *Dynamics of the party system*. Brookings Institute Washington, DC.

Tancredi, A., Auger-Méthé, M., Marcoux, M., and Liseo, B. (2013). Accounting for matching uncertainty in two stage capture–recapture experiments using photographic measurements of natural marks. *Environmental and ecological statistics*, 20(4):647–665.

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

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160.

Thibaudeau, Y. (1993). The discrimination power of dependency structures in record linkage. *Survey Methodology*, 19:31–38.

Winkler, W., Yancey, W., and Porter, E. (2010). Fast record linkage of very large files in support of decennial and administrative records projects. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pages 2120–2130.

Winkler, W. E. (1988). Using the em algorithm for weight computation in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, volume 667, page 671.

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage.

Winkler, W. E. (1993). Improved decision rules in the fellegi-sunter model of record linkage.

Winkler, W. E. and Thibaudeau, Y. (1991). An application of the fellegi-sunter model of record linkage to the 1990 us decennial census. *US Bureau of the Census*, pages 1–22.

Yancey, W. E. (2002). Bigmatch: A program for extracting probable matches from a large file for record linkage. Technical report statistical research report series rrc2002/01, U.S. Bureau of the Census, Washington, D.C.

Zanella, G. (2017). Informed proposals for local mcmc in discrete spaces. *arXiv preprint arXiv:1711.07424*.