Carnegie Mellon University
Dietrich College of Humanities and Social Sciences
Dissertation

Submitted in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy

Title: Methods for the Estimation of Large Scale Bayesian Models for Record Linkage Under One-to-One Matching

Presented by: Brendan S. McVeigh

Accepted by: Department of Statistics & Data Science

Readers:

_____
Jared S. Murray, Advisor

_____
Rebecca Nugent, Co-advisor

_____
Brian W. Junker

_____
Chad M. Schafer

Approved by the Committee on Graduate Degrees:

_____
Richard Scheines, Dean Date

Carnegie Mellon University

Methods for the Estimation of Large Scale Bayesian Models for Record Linkage Under One-to-One Matching

A Dissertation Submitted to the Graduate School

in Partial Fulfillment of the Requirements for the degree

Doctor of Philosophy

in

Statistics

by

Brendan S. McVeigh

Department of Statistics & Data Science
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University

May 2020

To Trish and Mary, who would have accepted nothing less

Acknowledgements

It would be impossible to properly thank everyone who contributed to my research and dissertation in some way. I, therefore, request forgiveness for any omissions. First, I'd like to thank my advisor, Jared Murray, who oversaw nearly every piece of research in this document and has continually provided insightful and motivating feedback. Remarkably, he managed this feat, via webcam, even after moving over 1,000 miles away. Rebecca Nugent also provided invaluable input from much closer by and continually encouraged me to start writing sooner – advice I often ignored (to my detriment). The rest of my committee, Brian Junker and Chad Schafer, also provided valuable insights that significantly improved the quality of the final document.

I would not have made it through my coursework, let alone writing a dissertation, without the support of my cohort in the Department of Statistics & Data Science. The entering class of 2014 was with me every step of the way, and I would certainly not have completed this work without them – thank you all! In particular I'd like to thank Shannon Gallagher, Amanda Luby, Maria Cuellar, and Jacqueline Mauro, who all had the misfortune of sharing an office with me. I am also indebted to many other department members for delightful, and insightful, office conversations, both on and off topic. These include: Michael Vespe, Benjamin Leroy, Purvasha Chakravarti, Mikaela Meyer, Jerzy Wieczorek, Paige Houser, and Zach Branson. I'm also grateful to my other CMU and Pittsburgh friends and housemates, who helped me aspire to a semblance of work-life balance – thank you to David Adler, Caroline Hopkins, Ryan Carlson, Haixin Dang, Vivian Feldblyum, Robert Steel. I would also like to thank Sarah Frisco for throwing an amazing dinner party at a moment when I really needed a good night with friends.

Finally, I would never have started on this path were it not for my parents, Trish and Chris. Had they not raised me with the appropriate mix of encouragement and brainwashing, I might have chosen a different path – and my life would be much poorer for it. A number of extended family members generously aided them in this effort, with a special thank you to Susan Huse for her comments on this document. Additional motivation was also provided by my younger siblings, Kieran and Quinn, who promised to never let me hear the end of it if one of them completed a PhD before me. My final thank you goes to my partner Manya Sleeper, who not only remained supportive throughout the completion of this document but did so while isolating with me in the midst of a global pandemic.

Abstract

Probabilistic record linkage (PRL) is the process of identifying pairs of records from two files or datasets that correspond to the same underlying entity. In the absence of an error-free unique identifier, links between records must be estimated and are inherently uncertain. Properly characterizing this uncertainty is a prerequisite for correctly performing any statistical estimation or inferential task with the resulting linked data. In nearly all real-world record linkage problems a lack of training data, which must be specific to the records contained in both datasets, presents an additional challenge and necessitates the use of unsupervised methods. Bayesian methods provide a powerful mechanism for quantifying uncertainty in the estimated link structure via the posterior distribution. Furthermore, they allow for the inclusion of additional information, such as structural constraints on the link structure, via an appropriate prior distribution. Incorporating such information into the estimation can significantly improve performance, particularly for unsupervised methods. The application of Bayesian methods to record linkage problems has thus far been limited by a lack of methods that can practically be applied to sets of records containing more than a few thousand entries.

In this dissertation we make several methodological advances that allow Bayesian methods to be applied successfully to sets of records that are orders of magnitude larger than is possible with existing methods. We first reexamine the standard two-step method for resolving a set of similarity weights between records into a link structure consistent with one-to-one matching. We develop a joint procedure that uses a modified optimization problem to induce sparsity, significantly reducing the computational complexity. This allows for the efficient calculation of a maximum a posteriori (MAP) estimate of the link structure, under an appropriate prior distribution over the link structure. We next address the more general question of estimating Bayesian models for record linkage via a MCMC sampler. Due to the discrete nature of the link structure, standard MCMC samplers converge extremely slowly and are impractical for almost all real problems. We develop a data-driven blocking scheme that separates the link structure in a large number of small disjoint regions within which a MCMC sampler can mix efficiently. Finally, we provide a method for transforming existing prior distributions over the link structure that are informative only on the number of matches, to distributions

that are also informative over the expected types of matches. Importantly the transformation maintains both one-to-one matching constraints and invariance to the ordering of the records.

We demonstrate the scalability of our contributions by linking two historical voter registrations that contain hundreds of thousands of records. Both files where generated by applying OCR to the original paper voter registries and, therefore, contain numerous missing or incorrect entries. Despite lacking a high quality blocking key, our approach allows a posterior distribution to be estimated on a single machine in a matter of hours. We further demonstrate, using a set of hand-labeled record pairs, that a fully Bayesian estimate significantly outperforms other methods, both linking more record pairs and achieving a lower false match rate.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Probabilistic record linkage (PRL) is the process of identifying sets of records from different files or databases that correspond to a unique underlying entity. In the absence of a uniquely identifying key, such as a social security number, links or matches between pairs of records must be estimated. In many cases the available fields, such as names, may not uniquely identify the underlying entity. Moreover, in many cases fields will contain errors or may be missing entirely. The task of identifying records which correspond to the same underlying entity is therefore a non-trivial one as the matches must be estimated using fields in common between the two records. Any estimated matches are thus inherently uncertain. This uncertainty in the estimated link structure must be incorporated into any analysis which relies on linked data.

With the explosion in digital data the demand for reliable methods for linking data, and for analyzing linked data, has grown accordingly. PRL methods are applied in a wide variety of scenarios from administrative data (Jaro, 1989; Winkler, 1988; Winkler and Thibaudeau, 1991; Winkler, 1993; Larsen and Rubin, 2001) to healthcare (Gutman et al., 2013; Dusetzina et al., 2014; Alicandro et al., 2017) and education (Mackay et al., 2015). Further recent applications include the estimation of casualty counts in conflicts (Sadinle, 2013, 2017; Steorts et al., 2015, 2016; Chen et al., 2018) and energy projects (Dalzell et al., 2017a). Traditionally, the results of many PRL models were reviewed for accuracy (Jaro, 1989; Winkler, 1991) with the most ambiguous cases could be sent for clerical review (Larsen and Rubin, 2001). However, with PRL problems commonly containing millions of records (Xin et al., 2018) or more, there is a need for unsupervised PRL models which nonetheless achieve a high level of accuracy.

In this dissertation we make several contributions to the field unsupervised PRL methods for merging two files each of which contains no duplicate records. For the remainder of this chapter we provide an introduction to existing methods for PRL. In Chapter 2 we reexamine the standard approach to estimating a link structure consistent with one-to-one matching. We find that advances from the optimization literature, specifically auction algorithms (Bertsekas, 1998), can be applied to PRL in a manner that both reduces their computational complexity and achieves better performance. Then, in Section 2.3, we use these advances to develop a new *penalized-likeihood* estimator that operates similarly to existing methods but maintains a one-to-one assignment constraint throughout the estimation processes. In Chapter 3 we introduce *post-hoc blocking*, a method which allows Bayesian models for record linkage to be applied to PRL problems several orders of magnitude larger than what can be handled by existing methods. We demonstrate the effectiveness of post-hoc blocking by matching registered voters in Alameda county, CA in 1932 and 1936. Using post-hoc blocking we successfully apply Bayesian PRL to these datasets, which contain hundreds of thousands of records. Finally, in Chapter 4, we introduce a new iterative method for constructing an informative prior for Bayesian PRL. The resulting prior allows expectations about what types of record pairs will be matched, in addition to the expected number of such pairs, to be incorporated into the estimation processes benefiting model performance.

1.2 Overview of the Record Linkage Process

We consider two sets of records, A and B, with each record containing identifying information, although the information may not be uniquely identifying. The goal of record linkage is to use this information to identify pairs of records (a, b), with $a \in A$ and $b \in B$, such that both records correspond to the same underlying entity. Let n_A by the number of records contained in A and n_B by the number of records contained in B. We further assume, without loss of generality, that $n_A \leq n_B$. Then let C be an $n_A \times n_B$ binary matrix, where:

$$C_{ab} = \begin{cases} 1 & \text{if record } a \text{ matches record } b \\ 0 & \text{otherwise} \end{cases} . \tag{1.1}$$

The value of C thus describes the linkage structure between records in A and B. An alternative interpretation, which we will make use of throughout, is to view C as the adjacency matrix for bipartite graph where each record corresponds to a node and links between them correspond to edges. We further assume that each entity corresponds to at most one record in A and one record in B. Since entities appear at most once in each dataset this implies that each record in A can be matched to at most one record in B and each record in B to at most one record in A. This is a common assumption in the literature (Jaro, 1989; Fortini et al., 2001, 2002; Larsen, 2010; Tancredi et al., 2011; Sadinle, 2017), which we refer to as *one-to-one matching'*. This contrasts with deduplication, the other canonical PRL problem, which seeks

to identify groups of entries within a single dataset that correspond to the same underlying entity. While there is significant overlap in approaches taken to these two problems we restrict our focus to record linkage under one-to-one matching in this dissertation. However, many of the insights we present seem likely to be application to deduplication problems as well.

Solving a PRL problem typically involves at least three stages: (1) the identification of a subset of the record pairs which are considered for linking, (2) the generation of comparisons between records for each record pair contained in this subset, and (3) estimation of the link link structure. We provide an overview of each step in turn in Sections 1.3-1.6. Finally, many applications contain a fourth step of analysis with linked data. That is, the goal of PRL is generally not the link structure itself but some quantity for which an estimate of the link structure is required. We provide a brief discussion of methods for uncertainty propagation in Section 1.7 and an example of an analysis with linked data in Section 3.3.1 of Chapter 3.

1.3 Blocking Methods

For PRL problems in which both A and B contain relatively few records a straight forward approach to estimating the link structure might begin by comparing each of the n_A records in A with all n_B records contained in B resulting in comparisons for $n_A \times n_B$ record pairs. However, due to the quadratic growth rate for the number of record pairs, this quickly becomes computationally intractable for even moderately sized sets of records. To make the generation of comparison vectors computationally tractable the set of record pairs considered for matching must first be reduced to a small enough set that the necessary comparisons can be made. Thus, in practice only a subset of the possible record pairs are considered for matching in all but the smallest PRL problems (Larsen, 2002, 2010; Sadinle, 2013). For simplicity we will refer to any process which reduces the set of record pairs considered for linking *before* modeling as either a *blocking scheme'* or an *indexing scheme*.

In an ideal setting a blocking scheme would included all record pairs corresponding to true matches, those that would be matched by an oracle, and few if any additional record pairs. In practice there is generally a trade-off based on the number of record pairs included within the blocking scheme. Selecting a blocking scheme which includes fewer total record pairs will also tend to exclude some true matches, but will result in a much easier problem from a computational standpoint. Since blocking is generally a first step before estimating a full model we should generally be willing to accept a scheme with returns a large number of non-matching record pairs, as long as it remains computationally tractable, to reduce the number of true matching records that are excluded. Thus, blocking criteria can be viewed as necessary but not sufficient condition for classifying a record pair as a true match. Alternatively, blocking can be thought to consider only those records with some minimum level of similarity as possible links. The result of the blocking scheme should be a large reduction in the set of record pairs considered for linking, equivalent to setting

the link probability of all comparisons outside of the blocks to zero. There are a wide range of strategies for conducting a blocking scheme and we offer only a brief overview here. For a more complete overview see Herzog et al. (2007); Christen (2012b,a); Steorts et al. (2014) and citations therein.

1.3.1 Traditional Blocking

Traditional blocking relies on constructing a mapping from records to a *blocking key*, on which matching records are assumed to match exactly. A blocking key thus partitions the two sets of records into disjoint groups or *blocks*. Each record is then only considered for matching against other records within the same block. Common examples of fields used as blocking keys include geographic identifies such as county or zipcode, and fields that are unlikely to change over time such as first name. Blocking is straight forward to implement and can dramatically reduce the number of record pairs that need to be considered for matching. An ideal blocking key will contain no errors and partition the sets of records into a large number of relatively small blocks. An additional benefit is that such a partition can be constructed without actually comparing record pairs on the blocking key, this allows blocks to be computed in linear time with respect to the number of records.

Consider a simple example: we are attempting to link two relatively small datasets which each contain 10,000 records (10^4). The full sets of possible record pairs therefore corresponds to 100,000,000 (10^8) total record pairs. If however, the data contains a field, such as a zipcode, which takes 20 distinct values and each value occurs with the same frequency (500 occurrences in each dataset). Then each record will only need to be compared with a total of 500 other records, not 10,000. This reduces the number of comparisons to "only" 5 million, a 95% reduction in the total number of comparisons which must be made.

While blocking can dramatically reduce the number of comparisons which must be made it comes with some distinct challenges. First, because it requires that record pairs match exactly on a blocking key any errors in the field (or fields) used in the blocking will be propagated through, potentially resulting in false non-matches. For this reason it is important to select a blocking key that contains as few errors as possible, something which may not always be available. Second, as the size of the datasets grow, a more discriminative blocking keys is generally required, so that a larger reduction in the number of record pairs can be made. This tends to exacerbate the first issue, since a more discriminating blocking key may be more error prone. To address this problem a variety of relaxations to traditional blocking have been developed (Kelley, 1985).

1.3.2 Relaxations of Traditional Blocking

Traditional blocking requires that record pairs match *exactly* on the blocking key, a requirement that may often be violated in practice. For continuous variables, such as ages or times, we might consider relaxing this requirement by only requiring that records match within a certain tolerance. For a birth year this might

be 5 years. As with traditional blocking an advantage to this approach is that the resulting set of record pairs can be generated without actually making $n_A \times n_B$ comparisons by employing a sorted neighborhood approach (Christen, 2012a). Roughly, this involves sorting both datasets on the blocking key and scanning through them simultaneously to identify the regions for which is difference in the value of the blocking key is within the tolerance. Unfortunately, this approach does not generalize well for many comparison methods since it requires an ordering on the field. For example, this means that it cannot be applied to similarity measures for string variables. Thus, while there is no theoretical reason why a similarity threshold for a string similarity metric cannot be used to construct a blocking scheme in practice the computational costs are generally prohibitive.

A second relaxation of traditional blocking is the use of multiple blocking keys. If we require that record pairs match exactly on multiple blocking keys then we have in essence performed traditional blocking and just combined the keys. If, however, we seek to include record pairs in our blocking scheme as long as they match on one *or more* of several blocking keys then we have expanded the set of record pairs considered for linking. Taking the union of the record pairs which would be considered by blocking on any of the individual keys. Refer to this approach as *indexing by disjuntions* or simply *indexing*, the terminology used in Murray (2016). An advantage to indexing is that it is robust to errors in any individual blocking key. All truly matching record pairs will be included in the indexing scheme as long as both records are error-free on at least one blocking key. We might, for example, consider record pairs which match on either first name or last name. The results would include all true matches except for those for which the records fail to match on both first name and last name.

A downside, relative to traditional blocking, of any approach relying on either approximate matching or on the union of multiple blocking keys is that it will not generally produce distinct blocks. That is, there will be overlap between the groups of record pairs. To use the example of applying indexing by disjuntions to first and last name may will be a group of record pairs that are included because they all contain the first name "james" and a second groups corresponding to the first name "william". There may be a third group of records which were included because they agree the last name, "smith". Finally it may also be the case that some records, those with the name "james smith" and "william smith", will be present in both the first group and the second group or the first and third group. Because of this overlap the link structure within the blocks must be estimated jointly, that is when estimating links for records with the first name james we must also consider the records with the first name william. This typically result, in a more computationally challenging estimation problem than estimating the link structure separately within each of a set of disjoint blocks. Given the computational advantages of having disjoint blocks we may wish to restore this property to blocking schemes which do not exhibit it.

1.3.3 Cluster based methods

A natural approach to creating disjoint blocks, such as those generated generated by a traditional blocking scheme, is to apply clustering methods. Clustering has an intuitive appeal as we might expect truly matching records to appear similar across a variety a metrics and therefore be relatively easy to place in the same cluster. However, standard clustering methods, such as hierarchical clustering, rely on computing a similarity measure (or distance) between observations to determine the clustering (Rokach and Maimon, 2005). While a similarity measure for a single field can be employed (Enamorado et al., 2019) in PRL problems it can often be challenging to determine a priori an appropriate field similarity. There are two common methods for making the problem tractable: (1) to cluster only a reduced set of record pairs, such a set of record pairs produced by an indexing scheme, for which more complicated distance measures can be computed or (2) construct a function that directly maps records to clusters can be constructed. Such a function removes the need for comparisons to be used in the clustering algorithm.

McCallum et al. (2000) provides an early example of the first approach, constructing what they refer to as *canopies*. Canopies are constructed by using a cheap similarity measure to generate overlapping subsets of record pairs, in essence generating a blocking scheme using indexing by disjunctions. More computationally expensive comparisons can be made between record pairs within the canopies. Finally, a clustering algorithm such as k-means or greedy agglomerative clustering is used to construct clusters of records. The clustering is executed using the more costly comparison metric, which is assumed to be more accurate, and the similarity between records which appear in none of the same canopies is set to zero. This approach yields identical results to those that would be obtained from applying the more costly comparison metric to all record pairs, as long as the clustering is covered by the canopies. For a clustering to be covered by the canopies all records which share a cluster in the clustering must also share at least one canopy. That is, if all records that would be placed in the same cluster given the full distance metric are grouped together by at least one canopy.

In contrast to the comparison-based approach taken by canopies, the second approach for generating clusters of records involves constructing a function that directly maps each record to a cluster. This is achieved by partitioning the space and then constructing a mapping that assigns each record to a partition. Each partition is then treated as a block, with links allowed only between records assigned to the same block. A naive and fast, implementation of this approach can be achieved by using a partitioning algorithm such as a k-d tree (Marchant et al., 2019). However, a more common approach to constructing such a mapping is to employ *locality sensitive hashing* (LSH) (Liang et al., 2014; Steorts et al., 2014). In most settings where hash functions are employed the goal is for similar values to be mapped to values that are essentially uncorrelated. In LSH the hash function is instead constructed in a manner such that similar input values are mapped to the *same value* with high probability. The entire record can then be used as an input value to the locality sensitive hash function and the resulting value used as a blocking key.

While including more information than provided by a single feature makes LSH potentially more robust than standard blocking, clustering methods do suffer from the same drawback of partitioning the records into disjoint blocks as traditional blocking. Namely, that if two truly matching records are mapped to different blocks there is no way to link them, no matter the quality of the model. They do, however, recover the benefit of allowing link structure to be estimated separately within each block or partition, often an easier task as we discuss further in both Chaper 2 and Chaper 3. As with other methods for constructing a blocking scheme, clustering approaches create a trade-off between larger clusters, which are more likely to contain all or most of the true links, and smaller clusters which offer larger computational benefits. Without access to ground truth labels, many approaches for determining how to weight this trade-off are, by necessity, ad-hoc. For many problems the more important features may be easy to identify, names are almost always important when matching lists of people. In other applications identifying the most important features may be less straight forward. However, if ground truth is known, for even on a subset of the data, then a variety of methods can be used to supplement background knowledge in the choice of blocking scheme.

1.3.4 Supervised Methods

In the rare problem where training data is available it can be used to inform the choice of blocking in a variety of ways. For example, Kelley (1985) suggests considering the space of admissible blocking schemes, those that reduce the number of record pairs sufficiently that the problem is computationally tractable, and then selecting the one that excludes the smallest share of matching record pairs. While Kelley (1985) uses a model-based estimate of the false non-match rate, these quantities are more easily, and generally more accurately, estimated when training data is available. Training data can also be used to select hyper-parameters, such as the number or maximum size of clusters, through standard methods such as cross validation. In general the reduction ratio, the fraction of the full $n_A \times n_B$ set record pairs that is excluded by the blocking scheme, and pairs completeness, the fraction of true matches included by the blocking scheme, are useful in evaluating the efficacy of any blocking scheme (Christen, 2012a).

Approaches to learning, as opposed to evaluating, blocking schemes have also been developed and fall broadly into two categories: (1) those that estimate a classification model for identifying record pairs likely to be linked and use it to aid in the construction of a blocking scheme and (2) methods that attempt to directly learn a blocking scheme from a large set of possible choices. Examples of the first approach include Cohen and Richman (2002) and Ventura and Nugent (2014). In the former a classifier, referred to as a *pairing function*, is used to evaluate the similarity of record pairs. A greedy clustering algorithm is then applied to split the records into a pre-specified number of clusters. In the latter training data is used to construct an ensemble classifier, the scores of which are then used as the similarity metric to which a hierarchical clustering is applied. These approaches contrast with approaches that select blocking keys themselves using

data. Michelson and Knoblock (2006) develop an algorithm that does this in a greedy fashion while Bilenko et al. (2006) use training to learn what is essentially a blocking scheme based on indexing by disjuntions (which they refer to as disjunctive normal form). The draw back to this approach is that the choice of the best possible subset of blocking keys is NP hard and so it can not generally be solved exactly.

More recent work has focused on learning blocking schemes even without training data. Kejriwal and Miranker (2013) employ what is refered to as *weakly labeled* training data (we note that this terminology appears in Bilenko and Mooney (2003) albeit only for negative training examples), record pairs that are above a predefined similarity threshold for some similarity metric. These weakly labeled record pairs are then used to learn a blocking scheme, as could be done with actual training data. This approach is somewhat fraught in that it relies on pre-determined thresholds to apply the weak labels but the authors claim that the thresholds are robust across a variety of problems (Kejriwal and Miranker, 2013). Several other authors have developed related procedures, often for application to problems involving record linkage in real time (Giang, 2015; Ramadan and Christen, 2015).

Selecting an appropriate blocking scheme is a crucial first step in the record linkage processes, and essential to making the problem tractable. The choice of blocking scheme is important both for which record pairs it includes, and which record pairs it excludes. This has ramifications for the estimation processes, particularly of unsupervised models, as we discuss further in Section 1.5.4.

1.4 Comparing Record Pairs

After determining a blocking scheme and generating the resulting set of candidate record pairs the second step in a typical record linkage processes is to generate a set of comparisons for each record pair. More formally, consider two records a and b drawn from A and B respectively. We compute a set of d comparisons between fields contained in A and fields in B. denote this set:

$$\gamma_{ab} = \{\gamma_{ab}^1, \gamma_{ab}^2, \ldots, \gamma_{ab}^d\}. \tag{1.2}$$

The use of comparisons between fields allows flexibility in the model as the choice of comparisons can be tailored to a given dataset. For some field types the selection of the comparison may be straight forward, taking an absolute difference in values may be appropriate for many continuous values and categorical fields can be compared using exact matching. However, for others field types including string-valued fields, dates, and locations determining the appropriate comparison metric may be considerably more challenging. Here we provide a brief overview of string comparison metrics, which are used widely in PRL and have constituted the primary area of focus in the existing literature. For a more comprehensive overview of both string

comparisons and comparisons metrics for other data types see Yancey (2005); Herzog et al. (2007); Christen (2012a) and citations therein.

1.4.1 String Comparisons

While the use of comparisons means that generative models do not need to be produced for hard to model fields, in many cases choosing an appropriate comparison metric is still challenging. Differences in string fields can be particularly challenging to model because there is often little information as to the source of the observed discrepancies. Typographical or transcription errors may be relatively straight forward to model but more general error processes are much harder to model without access to massive sets of labeled training data. In particular, string values fields such as names and addresses are common in PRL problems and are hard to model distributionally.

A common type of error in string fields that must be dealt with in PRL problems is the typographical error, when a value has been enter incorrectly or possibly omitted (Herzog et al., 2007). A natural approach for comparing string fields is therefore to employ edit distances (Winkler, 1990). One of the most basic string metrics is the Levenshtein distance, which counts the number of insertions, deletions or substitutions that must be made for two strings to match. The number of required edits can then be normalized by the greater of the two lengths of the strings being compared to yield a value between 0 and 1. A generalization of this measure known as the Damerau-Levenshtein edit distance allows for transpositions as well.

A variety of extensions to the basic edit distance framework exist. In general, such extensions allow different weights for different types of edits to be incorporated into the distance. Such weighting schemes can vary from the relatively simple, assigning different weights to each edit type, to more complex, assigning different weights to specific transpositions. This can be particularly effective if background information about expected errors in the fields is available. A common scenario is that one or both of the datasets was generated from scanned paper records and then converted to machine readable text using optical character recognition (OCR). With this type of data some common errors such as the letter “o” being replaced with a “0” or an “l” being replaced with a “1” (or vice versa) are known to occur frequently. Thus, one may wish to down weight the effect making such an edit has on the resulting distance (see Christen (2012a) and references therein). In other scenarios the field values may have been dictated and comparisons between strings that are spelled differently but would sound the same if spoken can be identified with phonetic encodings, discussed later in this section. Edit distances such as the Smith-Waterman metric can take advantage of such information (Christen, 2012a).

Another common approach is to split the string into multiple pieces or *tokens*. If the string field is expected to contain multiple words, such as commonly occurs with addresses, then the tokens often correspond to the individual words. For string fields which may contain only a single word a common approach is to generate

Q-grams, splitting the string into sequences of exactly q characters using a sliding window. For example, the set of 3-grams generated by the string "terry" will be {ter, err, rry}. Regardless of the generation process, once the tokens are generated the full set generated by each string can be compared, typically using a set-based metric such as a Jaccard or overlap coefficient (Christen, 2012a). Such comparisons are often more robust to changes in the order of the tokens and to omissions in the middle of the string than edit based distances. Token based comparisons can also be combined with similarity measures between the tokens. One example is the Monge-Elkan comparison metric, which combines a token-based approach with a similarity measure between tokens.

An important extension to token based string comparisons is to consider not just the set of tokens (or words) contained within a string but the frequency of occurrence. Frequencies are used to increase the weight given to two types of terms, those that occur multiple times within a string, and weights associated with uncommon terms, which may be a stronger marker of similarity than relatively common terms. For a specific term (or token) the determinants of such distances are usually the term frequency (TF), with tokens that occur more frequently within a string being given a higher weight, and the inverse document frequency (IDF), which assigns lower weight to those terms that occur relatively frequently across some larger set of records or documents. The combination of these two weights is typically abbreviated TFIDF (Cohen et al., 2003). The background frequencies may be based on the empirical distribution observed in the full set of records to be linked or from some larger database which is though to provide an appropriate baseline. In the case of names this may be, for example, based on known frequencies for the target population from a source such as a national census. Cohen et al. (2003) introduced a *soft-TFIDF* measure which applies a (user-specified) string similarity measure to compare tokens, as in the Monge-Elkan similarity measure, and then additionally weights the comparisons based on their relative frequencies with the weights combined via a cosine distance. This metric has proven particularly popular within the computer science literature related to record linkage.

Thus far we have discussed general sting metrics which may in principle be applied to any string field. However, for fields containing names a variety of more specialized metrics have been developed. one of the most common name specific string metrics is the Jaro string metric, which relies on a combination of edit distances and q-grams was developed at the US Census Bureau (Yancey, 2005). An even more widely used extension, the Jaro-Winkler string metric, adds additional weight to agreements observed among the first four characters of the string as empirical research at the US Census Bureau has shown that typos in names occur more frequently towards the end of the word. A second common approach, at least for English language, is the idea of using phonetic encodings such as Soundex (Christen, 2012a). Such encodings attempt to map strings to a numeric sequence corresponding to spoken sounds. This treatment is robust to names (and other words) such as "Sean" and "Shawn" which may sound the same when spoken but which have significantly different spellings. In some settings Soundex can also be applied to fields containing other English words

We have covered only a small portion of the available literature on string metrics but have sought to emphasize some of the important considerations when selecting an appropriate metric. Ideally the researcher will have some knowledge as to the fields contained in the data and can pick a string metric which is robust to the expected error types. As in other decisions to be made in developing a PRL algorithm, if ground truth is known for even a subset of appropriate data a variety of metrics can be tested and the best performing metric selected. If a supervised classification model is to be employed, it may be effective to compute multiple comparisons for each available field. One downside to this approach is that it may be computationally costly. We have not discussed the computational complexity of the string similarities addressed or algorithms for computing them, a good overview of both is provided by Charras and Lecroq (2004).

1.5 Models for Unsupervised Record Linkage

Once a set of comparisons has been constructed, a PRL model can be applied. In this work we focus on unsupervised models for PRL. The focus on unsupervised methods is driven both by the cost of acquiring training data for use with supervised methods and the difficulty of applying training data broadly. In general, the types of errors that a supervised method can model successfully will be specific to the dataset and therefore may not generalize to other record linkage problems. In contrast, unsupervised methods may perform somewhat worse that supervised methods on a specific problem but are more broadly applicable.

Throughout our review of PRL models we will generally assume the full $n_A \times n_B$ set of comparisons vectors has been computed. Thus, the model will be fit on a set $n_A \times n_B$ comparison vectors, with each comparison vector containing d comparisons. We denote the full set of comparison vectors Γ. This is the data that is the basis for our inferences about the link structure C. As discussed in Section 1.3 it is generally not feasible to compute this full set of comparisons but we wil proceed with this framework as it is consistent with how most models have been constructed in the literature. In Section 1.5.4 we discuss issues that can arise when these models are applied to a set of comparisons generated by a blocking scheme and some techniques for mitigating this bias.

1.5.1 Fellegi-Sunter

The PRL problem was first introduced by Newcombe et al. (1959); Newcombe and Kennedy (1962) but Fellegi and Sunter (1969), (here after FS), introduced a framework for modeling record comparisons and a decision rule for classifying record pairs into matches or links and others non-matches. Consider the product set of records pairs $A \times B$ generated by two files. Assuming that A and B contain some entities in common this set will consist of both matching and non-matching records pairs. The framework introduced by FS

models the distribution as a two component mixture model:

$$\begin{aligned} M &= \{(a,b) : \text{record } a \text{ matches record } b\} \\ U &= \{(a,b) : \text{record } a \text{ does not match record } b. \end{aligned} \tag{1.3}$$

As defined in (1.3) the M component contains comparisons between records that correspond to the same entity, while the U component contains comparisons between record pairs that correspond to different entities. For a link structure C, defined in Section 1.2, the M component corresponds to record pairs (a,b) such that $C_{ab} = 1$, and the U component record pairs such that $C_{ab} = 0$. The distribution of comparisons between record pairs is then modeled conditional on component membership. For a comparison vector g we model two densities determined entirely by component membership:

$$\begin{aligned} m(g) &= \Pr\left(\gamma_{ab} = g \mid (a,b) \in M\right) \\ u(g) &= \Pr\left(\gamma_{ab} = g \mid (a,b) \in U\right). \end{aligned} \tag{1.4}$$

Intuitively, these components should be well separated in most problems as the majority of record pairs in the M component should show high levels of similarity across most if not all comparisons. While, record pairs in the U component should agree on field only incidentally. We will refer to the log of the ratio of these densities, as defined in (1.5) as the *weight*.

$$w_{ab} = \log\left(\frac{m(\gamma_{ab})}{u(\gamma_{ab})}\right) \tag{1.5}$$

We note, as did FS, that we need not concern ourselves with the case where $m(g) = u(g) = 0$ since it will not arise in practice. Similarly, comparison vectors g for which only one of $m(g)$ or $u(g)$ is greater than zero can be trivially assigned to the correct component. We therefore limit our discussion to cases where $m(g) > 0$ and $u(g) > 0$. When a comparison vector g indicates significant agreement between the fields of two records we generally expect $m(g) >> u(g)$, so if $\gamma_{ab} = g$ then $w_{ab} >> 0$. In this case w_{ab} summarizes information about the *relative* likelihood of a record pair being a match versus non-match, mapping the full comparison vector to a single composite score (Newcombe and Kennedy, 1962).

High weight comparison vectors are much more likely to correspond to a matching record pair than a non-matching one, while low (negative with large absolute value) weight comparison vectors are much more likely to correspond to a non-matching record pair. We might therefore imagine a procedure in which all high weight record pairs are classified as matches, and all low weight record pairs are classified as non-matches. FS developed a formal version of this procedure where they define thresholds T_μ and T_λ such that a record pair (a,b) with $w_{ab} \geq T_\mu$ is classified as a match ($\widehat{C}_{ab} = 1$), and a record pair with $w_{ab} \leq T_\lambda$ is classified as a non-match ($\widehat{C}_{ab} = 0$). If $T_\lambda = T_\mu$ then this procedure with yield a classification for all record pairs, record

pairs with $w_{ab} = T_\mu = T_\lambda$ can be assigned proportional to achieve the desired rates. If however, $T_\lambda < T_\mu$ then any record pairs where $T_\lambda < w_{ab} < T_\mu$ are given an *indeterminate* match status and are evaluated manually. A summary of this procedure is shown in (1.6).

$$C_{ab} = \begin{cases} \text{Match} & T_\mu \leq w_{ab} \\ \text{Indeterminate} & T_\lambda < w_{ab} < T_\mu \\ \text{Non-match} & w_{ab} \leq T_\lambda \end{cases} \tag{1.6}$$

The thresholds T_μ and T_λ are set to simultaneously control the false positive or false match rate μ (the probability a non-matching pair is classified as a match) and false negative or false non-match rate λ (the probably a matching pair is classified as a non-match). This procedure was developed with the goal of automatically making the easy designations, near perfect matches and obvious non-matches (Jaro, 1989). FS prove that this procedure is optimal, minimizing the size of the indeterminate set, for given values of m and u and error rates μ and λ. Crucial to this result is that the indeterminate cases can be resolved perfectly through clerical review, an assumption which may not always be reasonable as we discuss in Chapter 3. The FS decision rule also requires that $m(g)$ and $u(g)$ be known which is in practice rarely the case.

1.5.2 Conditional Independence

The saturated model implied by (1.4) is typically not identified without additional restrictions on the parameter space. One frequently used restriction is to assume that the comparisons are independent conditional on component membership. Under this assumption the distributions in (1.4) separate by field and reduce to (1.7).

$$\begin{aligned} m(g) &= \prod_{j=1}^{d} \Pr\left(\gamma_{ab}^j = g^j \mid (a,b) \in M\right) \\ u(g) &= \prod_{j=1}^{d} \Pr\left(\gamma_{ab}^j = g^j \mid (a,b) \in U\right) \end{aligned} \tag{1.7}$$

While the conditional independence assumption fixes the identifiability issue, it often does not hold in practice (Smith and Newcombe, 1975). There is some disagreement in the literature as to how problematic violations of this assumption are. Winkler (1985) finds that it is not crucial that the assumption hold for some practical applications. However, Kelley (1986) finds that the error rate control of the FS decision rule, described in (1.6), is sensitive to even small violations of the conditional independence assumption.

Binned Comparisons

Many early applications made use of binary (0/1) comparison between fields (see e.g. (Jaro, 1989; Fortini et al., 2001)) within the conditional independence framework. While, the use of binary comparisons is less restrictive than simply employing exact matching, a similarity threshold can be used to distinguish between agreement and non-agreement (Winkler, 1990), it still severely limits the flexibility of the model. A common relaxation from binary comparison model involves binning a set of continuous comparisons (Winkler, 1990; Sadinle, 2017). Such models take continuous similarity metrics, generally normalized to be contained in the $[0, 1]$ interval, and partition the interval into k separate intervals. This provides discretizes the continuous comparison, transforming the continuous comparison into categorical one. The distributions within the M and U components can then be modeled using a Multinomial distribution. We define the probability that feature j records a similarity level (category) of h for each component as:

$$\begin{aligned} m_{jh} &= \Pr\left(\gamma_{ab}^{j} = h \mid C_{ab} = 1\right) \\ u_{jh} &= \Pr\left(\gamma_{ab}^{j} = h \mid C_{ab} = 0\right). \end{aligned} \tag{1.8}$$

If we assume, as before, that a total of d comparisons are generated and that comparison j is partitioned into k_j possible levels then the model given in(1.7) can be written as:

$$\begin{aligned} m(g) &= \prod_{j=1}^{d} \prod_{h=1}^{k_j} m_{jh}^{\mathbb{1}(g_j = h)} \\ u(g) &= \prod_{j=1}^{d} \prod_{h=1}^{k_j} u_{jh}^{\mathbb{1}(g_j = h)}. \end{aligned} \tag{1.9}$$

We also define the parameter set $m_j = \{m_{j1}, \ldots m_{jk_j}\}$ (with u_j defined similarly). We adopt the model given by (1.9) referring to the set of parameters within the M and U components as the $m-$*parameters* and $u-$*parameters* respectively. To denote both sets of parameters we adopt the terminology *matching parameters*.

The advantages of this formulation are twofold: (1) the transformation of continuous similarity measures into discrete ones greatly simplifies model estimation, this is particularity in an unsupervised setting, and (2) the model can always be made more flexible by increasing the number of bins allowing a better approximation of the underlying continuous distribution. Originally developed by Winkler (1990), this model continues to be frequently employed (see e.g. Sadinle (2017)). A limited version of this model, allowing for only three bins per comparison and imposing additional constraints on the estimated parameters, was recently implemented by Enamorado et al. (2019).

Relaxing Conditional Independence

While models relying on the conditional independence assumption are widely employed, the assumption is clearly violated in many applications. In such cases, work has been done to relax this assumption and develop a more flexible class of models. Notably a model introduced by Winkler (1993) included up to three-way interactions between comparisons. Similarly, Thibaudeau (1993) employed a model which accounted for correlations between fields, particularly those pertaining to household membership. Thibaudeau (1993) also included different sets of interactions within the M and U components as the correlation structures were thought to differ. In practice, these models often outperform those which assume complete independence between comparisons.

A related approach is to introduce additional components into the mixture model (Winkler, 1995; Larsen and Rubin, 2001). The additional flexibility given by introducing addition components allows for models which assume a simpler structure within the components (e.g. those without interactions) to perform better. The components can then be aggregated into matching and non-matching categories. (Winkler, 1995) used this approach where the three components were interpreted as (1) matches, (2) non-matches at the same address, likely household members, and (3) non-matches as different addresses. A general discussion of multi-component log-linear models is provided by Larsen and Rubin (2001) and references therein.

Frequency Weights

A final commonly employed method for improving model performance, particularly in the modeling of recurring words such as names, is to adjust the weights to account for word frequency. The intuition behind this approach is that agreement on common names such as "smith" is more likely to happen in a non-matching pair than it is for uncommon names such as "zabrinksy" (Winkler, 1988). This approach is similar to the similarity measures that incorporate inverse document frequencies such as the soft-TFIDF string metric discussed in Section 1.4.1. With the obvious different that the information is incorporated at the modeling stage rather than within the similarity metric. As with the IDF weights in some applications it may make sense to use external sources for reference frequencies, such as the US Census for names, while in others it may be sufficient to use the empirical frequencies observed in the data. For a more thorough overview of frequency weighting see Winkler (1995).

1.5.3 Estimation

Fellegi and Sunter (1969) proposed computing the m and u parameters 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 expectation-maximization (EM) algorithm (Dempster et al., 1977)

to maximize the likelihood given by:

$$L(m, u, \pi \mid \Gamma) = \prod_{(a,b) \in A \times B} \pi m(\gamma_{ab}) + (1 - \pi) u(\gamma_{ab}), \tag{1.10}$$

where $\pi = \Pr(C_{ab} = 1)$, treating each comparison vector as an independent observation (Winkler, 1988). While convenient for estimation this approach admits a large range of estimates which are inconsistent with the one-to-one matching assumption. We note for example that in a one-to-one matching framework the constraint $\pi \leq \min(n_A, n_B)/(n_A \times n_B)$ should always hold. While it is possible to enforce this particular constraint within an EM framework (for example via regularization) the point is that(1.10) does not fully capture the structure imposed by one-to-one matching. A method for resolving to a one-to-one matching estimate in a post-hoc fashion is described in 1.5.6.

One reason for the popularity of the EM algorithm in the estimated of record linkage models is the ability to account for missing data (Dempster et al., 1977), a common occurrence in nearly all record linkage applications. This is usually done via a missing at random assumption (Sadinle, 2017). Within this framework a variety of algorithms have been developed for different models, including frequency weighting (Winkler, 1988) and convex constraints on the parameter space (Winkler, 1993).

EM estimates of the weights can work well in settings where there are many attributes available for matching and where there is a high degree of overlap between the datasets, with many records in A also appearing in B (Winkler, 2002). In contrast, when the two files have few fields in common, there is s significant error rate in the fields available, or when there is limited overlap between the files, EM-estimated weights can perform quite poorly (Winkler, 2002; Tancredi et al., 2011; Sadinle, 2017). More reliable weights can be obtained by estimating the m and u parameters while accounting for the one-to-one matching constraint. We introduce a new procedure which addresses this failing in Chapter 2.

1.5.4 Parameter Bias from Blocking

In our discussion of Many models for PRL, we have so far assumed that all $n_A \times n_B$ comparison vectors will be used in estimating C. However, in most practical applications some sort of blocking scheme is employed and thus only a subset of the record pairs are considered for linking. The use of a blocking scheme censors the set of record pairs seen by the model, which can significantly alter the resulting estimates if not accounted for. In theory this problem could be addressed as Fellegi and Sunter (1969) recommend, by restricting inference to the specific subspace of interest (i.e. record pairs within the blocking scheme). Kelley (1986) notes that different thresholds T_μ and T_λ should be used in the Fellegi-Sunter decision procedure in the presence of blocking but that it is unclear how they should be selected. Under a traditional blocking scheme each block can be modeled separately, with the information potentially combined in a hierarchical fashion

(Larsen, 2002). It is less clear how to achieve this if a more general blocking scheme, such as indexing by disjunctions, is employed.

As noted in Section 1.3, it is usually not the case that there are no matching record pairs outside of the blocks (Jaro, 1989). Thus, blocking increases the number of false negatives, since all record pairs outside of the blocking will be automatically declared non-matches. B, blocking can also affect the model estimates in a less obviously way: by limiting the set of comparison vectors which the model uses to draw its conclusions. For unsupervised models of the sort discussed in Sections 1.5.1 and 1.5.2, this introduces bias into the estimated parameters (Murray, 2016), particularly those in the U component. The mechanism for this is easy to observe, suppose blocking is performed so that only records with similar first names are considered. As a result, every single comparisons vector computed will denote some minimal level of similarity on first name. A model trained only on the computed comparison vectors will therefore estimate that, even for non-matching records, first name always displays at least the minimum level of similarity. This bias can be quite severe in practice (Neter et al., 1965). Furthermore, blocking can induce correlations between the comparisons, violating the conditional independence many unsupervised models assume (Thibaudeau, 1993).

Fortunately, correcting for this bias in the u parameters is relatively straight forward. Since C_{ab} is fixed to zero for all record pairs outside of the blocking scheme, the distribution of these "missing" comparison vectors can be easily estimated by randomly sampling record pairs excluded from the blocking scheme (typically the vast majority of all record pairs) (Jaro, 1989). Under a conditional independence assumption, a more precise adjustment is possible as the missing comparisons need only be generated marginally. That is, for each record pair outside of the indexing or blocking scheme, it is not necessary to compute the comparisons individually for each record pair. Instead, to calculate the frequency with which each similarity level occurs for each feature it is sufficient to compute the set of unique comparisons separately for each field. This is a result of the fact that under the conditional independence assumption the likelihood factors in such a way that only the marginal frequencies with each component (match and non-match) are necessary, as can be seen in (1.9). Computing these marginal frequencies is much more tractable as it can be done by computing similarities only for observed unique values of each field and weighting appropriately. As we demonstrate in Chapter 3, removing the bias from the u parameters can result in a substantial improvement in model performance.

1.5.5 Supervised models

In this work we restrict our focus to unsupervised models for PRL. However, if training data is available, or can be easily acquired, then it is easily applied to PRL problems. Perhaps the most straight forward use is to use training data to train a standard classification model, which can be applied to the comparison of record pairs. Popular classifiers include support vector machines (Christen, 2008a; Fu et al., 2011)

and random forests (Ventura and Nugent, 2014; Frisoli and Nugent, 2018). The use of supervised models also eliminates the need for the conditional independence assumption and can even allow the inclusion of additional information, such as household membership, which span multiple observations and cannot be incorporated into models which assume conditional independence. The inclusion of such additional features often significantly improves model performance (Frisoli and Nugent, 2018).

As with other classification problems, care must be take to ensure that the training set is representative of the data to be matched (Winkler, 2002). Given the costliness of the acquisition of labeled training data, methods have been developed for the selection of appropriate training examples. Larsen and Rubin (2001) developed a method for iteratively combining unlabeled data with labeled examples although, at least in some cases, this procedure has been found to require a prohibitive number of training examples before converging (Enamorado, 2018). Christen (2008b) developed a two-step method for record linkage in which the first step involves automatically selecting training examples which are used to train a classifier employed in the second step of the procedure. More recently, interest has been shown in applying active learning techniques to the selection of training data ((Christen, 2012a, Chapter 6),Enamorado (2018)).

Finally, we note that while error rates can generally be estimated for classification models, this provides, at best, a measure of the uncertainty for the misclassification rate of a single record pair. This generally *does not* address the goal of quantifying uncertainty in the full link structure. To our knowledge quantification of uncertainty in the link structure has, to date, primarily been achieved through the use of Bayesian methods, although we discuss several proposed alternative approaches in Section 1.7.

1.5.6 Enforcing One-to-one Matching

As originally constructed, neither the methods for inferring the matching parameters, nor the FS decision rule for generating an estimate of the matching structure C necessarily produce an estimate consistent with one-to-one matching. In particular, in the FS decision rule, if $w_{ab} > T_\mu$ and $w_{ab'} > T_\mu$, for record pairs (a, b) and (a, b'), then both are declared links. Even though this violates one-to-one matching. Jaro (1989) proposed a three-step approach for adapting the Fellegi-Sunter decision rule to respect one-to-one matching. In the first step are estimated by maximizing (1.10). In the second step C^*, a link structure consistent with one-to-one matching is computed by solving the following assignment problem:

$$
\begin{aligned}
C^* = &\max_C \sum_{i=1}^{k}\sum_{j=1}^{k} C_{ij}\hat{w}_{ij} \\
&\text{subject to } C_{ij} \in \{0,1\} \\
&\qquad \sum_{j=1}^{k} C_{ij} = 1, \qquad \text{for } i = 1,\ldots,k \\
&\qquad \sum_{i=1}^{k} C_{ij} = 1 \qquad \text{for } j = 1,\ldots,k.
\end{aligned}
\tag{1.11}
$$

Where the full set of weights $\widehat{W}$ was assumed to be a square $k \times k$ matrix in the original formulation given by Jaro (1989). Since A and B are rarely the same size in practice, Jaro (1989) proposed augmenting $\widehat{W}$ with additional columns or rows to make the problem square with the values of the augmented weight set to negative values larger than any of the estimated weights to avoid assignment. In the final step, the matching estimate $\widehat{C}$ is obtained from C^* by setting $\widehat{C}_{ab} = C^*_{ab}\mathbb{1}(\hat{w}_{ab} \geq T_\mu)$, where T_μ plays the same role as in the FS decision rule (1.6).

While this procedure leads to an estimate of C that respects one-to-one matching it comes with several drawbacks. First, it is critically dependent upon good estimates of the matching parameters. However, it has been observed empirically that failing to enforce the one-to-one constraint during estimation can lead to poor estimates of the matching parameters (Tancredi et al., 2011; Sadinle, 2017). Second, attempting to match all records, as initially done by the assignment problem, and then removing those that appear to be non-matches may result in a sub-optimal set of estimated matches, as we shown in Chapter 2. Finally, the procedure outlined in Jaro (1989) is inefficient from a computational perspective unless is it combined with a relatively precise traditional blocking scheme. While several greedy procedures for resolving a set of matches consistent with one-to-one matching have also been employed in the literature (Chipperfield and Chambers, 2015; Enamorado et al., 2019) these are typically employed on the basis of computational convenience rather than some sense of optimality. In Chapter 2 we introduce a new estimator which mitigates all of these issues by jointly estimating m, u, and C using a modified assignment problem which can be solved much more efficiently than the one in 1.11. But, in the existing literature, the most common solution to the first two problems appears to be full Bayesian modeling.

1.6 Bayesian 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 quantifies uncertainty in the link structure (through posterior samples of C) that can be propagated

to subsequent inference. As we discuss in Section 1.7 there are other approaches to propagating uncertainty in the estimate of C but none is as general.

Early approaches to Bayesian PRL utilized the same comparison-vector based model as in (1.7), 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; Marchant et al., 2019) 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., 2017b). While we focus on comparison based models, the method we introduce for scaling Bayesian record linkage in Chapter 3 is model-agnostic and could be utilized in any of these models.

Modeling C directly

One of the main advantages of Bayesian methods is that the link structure C is modeled directly, rather than being inferred separately from the parameter estimation as described in Section 1.5.6. Additionally, estimation methods introduced in Section 1.5.3 generally assume that the observed record pairs are independent. Under a one-to-one matching assumption this is clearly incorrect as $C_{ab} = 1$ implies that $C_{ab'} = 0$. We can easily observe this difference in the likelihood by noting that only the proportion of matches, denoted π is present in (1.10). In contrast, in Bayesian models, where a specific value is assigned to C throughout, the likelihood includes the interaction between C and specific comparison vectors:

$$\ell(m, u, C \mid \Gamma) = \sum_{a,b \in A \times B} \log\left(m(\gamma_{ab})\right) C_{ab} + \log\left(u(\gamma_{ab})\right)(1 - C_{ab}) \tag{1.12}$$

$$= \sum_{a,b \in A \times B} \left[\log\left(m(\gamma_{ab})\right) - \log\left(u(\gamma_{ab})\right)\right] C_{ab} + \sum_{a,b \in A \times B} \log\left(u(\gamma_{ab})\right)$$

$$= \sum_{a,b \in A \times B} w_{ab} C_{ab} + \sum_{a,b \in A \times B} \log\left(u(\gamma_{ab})\right). \tag{1.13}$$

We show the equivalence of the formulations of (1.12) and (1.13) to draw the connection with (1.11). It is obvious from 1.13 that to maximize (1.12) , with respect to C, one need only consider the weights not the matching parameters. Although we note that under most models C may contain some rows, (records in dataset A) which are unmatched in contrast to the approach taken by Jaro (1989), a point we discuss further in Chapter 2. Enforcing one-to-one matching incorporates significantly more structure into the problem and can lead to significantly improved estimates, particularly when the problem fails to meet the criteria set out by Winkler (2002) for EM algorithm based estimates to perform well.

1.6.1 Priors for one-to-one matching

Within a Bayesian framework one-to-one matching is enforced through the use of an appropriate prior distribution over C. In general, constructing an informative prior for C is challenging, as it is difficult to incorporate the information available before the analysis as to which record pairs are likely to correspond to links, problem we address in Chapter 4. However, in many applications some prior information will be available on the expected number or proportion of records that will be matched. It is therefore common to construct a prior by first defining a distribution over the number of matches, which we denote L. The prior probability of the link structure containing L links is then split uniformly across the set of link structures containing exactly L links. This second feature is typically justified on the basis that it ensures that the prior is invariant to the ordering of the records within either file. Zanella (2019) refers to this property as *invariance under permutation*.

The number of possible values of the link structure C with L matches can be written in closed form for datasets of known size. For a $n_A \times n_B$ link structure C the number of possible link structures with exactly L matches is given by:

$$N(L, n_A, n_B) = \frac{n_A! n_B!}{L!(n_A - L)!(n_B - L)!} \tag{1.14}$$

for $0 \leq L \leq \min(n_A, n_B)$. We note the factorial nature of the relationship between the number of possible link structures $N(L, n_A, n_B)$ and L. This means that even if $\pi(L)$, the prior probability of L matches and $\pi(L+1)$ take similar values $\pi(C_L)$, the prior probability of a *specific* value of C with L links, and $\pi(C_{L+1})$ may differ significantly.

Perhaps the most common prior distribution, which we refer to as a *Beta-bipartite* distribution, was introduced by Fortini et al. (2001, 2002), generalized by Larsen (2005, 2010) and studied further by Sadinle (2017). The Beta-bipartite distribution can be derived using the general approach described above. The derivation first places a Beta-binomial distribution on the number of links, or equivalently places a Beta prior distribution over the proportion of records (from the smaller file) that are expected to be linked and then models the number of links as a binomial distribution. Then, given L, the prior density is split uniformly matching matrices C with exactly L links, so that:

$$\begin{aligned} \pi(C \mid \alpha, \beta, n_A, n_B) &= \frac{\text{Beta-binomial}(L, n_A, \alpha, \beta)}{N(L, n_A, n_B)} \\ &= \frac{(n_B - L)!}{n_B!} \frac{\mathrm{B}(L + \alpha, n_A - L + \beta)}{\mathrm{B}(\alpha, \beta)}. \end{aligned} \tag{1.15}$$

Zanella (2019) developed an alternative prior the same general process. First defining a truncated Poisson distribution with parameter λ which models the distribution of the total number of unique entities across the two datasets. Each entity then appears in both datasets with probability p_{match}, in only A with probability

$(1 - p_{match})/2$ and in only B with probability $(1 - p_{match})/2$. The resulting density is:

$$\begin{aligned}\pi(C \mid \lambda, p_{match}, n_A, n_B) &= \frac{\text{Poisson}(n_A + n_B - 2L, \lambda)\text{Multinomial}\,(n, p)}{N(L, n_A, n_B)} \\ &= \frac{e^{-\lambda}\lambda^{n_A+n_B-L}}{n_A! n_B!}\,(p_{match})^L \left(\frac{1 - p_{match}}{2}\right)^{n_A+n_B-2L}\end{aligned} \tag{1.16}$$

where the Multinomial parameters are $n = \{L, n_A - L, n_B - L\}$ and $p = \{p_{match}, \frac{1 - p_{match}}{2}, \frac{1 - p_{match}}{2}\}$. Zanella (2019) further defines priors over p_{match} and λ allowing posterior inference to be conducted on these parameters.

A final example of a prior that enforces one-to-one matching is given by Green and Mardia (2006). The derivation followed by Green and Mardia (2006) again proceeds in a similar fashion. They first assume a homogeneous Poisson process with a rate λ over a volume v. Arriving points are then categorized as appearing in both A and B, appearing only in A, appearing only in B, or appearing in neither A nor B with probabilities $\rho p_a p_b$, p_a, p_b, and $1 - p_a - p_b - \rho p_a p_b$ respectively. The parameter ρ controls the correlation between appearing in A and B. Conditional on the values of n_A, n_B, and the (unobserved) total number of arrivals, the number of entities appearing only in A, only in B, and appearing in both A and B means that the counts follow independent Poisson distributions and therefore:

$$\begin{aligned}\pi(L \mid \lambda, v, \rho, p_a, p_b, n_A, n_B) &\propto \text{Poisson}(n_A - L, \lambda v p_a)\text{Poisson}(n_B - L, \lambda v p_b)\text{Poisson}(L, \lambda v \rho p_a p_b) \\ &= \frac{e^{-\lambda v p_A}(\lambda v p_A)^{n_A - L}}{(n_A - L)!}\frac{e^{-\lambda v p_B}(\lambda v p_B)^{n_B - L}}{(n_B - L)!}\frac{e^{-\lambda v \rho p_a p_b}(\lambda v \rho p_a p_b)^L}{L!} \\ &= \frac{e^{-\lambda v (p_A + p_B + \rho p_A p_B)}(\lambda v)^{n_A + n_B - L} p_A^{n_A} p_B^{n_B} \rho^L}{(n_A - L)!(n_B - L)! L!} \\ &\propto \frac{(\rho/(\lambda v))^L}{(n_A - L)!(n_B - L)! L!}.\end{aligned} \tag{1.17}$$

Interestingly, we can observe from the final form of (1.17) that this implies:

$$\pi(L \mid \theta, n_A, n_B) \propto e^{-\theta L} N(L, n_A, n_B) \tag{1.18}$$

where we define $\theta = -\log(\rho/(\lambda v))$. As with other priors Green and Mardia (2006) assume that, conditional on L the distribution is uniform over the possible link structures and therefore:

$$\pi(C \mid \theta, n_A, n_B) \propto \frac{e^{-\theta L} N(L, n_A, n_B)}{N(L, n_A, n_B)} = e^{-\theta L}. \tag{1.19}$$

Thus, an alternative derivation of this prior is to define the ratio in prior probabilities between link structures with different numbers of links rather than a distribution over L. That is, assuming a prior of the form given in (1.19), for link structures C and C' of the same dimension that contain L and $L + 1$ links respectively

$\pi(C' \mid \theta, n_A, n_B)/\pi(C \mid \theta, n_A, n_B) = e^{-\theta}$. While this property is appealing, in practice the resulting distribution over L can be extremely concentrated, especially when n_A and n_B are large. We discuss this prior further in Chapter 2 but in general do not apply it to fully Bayesian record linkage problems. However, we note that the concentration in the marginal distribution of L could in theory be mitigated by placing an appropriate prior distribution over θ, as Zanella (2019) does with λ.

Given its simplicity and ease of interpretation, we will employ the Beta-bipartite prior defined in (1.15) throughout in our simulations and real-world examples. However, none of the theory we develop is dependent on this choice of prior and we do not expect that any of our results are an artifact of this choice.

1.6.2 Estimating Bayesian Models

The primary limitation 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 discrete parameter (the unobserved link structure) is particularly difficult. Computational considerations have thus far limited the application of Bayesian methods for PRL to small problems, or large problems that can be effectively made small using clean quasi-identifiers to construct an extremely high quality blocking scheme.

With the exception of Sadinle (2017) most implementations of Bayesian PRL under one-to-one matching update C using local Metropolis-Hastings moves within Markov chain Monte Carlo (MCMC) algorithm. 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 (2019), 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 combined with traditional blocking or a cluster based blocking scheme. In these cases the records in both files are partitioned such that links between records can only occur within elements of the partition. 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, 2018). While effective, this approach requires that the blocks be extremely small as the number of possible link structures grows with factorial order (Gutman et al., 2013). For example, there are only 7 possible linkage structures (distinct values of C) for 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. For generative models, which estimate a latent true entity, it is sometimes possible to partition the space in such a way that mixing can be accelerated (see e.g. (Tancredi et al., 2011; Marchant et al., 2019)) but this method does not apply generally to comparison based models.

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 primary bottleneck is

updating the linkage structure C, due to its sheet size. We focus on this bottle neck in Chapter 3, proposing a general method for scaling Bayesian inference to much larger problems.

1.7 Propagation of uncertainty

The final target of inference in PRL is almost never the link structure itself but some quantity which *depends* on the estimated link structure. Therefore, a primary component of any PRL estimation is a characterization of the uncertainty in the link structure. Common examples of quantities for inference include the degree of overlap between the files, regression coefficients based on linked data (Tancredi et al., 2011; Lahiri and Larsen, 2005). In such, scenarios, simply ignoring the uncertainty in the link structure and treating the links as certain will typically result in biased estimates due to the presence of false matches in the set of links considered (Neter et al., 1965).

A naive approach to reducing this bias might involve considering only a subset of the estimated links, such as those that are nearly certain to correspond to true matches. While this approach may limit the bias introduced by false matches it does not provide a characterization of the uncertainty inherent in the PRL estimation. Furthermore, it may be the case that there are differences between the records for which links can be estimated with a high degree of certainty and those for which estimated the links are more uncertain. For example, in historical data it is often much easier to link men across time than women as women changed their surname at marriage and were less likely to list an occupation other than housewife. Thus, an analysis limited to near certain links will tend to exclude a large share of the women. An alternative approach is to model the link structure and the quantity of interest jointly as done by Hof et al. (2017). However, this requires the development of an appropriate model for each PRL limiting its application in practice. A more general method of uncertainty propagation is therefore desirable.

Perhaps the most common setup for analysis with linked data involves fitting a regression model where the response variable is contained in one dataset that is to be linked with another dataset containing the desired covariates. The problem has been studied extensively, with Neter et al. (1965) first noting that errors in the linking process will introduce bias into estimated regression coefficients and suggesting that if an estimate of the false match rate was available that it might be used to remove this bias. Scheuren and Winkler (1993) developed an estimator to reduce the bias introduced by false matches. The estimator considers the top two candidates for each record, relying on estimates of the false match rate. In subsequent work Scheuren and Winkler (1997) proposed an iterative procedure to further reduce bias in estimated regression coefficients. Unbiased estimates of regression coefficients using linked data were developed by Lahiri and Larsen (2005). With various extensions to logistic regression and estimating equations (Chambers, 2009), multiple datasets (Kim and Chambers, 2012), and nested errors (Samart and Chambers, 2014) having been more developed

more recently. Crucially, these approaches all rely on estimates of the false match rate such as those outlined by Belin and Rubin (1995), which typically require labeled data to estimate accurately.

While the majority of existing work appears to have been done in the context of regression with linked data, some additional problems have been considered. Chipperfield et al. (2011) developed an EM algorithm based method for contingency tables and logistic regression and Chipperfield and Chambers (2015) introduced a parametric bootstrap for tabulated data which can incorporate a one-to-one matching constraint. However, to our knowledge a fully general approach for propagating linkage uncertainty has not been developed within a frequentest framework. Although, we note that Goldstein et al. (2012) argues that full PRL is unnecessary and that unbiased parameter estimates can be achieved through multiple imputation. In contrast to the approaches discussed so far, Bayesian methods provide a general method for modeling uncertainty in the linkage structure by estimating a posterior distribution. Because this approach allows posterior samples of the full link structure to be generated, a posterior distribution is also easily generated for any quantity that depends on the link structure. Hence, uncertainty propagation via a posterior distribution over the link structure is generally more broadly applicable than the specific methods discussed in this section.

Chapter 2

Improved Optimization for One-to-one Matching

As introduced in Section 1.5.6 Jaro (1989) first applied a linear sum assignment problem (LSAP) to the task of resolving estimated weights into an estimate of C consistent with one-to-one matching in record linkage problems. Yet little, if any, advances have been made to methods for enforcing the one-to-one matching constraint since the original work by Jaro. The recent R package fastLink (Enamorado et al., 2018) provides a method for enforcing one-to-one matching but the accompanying paper does not provide any details about the procedure, which appears to be based on a greedy algorithm (Enamorado et al., 2019). Chipperfield and Chambers (2015) also employs a greedy algorithm for resolving a link structure consistent with one-to-one matching. Yet, both of these methods seem to have been developed for speed and convenience without any theoretical justification for the resulting estimate.

To address this gap we present a modified LSAP for enforcing one-to-one matching which, produces an estimate of C that both theoretically preferable in that it is more closely aligned with the typical record linkage problem. We then show how this modified LSAP can be transformed into a sparse assignment problem allowing for significant computational gains. We use the computational improvements achieved by inducing this sparsity to develop a new unsupervised record linkage *penalized likelihood* estimator which incorporates one-to-one matching into the estimator. Finally, we demonstrate how our estimator can be used to efficiently perform a sensitivity analysis in an unsupervised setting and yields considerable performance gains over methods which fail to enforce the one-to-one matching constraint throughout the estimation process.

2.1 Modified one-to-one assignment problem

Given a set of estimated weights, Jaro (1989) suggested solving (1.11) by constructing a canonical LSAP, which assumes that each record in A will be matched to some record in B. Since in practice n_A and n_B are almost never equal we first define $\check{w}_{ab}$:

$$\check{w}_{ab} = \begin{cases} w_{ab} & a \leq n_A \text{ and } b \leq n_B \\ w_{-max} & a > n_A \text{ or } b > n_B \end{cases}, \tag{2.1}$$

where w_{-max} is a negative value that is larger (in absolute value) than any of the observed weights. To transform the $n_A \times n_B$ weight matrix into a square assignment problem the approach introduced by Jaro (1989) adds columns (or rows) with values of w_{-max} such that a square weight matrix is constructed. A set of matches is then estimated by solving the following LSAP:

$$\begin{aligned} \check{C}^* &= \max_C \sum_{i=1}^{k} \sum_{j=1}^{k} C_{ij} \check{w}_{ij} \\ &\text{subject to } C_{ij} \in \{0, 1\} \\ &\sum_{j=1}^{k} C_{ij} = 1, \qquad \text{for } i = 1, \ldots, k \\ &\sum_{i=1}^{k} C_{ij} = 1 \qquad \text{for } j = 1, \ldots, k, \end{aligned} \tag{2.2}$$

where $k = \max(n_A, n_B)$. Finally, any matches with weights under a threshold T_μ are dropped, which automatically removes any matches that correspond to augmented entries in C. The solution to this optimization problem is typically found through the use of the Kuhn-Munkres or "Hungarian" algorithm (Kuhn, 1955). In much of the optimization literature assignment problems are formulated as finding a minimal assignment for a cost matrix. Here we will treat W as a *reward* matrix, the negative of a cost matrix, and frame the problem as one of finding a maximal assignment. Since any reward matrix can be transformed into a cost matrix (and any cost matrix into a reward matrix) this imposes no limits on the set of algorithms for solving LSAPs which can be employed. For consistency with other chapters we will use the term "weight matrix" instead of "reward matrix" unless we are directly referencing the optimization literature.

While in its original formulation the Hungarian algorithm required a square problem with an equal number of rows and columns in the cost matrix such as the one defined in (2.2), subsequent work has provided algorithms for solving rectangular or asymmetric problems and such algorithms have since become standard (Bourgeois and Lassalle, 1971; Bijsterbosch and Volgenant, 2010; Bertsekas and Castanon, 1992).

In an asymmetric assignment problem there is no need to add the supplemental weights provided by $\breve{w}$. We will therefore proceed taking the asymmetric version of the optimization problem as defined in (2.3) as our starting point.

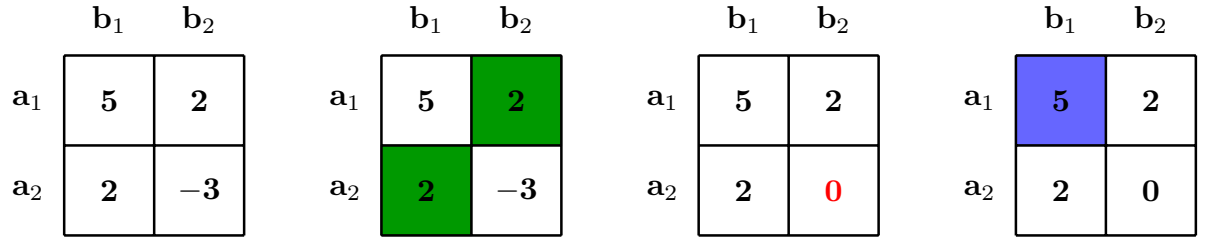
$$\begin{aligned} C^* = \max_C & \sum_{a=1}^{n_A}\sum_{b=1}^{n_B} C_{ab} w_{ab} \\ \text{subject to} \quad & C_{ab} \in \{0,1\} \\ & \sum_{b=1}^{n_B} C_{ab} = 1 \qquad \forall a \in A \\ & \sum_{a=1}^{n_A} C_{ab} \leq 1 \qquad \forall b \in B. \end{aligned} \tag{2.3}$$

For simplicity we will assume, without loss of generality, that $n_A \leq n_B$. With the asymmetric modification the solution to (2.3) requires that all records in A (the smaller of the two files) be matches with some record in B (the larger file). For some applications, such as linking to the US census, this may be a reasonable assumption. But in many applications, such as in a capture-recapture setting, or when files are believed to be incomplete we may expect that at least some records from each file will not be present in the other. In such scenarios it may be more desirable to find a only a partial assignment, leaving some records from A unassigned. In this setting an estimate of C generated by solving (2.3) and then dropping matches with weights below a threshold T_μ may not lead to the estimated matches with the highest total weight.

2.1.1 Thresholded Weights

The reason solving (2.3) may fail to result in an optimal estimate of C after removing matches with weights below the threshold is due to the full assignment constraint. To maximize the objective function under this constraint, high quality matches (large positive weights) must be maximized, while low quality matches (those where the corresponding weight is large and negative) must also be minimized. Yet, because low quality matches will correspond to weights below the threshold, they will be removed by the final step in the Jaro procedure regardless. It is therefore suboptimal to apply a procedure which prefers to link record pairs with weights marginally below the threshold to those significantly below the threshold.

Figure 2.1 provides a simple example of this dynamic. The optimal assignment for the weight matrix shown in Panel (a) has a lower total weight as a result requiring that all records be assigned. Since the solution to (2.3) 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 (shown in Panel (b)) corresponds to the optimal assignment, because of the negative weight on the pair (a_2, b_2). But inspecting the weight matrix shows that the best *overall* matching – allowing some records to remain unassigned – is obtained by linking a_1 and b_1, leaving a_2 and b_2 unmatched. Furthermore, leaving some



(a) estimated weights **(b)** standard solution **(c)** thresholded weights **(d)** modified solution

Figure 2.1: Simple assignment problem with and without thresholded weights. (a) shows an example of estimated weights. (b) highlights the assignment that maximizes the assigned weights for the weights given in (a) if all rows are assigned. (c) Adjusts the weights shown in (a) by thresholding at 0. (d) the maximal assignment solution if the thresholded weights are used and zero weight assignments are then removed. The resulting assignment has a larger total weight than the one given in (b).

records unassigned is consistent with the approach of Jaro (1989), which removes links below a predetermined threshold *after* obtaining a full assignment. By thresholding the weights at zero, as shown in Panel (c), we obtain a modified weight matrix for which finding a maximal full assignment will contain the best overall matching, accounting for the subsequent removal of any assignments with weights below zero as shown in Panel (d). Intuitively, when finding an assignment, we should not differentiate between record pairs with weights below the assignment threshold. Such record pairs will be removed from the final set of matches (classified as nonmatches) regardless of whether their corresponding weights are marginally below the assignment threshold or well below the threshold. The example shown in Figure 2.1 demonstrates the consequences of failing to include this information in the estimation processes. We might therefore wish to incorporate the constraint that we only consider assigning record pairs if their corresponding weight is greater than a threshold T_μ.

The solution to this problem is to incorporate the threshold directly into the LSAP. The procedure will remove any links made where the corresponding weight is below a threshold T_μ. We therefore propose thresholding the weights used in the optimization procedure so that all weights below the threshold are assigned a value of zero. We then carry out a modified version of the procedure described in Jaro (1989) utilizing the thresholded weights. In this formulation, removing all assigned record pairs for which the weight is below a threshold is equivalent to removing assigned record pairs with thresholded weight of zero. We define the *thresholded weights* as:

$$\underline{w}_{ab} = \begin{cases} w_{ab} & w_{ab} \geq T_\mu \\ 0 & w_{ab} < T_\mu \end{cases} . \tag{2.4}$$

Substituting the thresholded weights for the standard weights into the LSAP defined in (2.3) results in the modified LSAP:

$$
\begin{aligned}
\underline{C}^* = \max_{C} & \sum_{a,b \in A \times B} C_{ab} \underline{w}_{ab} \\
\text{subject to} \quad & C_{ab} \in \{0,1\} \\
& \sum_{b \in B} C_{ab} = 1 \qquad \forall a \in A \\
& \sum_{a \in A} C_{ab} \leq 1 \qquad \forall b \in B.
\end{aligned}
\tag{2.5}
$$

After solving (2.5) we remove all assignments where the thresholded weight $\underline{w}_{ab}$ is zero. To evaluate C^* and $\underline{C}^*$ we sum the original weights w_{ab} for all record pairs which remain assigned after removing the entries of C for which $w_{ab} < T_\mu$. Thus, we define the post-removal objective functions:

$$
\begin{aligned}
f^*_{T_\mu} &= \sum_{a,b \in A \times B} C^*_{ab} w_{ab} \mathbb{1}(w_{ab} \geq T_\mu) \quad = \sum_{a,b \in A \times B} C^*_{ab} \underline{w}_{ab} \\
\underline{f}^*_{T_\mu} &= \sum_{a,b \in A \times B} \underline{C}^*_{ab} w_{ab} \mathbb{1}(w_{ab} \geq T_\mu) \quad = \sum_{a,b \in A \times B} \underline{C}^*_{ab} \underline{w}_{ab}
\end{aligned}
\tag{2.6}
$$

Where C^* and $\underline{C}^*$ are obtained by solving (2.3) and (2.5) respectively.

Lemma 2.0.1 (Improved Objected Value from Thresholded Problem)**.** *For any real-valued weight matrix W and any threshold T_μ $f^*_{T_\mu} \leq \underline{f}^*_{T_\mu}$.*

Proof for Lemma 2.0.1.

$$
\begin{aligned}
f^*_\theta &= \sum_{a,b \in A \times B} C^*_{ab} w_{ab} \mathbb{1}(w_{ab} \geq T_\mu) \\
&= \sum_{a,b \in A \times B} C^*_{ab} \underline{w}_{ab} \\
&\leq \sum_{a,b \in A \times B} \underline{C}^*_{ab} \underline{w}_{ab} \\
&= \underline{f}^*_\theta
\end{aligned}
$$

□

Lemma 2.0.1 makes it clear that within the Jaro framework that $\underline{C}^*$ (with zero thresholded weight assignments removed) is generally a more desirable estimate of C than C^*. The key insight is that because the framework requires the final estimate of C to correspond to a full assignment, this requirement should not be enforced within the estimation processes. This suggests that we might consider solving a different optimization problem to generate our estimate of C, namely one which does not require that all records be assigned. Such an approach would correspond to solving the following LSAP:

$$
\begin{aligned}
C^*_{partial} = \max_{C} & \sum_{a=1}^{n_A}\sum_{b=1}^{n_B} C_{ab} w_{ab} \\
\text{subject to} \quad & C_{ab} \in \{0,1\} \\
& \sum_{b=1}^{n_B} C_{ab} \leq 1 \qquad \forall a \in A \\
& \sum_{a=1}^{n_A} C_{ab} \leq 1 \qquad \forall b \in B.
\end{aligned}
\tag{2.7}
$$

The problem formulated in (2.7) has the drawback that, as defined, it is not easily solved with standard algorithms for assignment problems, which have been developed for finding full assignments. Define the objective of (2.7):

$$
f^*_{partial} = \sum_{a,b \in A \times B} C^*_{partial,ab} w_{ab} \tag{2.8}
$$

We show in Lemma 2.0.2 that $f^*_{partial} = \underline{f}^*_0$. Where $\underline{f}^*_0$ is the objective achieved by (2.3) with $T_\mu = 0$. An intuition for Lemma 2.0.2 can be gained by observing that, absent the requirement to link all records (which is removed in (2.7)) the objective in (2.3) is increased only by linking record pairs with positive weights. Recalling the definition of the weights (1.5) we see that these weights correspond to record pairs for which $P(\gamma_{ab} \mid C_{ab} = 1) \geq P(\gamma_{ab} \mid C_{ab} = 0)$. Given that the U component, which contains the non-matching record pairs, is typically far larger, it makes sense to restrict our consideration to record pairs that are more likely in the M component (for these record pairs it may still be the case that $P(C_{ab} = 1 \mid \gamma) < P(C_{ab} = 0 \mid \gamma)$).

In essence, setting $T_\mu = 0$ and solving (2.3) removes the full assignment constraint. As we will discuss in Section 2.2 setting many of the weights within the optimization procedure to zero yields computational benefits in addition to the improved objective function. In the next Section we introduce an alternative modification which may be made to the weights which achieves many of the same objectives.

Lemma 2.0.2 (Objective Equivalence of Zero-Thresholding and Partial Assignment)**.** *For any real-valued weight matrix* W $f^*_{partial} = \underline{f}^*_0$.

Proof for Lemma 2.0.2. Let W be a set of real-valued weights, $\underline{C}^*_0$ be a solution to (2.3) using W with link threshold $T_\mu = 0$, and $C^*_{partial}$ a solution to (2.7) also using W.

Suppose that $f^*_{partial} > \underline{f}^*_0$. Then construct $C^\dagger$ from $C^*_{partial}$ by fixing all links in $C^*_{partial}$ and then sequentially assigning any unassigned rows to the first unassigned column. Since by assumption W will have at least as many rows as columns this will always result in a full assignment. Furthermore $\underline{w}_{ab} \geq 0$ for all

$a, b \in A \times B$. Hence,

$$\sum_{a,b \in A \times B} C^{\dagger} \underline{w}_{ab} \geq f^*_{partial} > \underline{f}^*_0 = \sum_{a,b \in A \times B} \underline{C}^*_{ab} \underline{w}_{ab}.$$

Then $\underline{C}^*$ is not a solution to (2.3) ($\Rightarrow\Leftarrow$).

Suppose that $f^*_{partial} < \underline{f}^*_0$. Construct $C^{\ddagger}$ from $\underline{C}^*$ by setting $C^{\ddagger}_{ab} = 0$ for all links from $\underline{C}^*$ where $\underline{w}_{ab} = 0$:

$$C^{\ddagger}_{ab} = \begin{cases} \underline{C}^*_{ab} & \text{if } w_{ab} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Because $T_\mu = 0$, then $w_{ab} \geq 0$ for all a, b where $C^{\ddagger}_{ab} = 1$. Thus,

$$\sum_{a,b \in A \times B} C^{\ddagger} w_{ab} = \underline{f}^*_0 > f^*_{partial}$$

where the equality holds because only links for which $\underline{w}_{ab} = 0$ were removed from $\underline{C}^*$ to construct $C^{\ddagger}$. Then $C^*_{partial}$ is not a solution to (2.7) ($\Rightarrow\Leftarrow$).

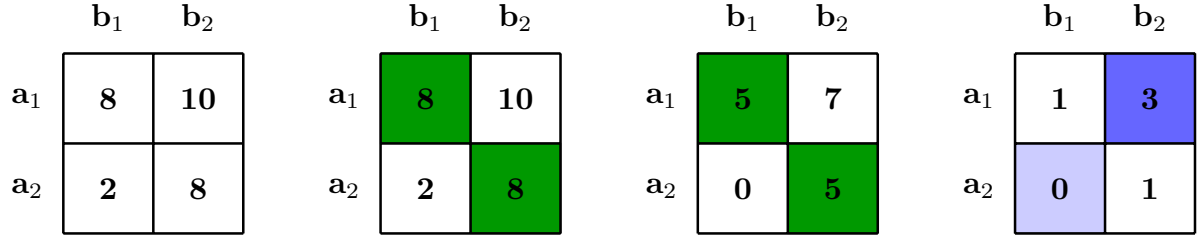
Therefore $f^*_{partial} = \underline{f}^*_0$ □

2.1.2 Penalized Weights

An alternative adjustment to the hard thresholding used to constructed the weights defined in (2.4) uses a penalty or soft-thresholding. The penalty θ is subtracted from all weight values and any weights for which this would yield a negative value are set to zero. This transformation is defined in (2.9) and we will refer to these values as *penalized weights*.

$$\tilde{w}_{ab} = \begin{cases} w_{ab} - \theta & w_{ab} > \theta \\ 0 & w_{ab} \leq \theta \end{cases} \tag{2.9}$$

The transformations defined by (2.9) and (2.4) are equivalent when $T_\mu = \theta = 0$. Thus, Lemma 2.0.2 applies to the penalized weights for $\theta = 0$ as well as the thresholded weights with $T_\mu = 0$. This behavior suggests that setting $\theta = T_\mu$ might generally lead to similar estimates of C. However, despite the equivalence between the thresholded and penalized weights at zero, these transformation can lead to significantly different estimates of C for larger values of the penalty (or threshold). An estimate of C, $\widetilde{C}^*$ can be acquired by solving the LSAP given by (2.10).



(a) Weights **(b)** Thresholded weight $T_\mu \in [0, 8)$ **(c)** Penalized weights $\theta = 3$ **(d)** Penalized weights $\theta = 7$

Figure 2.2: Example assignment problem where thresholded and penalized weights yield different estimates. (a) shows example weights, (b) the optimal assignment using thresholded weights for any threshold $T_\mu \in [0, 8)$. (c) and (d) show the solution using penalized weights when $\theta = 3$ and $\theta = 7$ respectively. The solution in (d) contains fewer links but favors those with large weights.

$$\begin{aligned} \widetilde{C}^* = \max_C & \sum_{a,b \in A \times B} C_{ab} \tilde{w}_{ab} \\ \text{subject to} \quad & C_{ab} \in \{0, 1\} \\ & \sum_{b \in B} C_{ab} = 1 \qquad \forall a \in A \\ & \sum_{a \in A} C_{ab} \leq 1 \qquad \forall b \in B. \end{aligned} \tag{2.10}$$

We consider an example set of weights in Figure 2.2. Panel (a) shows the original weights and (b) shows the solution to the assignment problem that results from using thresholded weights for any $T_\mu \in [0, 8)$ *. Panels (c) and (d) show penalized weights with $\theta = 3$ and $\theta = 7$ respectively as well as the solutions to the corresponding assignment problem. The solution show in (c) would be found for any $\theta \in (-\infty, 6)$† and that shown in (d) for any $\theta \in (6, 10)$. In these example setting identical threshold and penalty values does not always yield the same assignment, notably if $T_\mu = \theta = 7$ then thresholding will yield the solution in (b) and penalization will yield the solution in (d).

This difference in behavior can be understood as typical in the context of regularization. We can re-write the objective in (2.10) as follows:

$$\sum_{a,b \in A \times B} C_{ab} \tilde{w}_{ab} = \sum_{a,b \in A \times B} C_{ab} w_{ab} \mathbb{1}(w_{ab} > \theta) - \theta \sum_{a,b \in A \times B} C_{ab} \mathbb{1}(w_{ab} > \theta) = \sum_{a,b \in A \times B} C_{ab} w_{ab} \mathbb{1}(w_{ab} > \theta) - \theta L, \tag{2.11}$$

where $L = \sum_{a,b \in A \times B} C_{ab} \mathbb{1}(w_{ab} > \theta)$, the number of assigned record pairs with nonzero penalized weights. The formulation in (2.11) makes it clear that, allowing for some records to remain unlinked, the penalty

*In fact this is the optimal assignment for any $T_\mu \in [-\infty, 8)$ but we limit our investigation to the case where $T_\mu \geq 0$. As selecting a threshold $T_\mu < 0$ may result in a change in the ranking of the thresholded weights relative to the original weight.

†We allow the case where θ is less than it does disrupt the ranking of weights in the same manner

serves to regularize the estimate of C. Hence, optimizing over the penalized weights will tend to yield fewer links with larger weights than would result from optimizing over the thresholded weights. In fact (2.10) finds, conditional on the weights, a maximum a posteriori (MAP) estimate of C under the prior introduced by Green and Mardia (2006) and outlined in Section 1.6.1. A similar insight can be gained by observing that the penalized weights optimize for total linked weight above the penalty while the thresholded weights optimize the total weight, conditional on all linked record pairs having a weight above the threshold.

The penalized weights therefore place more emphasis on the margin by which a weight exceeds the penalty. Within a one-to-one matching context the penalized weights may be more desirable as in many contexts we would typically prefer extremely good matches over simply good matches, which might correspond to individuals such as household members which are merely similar. However, regardless of the choice between penalized and thresholded weights both transformations offer the possibility of significantly improving the computation tractability of the optimization by inducing sparsity into the assignment problem as we discuss in the next section.

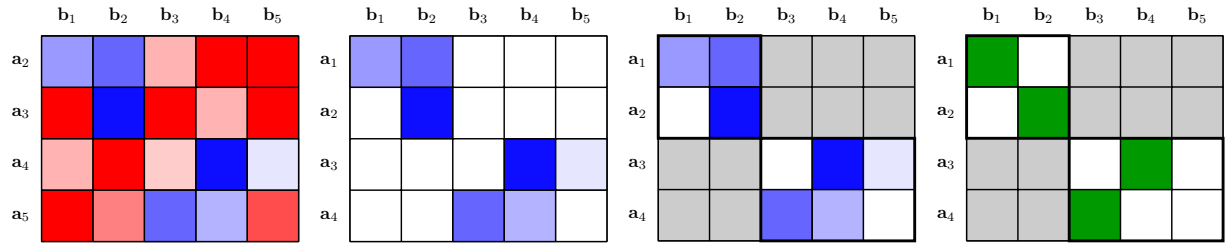
2.2 Solving Sparse Assignment Problems

Since the number of non-matching record pairs is typically much greater than the number of matching record pairs, the value of the vast majority of the penalized $\tilde{w}_{ab}$ will typically be zero. This means that there will be many values of C which correspond to a maximal full assignment over these weights. While this level of redundancy means that an optimal solution may be easier to find on average it does not reduce the size of the solution space, which will remain large, and thus computationally costly. We can significantly reduce the computational cost of solving (2.10) by transforming $\widetilde{W}$ into a sparse matrix. While in this section we will frame our discussion of sparsity in terms of penalized weights as they are our preferred measure, the computational advantages of induced sparsity we discuss apply equally well to the thresholded weights $\underline{w}_{ab}$ introduced in Section 2.1.1.

2.2.1 Connected Component Separation

There are a variety of ways that sparsity can be leveraged to increase the speed at which the assignment problem can be solved. If a traditional blocking scheme has been applied to the record linkage problem then the assignment problem need be solved only *within* the blocks, generally resulting in significant computational gains given the $O(n^3)$ worst case complexity of traditional algorithms for solving dense LSAPs. Indeed this approach was applied by Jaro (1989).

However, it is possible to apply this intuition in a more sophisticated manner, as we show in Figure 2.3. Consider the link structure as a bipartite graph, with the non-zero entries of $\widetilde{W}$ detonating an edge matrix E.



(a) Weights **(b)** Thresholded Weights **(c)** Connected Components **(d)** Final Assignment

Figure 2.3: Example of how weight sparsity can reduce problem complexity. (a) shows an example of estimated weights and (b) shows a penalized version. In (c) penalized weights are separated into connected components with each row and column (records from A and B respectively) appearing in at most one component. Finally, (d) shows the solution to the assignment problem, which can be computed separately for each component.

Then we need solve the LSAP only within the connected components of this graph, as the assignment within one component does not affect the set of assignments which can be made within a separate component. In Figure 2.3 the blue entries in panel (b) correspond to the edges in E. The blocks outlined in panel (c) show the separation which occurs between the connected components. It also need not be the case that all records are assigned to a component. For example record b_5 is included in none of the components and therefore is not considered for linking. The assignments within the components, show in panel (d) are then significantly easier to make than if the entire link structure is considered.

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 penalizing) (Tarjan, 1972). 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. In the case where $k << \min(n_A, n_B)$ we can obtain dramatic reductions in computational complexity by partitioning the original problem. This leads to the procedure outlined in Algorithm 1.

Algorithm 1 Connected-Component Based Assignment Problem

Input: Thresholded weight matrix $\widetilde{W}$ (from Eq (2.9))
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.

If either traditional blocking or a cluster based blocking scheme has been applied to the problem then the connected components are guaranteed to be no larger than the blocks or clusters. However, with a blocking

scheme such as indexing by disjunctions it is possible for the largest component of the graph to contain all or nearly all of the records. In such situations applying Algorithm 1 will yield little in terms of the overall complexity of solving the assignment problem. The size of the largest component will in turn depend on θ (with larger penalty values resulting in a smaller largest component).

An alternative approach is to simply apply an algorithm for solving assignment problems which is designed to handle a sparse reward matrix. For example, a shortest augmenting path algorithm of this type was introduced by Jonker and Volgenant (1987). Hong et al. (2016) describes a version of the Hungarian algorithm for sparse problems with complexity $O(nE)$ where n is the smaller of the number of rows and the number of columns in the reward matrix and E is the number of entries. Practical performance is often much better than these worst-case complexity results might suggest. 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).

Algorithms for solving sparse LSAPs generally require that a feasible solution exists but the transform from $\widetilde{W}$ to $\widetilde{W}'$ guarantees that a trivial feasible solution exists: $C_{a,a+n_B} = 1$ a $= 1, \ldots, n_A$. If desired these can be applied jointly with the graph clustering procedure outlined in Algorithm 1. A class of algorithms which appear particularly well suited to solving sparse assignment problems in a record linkage context are auction algorithms.

2.2.2 Auction algorithms

The auction algorithm was inspired by price auctions in which people (rows) bid for objects (columns). The value person a places on object b is given by the reward w_{ab} that would be achieved by assigning object b to person a. As in other algorithms for solving assignment problems dual variables are updated and assignments are made in cases where $w_{ab} = u_a + v_b$ where u_a is the profit person a expects and v_b is the price object b commands. The Hungarian algorithm and other shortest path algorithms maintain a similar set of dual variables, although these are not typically given the same price interpretation.

Both types of algorithms work by successively adding assignments while adjusting the dual variables until all rows have been assigned. A key difference however is that auction algorithms typically find only an *approximate* solution to the assignment problem whereas shortest path algorithms will find an exact solution. Auction algorithms find only an approximate solution enforcing a constraint on the dual variables known as ϵ-complementary slackness. Under this condition the inequality $w_{ab} - \epsilon \leq u_a + v_b$ is maintained for all a, b throughout the execution of the algorithm. In shortest path algorithms the analogs constraint requires $w_{ab} \leq u_a + v_b$ (i.e. $\epsilon = 0$). One interpretation of ϵ-complementary slackness is that the solution found is optimal for some reward matrix with rewards w'_{ab} such that $|w_{ab} - w'_{ab}| \leq \epsilon$ for all a, b. The sum of the assigned rewards is thus at most ϵn_A less than the true optimum since each assigned reward is below

the optimal reward by at most ϵ. Increasingly accurate solutions can be found through a process known as ϵ-scaling, where a problem is solved repeatedly with successively smaller ϵ's using the previous solution as a starting point. This process will eventually yield an optimal solution[‡].

As with the Hungarian algorithm, early auction algorithms worked only for rectangular problems. They also frequently performed poorly on a subclass of assignment problems in which many entries in the reward matrix contained similar values. In such problems the auction algorithms were prone to engaging in "bidding wars", in which assignments are updated frequently yielding only small changes in the dual variables. However, modern implementations of the algorithm addressed both the bidding war issue and provided a version appropriate for asymmetric problems (Bertsekas and Castanon, 1989, 1992; Bertsekas et al., 1993). For a more comprehensive overview of auction algorithms see (Bertsekas, 1998, Chapter 7).

Auction algorithms have proven particularly successful when applied to sparse problems. One issue that can arise in sparse assignment problems, although not in the problem we solve in (2.10), is that of infeasibility, when no allowed full assignment exists. Auction algorithms can be designed in a manner to detect this (infeasibility will cause the dual variables to diverge) (Bertsekas, 1992), a desirable property. Finally, the computational complexity achieved by auction algorithms in sparse problems is as good or better than that of competing algorithms.

Auction algorithms have a proven worst-case complexity of $O(nE\log(nW_{\max}))$ where n is the number of rows of the reward matrix, E is the number of non-missing entries in the reward matrix, and $W_{\max}$ the is largest (in absolute value) reward(Bertsekas and Eckstein, 1988; Bertsekas and Tsitsiklis, 1989; Bertsekas, 1998). This bound can be further improved by combining an auction algorithm with a primal-dual method to achieve $O(E\sqrt{n}\log(nW_{\max}))$ complexity, the best known worst-case complexity for an assignment problem (Bertsekas, 1998, Chapter 7.1.4). However, this complexity analysis assumes that all rewards are integer valued and therefore selecting $\epsilon < W_{\max}/n$ guarantees an optimal assignment. For non-integer rewards the smallest margin between reward values governs how small ϵ must be to guarantee an optimum. This may result in a somewhat worse complex but the log factor means that in practice this is not too problematic. Importantly, this covers only worse case complexity and there is reason to believe that the average case may be significantly faster, perhaps as fast as $O(E\log(n)\log(nW_{\max}))$(Bertsekas, 1998, Chapter 7.1.4). Other work supports this hypothesis that it may be possible to solve sparse assignment problems in near linear time (on average) with respect to the number of edges (Orlin and Lee, 1993; Amini, 1994).

Intuition for why average complexity may be significantly better than the worse case can be gained by considering the case where the graph is partitioned into several distinct components, as considered by Algorithm 1. In this scenario the complexity will be, at worst, a product of the number of components and the complexity of solving the problem in the largest component. Because auction algorithms only allow

[‡]Assuming that values of w_{ab} and $w_{a'b'}$ which are not *exactly* equal differ by at least δ a choice of $\epsilon < \delta/n_A$ will guarantee that the optimal solution is found.

bids for feasible assignments they will leverage problem structure of this type automatically. However, even if clusters are only "weakly" separated, that is allowing a few record pairs with membership of multiple clusters, bids will still mainly fall within clusters. Thus, we might hope for performance close to that of the case where the clusters are fully separated.

Several features of both auction algorithms and record linkage problems suggest that they may perform particularly well together. First, auction algorithms include natural extensions to both parallel and distributed settings (Bertsekas and Tsitsiklis, 1989; Amini, 1994). Thus, they can be employed practically on potentially very large problems, an increasingly common setting for record linkage. There is also reason to believe that the expected distribution of record linkage weights (rewards for the auction algorithm) may be especially favorable in record linkage problems. For example, if a record linkage problem displays the microclustering property (Betancourt et al., 2016) then we might expect relatively few large weights occurring almost entirely within the clusters, similar to the cluster based scenario discussed above. Even absent microclustering it was observed by Newcombe and Kennedy (1962) that the estimated weights are usually split into many very low weights (near certain non-links), some large weights (near certain links) with comparatively few intermediate values, which are harder to assign. We might therefore expect our thresholded weights (2.9) to produce an especially sparse reward matrix. Finally, use of ϵ-feasibility within algorithm iterations and ϵ-scaling can work well with the statistical estimation processes as we describe in more detail in Section 2.3.

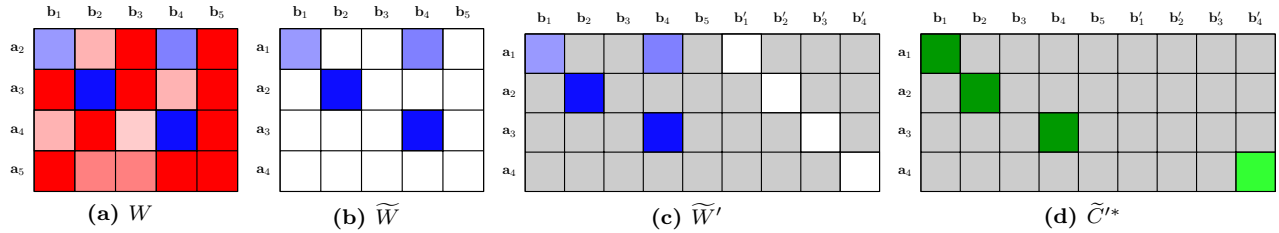
2.2.3 Ensuring a Feasible Sparse Problem

The computational advantages of applying a sparse transformation to W are clear. However, many algorithms will fail if a feasible solution does not exist, that if after inducing sparsity some records have no possible assignments. To address this potential complication we define the following transformation:

$$\tilde{w}'_{ab} = \begin{cases} w_{ab} - \theta & w_{ab} - \theta > 0,\ ab \in A \times B \\ 0 & a \leq n_A,\ b = n_B + a \\ -\infty & \text{otherwise} \end{cases}, \tag{2.12}$$

Alternatively, $\widetilde{W}' = \left[\widetilde{W}_{sp} \mid \text{diag}(\mathbf{0})_{n_A}\right]$ where $\widetilde{W}_{sp}$ is a sparse (setting all values of zero to $-\infty$) of $\widetilde{W}$ and $\text{diag}(\mathbf{0})_{n_A}$ is a $n_A \times n_A$ diagonal matrix with the values of zero on the diagonal and $-\infty$ for all off-diagonal entries. The diagonal matrix is added to ensure that a feasible assignment always exists. While developed independently, the transform to $\widetilde{W}'$ appears similar to the transformation used in the k-cardinality LSAP outlined by Bijsterbosch and Volgenant (2010).

An example of this transformation is shown in Figure 2.4. Panel (a) shows example weights, with those in blue having values larger than the selected penalty θ and those in red having values smaller than the penalty.



(a) W **(b)** $\widetilde{W}$ **(c)** $\widetilde{W}'$ **(d)** $\widetilde{C}'^{*}$

Figure 2.4: Example transformation of weights and solution equivalence. (a) shows an example of estimated weights. (b) shows the effect of transforming W to $\widetilde{W}$. (c) demonstrates the transformation from $\widetilde{W}$ to $\widetilde{W}'$ adding sparsity and dummy observations. Finally, (d) shows $\widetilde{C}'^{*}$ the optimal solution using the weights $\widetilde{W}'$ with the dark green entries denoting assignments which will be kept and the light green denoting zero-reward assignments which make the assignment feasible but do not affect the objective value and will therefore be removed.

In Panel (b) we show the penalized weights, transforming W to $\widetilde{W}$ with white entries corresponding to cases where $\tilde{w}_{ab} = 0$. Sparsity is induced in Panel (c) demonstrating the transformation from $\widetilde{W}$ to $\widetilde{W}'$. All white entries from (b) are coded as grey indicating that they have been removed (assigned a weight of $-\infty$) and are not available for linking. To ensure the existence of a full assignment supplemental columns representing the diagonal matrix are added with entries $a, n_B + a$ (labeled a_i, b'_i) available for linking. Finally, (d) shows $\widetilde{C}'^{*}$ the optimal solution to both $\widetilde{W}$ and $\widetilde{W}'$. The dark green entries in (d) denote assignments for which $\tilde{w}_{ab} > 0$ which will be retained and the light green entries are assignments for which $\tilde{w}_{ab} = 0$ that will be removed. The link in cell a_4, b_4 will be present if $\widetilde{C}^{*}$ is estimated based on $\widetilde{W}$ (Panel (b)) and the entry a_4, b'_4 corresponding to a value of $\widetilde{C}'^{*}$ estimated based on $\widetilde{W}'$ (Panel (c)). Since $\tilde{w}_{a_4 b_4} = \tilde{w}'_{a_4 b'_4} = 0$ they do not affect the objective value. The solution using the weights $\widetilde{W}'$ displayed in (d) can be found by solving the LSAP:

$$\begin{aligned} \widetilde{C}'^{*} = \max_{C} & \sum_{a,b \in A \times B} C_{ab} \tilde{w}'_{ab} \\ \text{subject to} \quad & C_{ab} \in \{0,1\} \\ & \sum_{b \in B} C_{ab} = 1 \qquad \forall a \in A \\ & \sum_{a \in A} C_{ab} \leq 1 \qquad \forall b \in B. \end{aligned} \tag{2.13}$$

Adopting the formulation in (2.13) has the advantage 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 penalizing, the weight matrix will typically be very sparse. Indeed, depending

on the degree of overlap between the two files there may be entire rows and columns of zeros (as in row a_4 of Figure 2.4) – effectively reducing n and yielding an easier optimization problem.

Auction algorithms in particular can be effective for solving such sparse problems, as described in the previous section. Thus, there is reason to believe that employing penalized weights to induce sparsity will make it possible to solve (2.13) sufficiently quickly that it can be solved many times within an estimation procedure as opposed to a single time, as done by Jaro. In the next section we introduce an estimator which uses this approach to enforce one-to-one matching throughout the estimation procedure resulting in considerable performance gains.

2.3 Penalized Likelihood Estimator

The standard approach for producing $\widehat{C}$, a point estimate of C, (summarized in Section 1.5.6) has several drawbacks. The most significant of which is that the one-to-one matching constraint is not incorporated into the (typically EM-based) estimates of the $m-$ and $u-$ parameters. In this section we outline a new *penalized likelihood* estimator that significantly improves over the standard approach by incorporating all three steps of the standard approach into a single-stage estimation procedure, simultaneously maximizing a joint likelihood in C, m, and u while penalizing the total number of matches.

2.3.1 Algorithm

The penalized likelihood takes the following form, similar to that of the objective for the partial assignment problem defined in (2.7) but with penalized weights:

$$\begin{aligned} l(C, m, u \mid \Gamma) &= \sum_{ab} C_{ab} w_{ab} + \sum_{ab} \log(u(\gamma_{ab})) - \theta \sum_{ab} C_{ab} \\ &= \sum_{ab} C_{ab}(w_{ab} - \theta) + \sum_{ab} \log(u(\gamma_{ab})) \\ &= \sum_{ab} C_{ab} \tilde{w}_{ab} + \sum_{ab} \log(u(\gamma_{ab})). \end{aligned} \tag{2.14}$$

The last term in (2.14) is the penalty and the leading terms are the same log-likelihood corresponding to the standard two-component mixture model 1.13. The form of the penalized likelihood in (2.14) 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 prior for C introduced in Green and Mardia (2006) (defined in (1.19)); the penalized likelihood estimate corresponds to a MAP estimate under the corresponding Bayesian model.

Finding a local mode of (2.14) is straightforward via alternating maximization steps, which are iterated until the change in (2.14) is negligible:

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

$$\begin{aligned} \max_{C} & \sum_{a,b \in A \times B} C_{ab} \tilde{w}_{ab} \\ \text{subject to} \quad & C_{ab} \in \{0,1\} \\ & C_{ab} = 0 \text{ if } \tilde{w}_{ab} = 0 \\ & \sum_{b \in B} C_{ab} \leq 1 \qquad \forall a \in A \\ & \sum_{a \in A} C_{ab} \leq 1 \qquad \forall b \in B. \end{aligned} \tag{2.15}$$

   As discussed in the previous section we recommend solving (2.15) by first solving (2.13) with an auction algorithm and then removing entries of $\widetilde{C}'^{*}_{ab} = 1$ for which $\tilde{w}'_{ab} = 0$.

2. Maximize m and u probabilities, given a value of C. These updates are available in closed form under the conditional independence model (Equation 1.9):

$$m_{jh} = \frac{n_{mjh} + \sum_{ab} C_{ab} \mathbb{1}(\gamma^j_{ab} = h)}{\sum_h n_{mjh} + \sum_{ab} C_{ab}} \tag{2.16}$$

$$u_{jh} = \frac{n_{ujh} + \sum_{ab} (1 - C_{ab}) \mathbb{1}(\gamma^j_{ab} = h)}{\sum_h n_{ujh} + \sum_{ab} (1 - C_{ab})}. \tag{2.17}$$

   where the n's are optional pseudocounts used to regularize the estimates. (These terms correspond to an additional penalty, omitted from (2.14) and(2.15) 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). In fact they correspond to MAP estimate (conditional on $\widehat{C}$) under a Dirichlet prior for m_j and m_j

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 $a \in A$ has two exact matches $b, b' \in B$, then the penalized likelihood function will have at least two modes with the highest possible value. However, the values of the $m-$ and $u-$parameters 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 $u-$parameters.

While the simultaneous maximization of the C, m, and $u-$parameters achieved by the penalized likelihood estimator is conceptually appealing it is not at first clear that the procedure is computationally feasible given that it may require solving many LSAPs. We have found that in practice the solutions are often computationally tractable, particularly when auction algorithms (introduced in Section 2.2.2) are employed to solve the LSAPs. First, by thresholding or penalizing the weights many record linkage problems can be transformed to display a high level of sparsity meaning they are relatively easy to solve. But perhaps more importantly the use of ϵ-scaling in auction algorithms means that the solution from a previous maximization provides an extremely efficient starting value for the next LSAP. Exactly how efficient (corresponding to what value of ϵ the procedure should be initialized at) depends on how much the weights have changed (due to updates in the m and $u-$parameters) from the previous iteration. In many cases small updates to the m and $u-$parameters mean the next LSAP can be solved extremely quickly. The $u-$parameters in particular are typically very stable and closely match the overall empirical similarity distribution. We are not aware of this feature of auction algorithms being used to solved a sequence of related assignment problems having been exploited previously in the literature.

2.3.2 Italian Census Example (Tancredi et al., 2011)

We consider a small scale example dataset from the existing literature to illustrate the performance of the penalized likelihood estimator. The data in this example was published by Tancredi et al. (2011) and comes from a small geographic area in Italy; 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.

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 a conditional independence model for $m-$ and $u-$probabilities as in (1.9). Each vector of conditional probabilities is assigned a Dirichlet prior distribution. We assume that $m_j \sim \text{Dir}(1.9, 1.1)$ and $u_j \sim \text{Dir}(1.1, 1.9)$ 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 first estimate links between the two files using first a standard EM algorithm to estimate the matching parameters and corresponding weights, and then resolve the link structure to one-to-one matching setting the link threshold $T_\mu = 0.0$. We next estimate the link structure using our penalized likelihood estimator for several different values of the penalty θ. A summary of the record comparisons, estimated weights, and estimated links are shown in Table 2.1. As expected, larger values of θ result in the penalized likelihood

Observed Data				EM Algorithm		$\theta \in (0.0, 0.24)$		$\theta \in (0.24, 2.23)$		$\theta \in (2.23, 3.04)$		$\theta \in (3.04, 6.21)$	
Last	Sex	Edu	Count	Weights	Matches	Weights	Matches	Weights	Matches	Weights	Matches	Weights	Matches
1	1	1	25	5.27	24	6.09	24	6.16	24	6.20	24	6.21	24
1	0	1	8	3.77	1	2.72	1	2.89	1	3.04	1	0.66	0
1	1	0	13	-0.94	0	2.04	3	2.23	3	-1.17	0	-1.11	0
0	1	1	126	2.68	4	0.24	3	-3.13	0	-2.91	0	-2.83	0
1	0	0	21	-2.45	0	-1.34	0	-1.03	0	-4.32	0	-6.66	0
0	0	1	78	1.18	0	-3.14	0	-6.39	0	-6.06	0	-8.38	0
0	1	0	601	-3.53	0	-3.81	0	-7.05	0	-10.28	0	-10.16	0
0	0	0	658	-5.04	0	-7.19	0	-10.32	0	-13.43	0	-15.71	0

Table 2.1: Comparison vectors observed in data, weights estimated using an EM algorithm and using our penalized likelihood estimator for a range of θ values. Comparisons vectors for which the EM estimate of the weights differs substantially from the penalized likelihood based estimate are highlighted in grey.

identifying fewer record pairs as matches. Setting T_μ to a larger value and relying on an EM-based estimate of the weight would similarly cause the to identify fewer matches but would not result in different estimates of the weights, as done by the penalized likelihood estimator. A comparison of the estimated weights yields more insight as to where the estimates differ. Rows of Table 2.1 shaded grey highlighting specific comparisons for which the penalized likelihood estimator and the EM algorithm estimate significantly different weights.

For the first grey row, corresponding to a comparison between records which agree on last name and sex but not education level, this difference is determined by the value of θ. For $\theta < 2.23$ the weights estimated using the penalized likelihood estimator are significantly larger, and the estimated link structure several of these record pairs. While for $\theta > 2.23$ the estimated weights using the penalized likelihood estimate more or less agree with those estimated by the EM algorithm. This might be taken to imply that the weights estimated by the EM algorithm are similar to those estimated by the penalized likelihood estimator for larger values of θ. However, we see that this is not the case by examining the estimated weights for the additional comparison vectors. The remaining two highlighted rows both correspond to record pairs which agree on education level and disagree on last name, the first also agrees on sex while the second comparison vector disagrees on the sex comparison. For these record pairs the EM-based weights assign a positive value, while the penalized likelihood estimates generally place a substantial negative weight (for $\theta = 0$ the penalized likelihood estimator places a small positive weight on the record pairs which also agree on sex). Overall we see that for no value of θ does the penalized likelihood estimator assign a positive weight to any of the comparison vectors (rows in the table) which show agreement on only one of the three fields. For larger values of θ even some of the comparison vectors which correspond to agreement on two are assigned a negative weight. In contrast the EM-based weights are positive, indicating that it is more likely to have been generated by the M component than the U component, for every comparison vector which shows agreement on the education level field. This suggests that the EM-based model views the education field as extremely informative in determining the matching status of record pairs. However, even using the EM-based weights, none of the record pairs which match only on education level are actually classified as links once the one-to-one matching

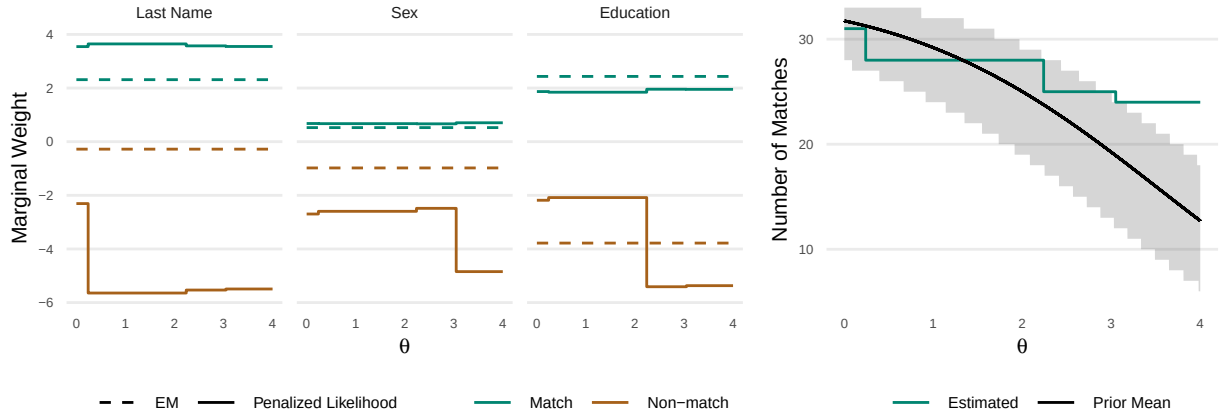


Figure 2.5: Effect of θ on the estimated weights and the number of matches. The left panel shows the marginal estimated weight for each field. In the right panel the number of estimated links is shown as a function of θ in green and the expected number of links for the corresponding prior is shown in black with a 99% interval shown in grey.

constraint is enforced. We might therefore suspect that a failure to enforce one-to-one matching results in a worse estimate of the weight (i.e. that we should prefer the penalized likelihood estimator).

Under a conditional independence model the overall weight assigned to a comparisons vector is simply the sum of the weights for each field. Therefore, it is straightforward to examine the marginal effect of each field on the total estimated weight. We plot these marginal weights in the left panel of Figure 2.5, with the solid lines showing the estimate for the penalized likelihood estimator, which vary with θ, and the dashed lines the EM-based estimates. Because the marginal weights are simply the log-likelihood ratio between the M and U components for the corresponding field, a positive weight indicates that a comparison is estimated to occur with greater probability within the M component than the U component. Similarly, a weight near zero indicates that the comparison vector occurs with a probability within each component, and a negative weight indicates that it is more likely within the U component. Because the U component is typically much larger than the M component it may still be the case that, even for comparison vectors with positive weight, a greater absolute number of them are estimated to occur within the U component. We see in Figure 2.5 that, for agreements on each of the fields (shown in green) the algorithms generally estimate similar weights, with the penalized likelihood estimate estimating a larger marginal weight for last name, and the EM algorithm estimating a larger marginal weight for education level. Looking again at Table 2.1 we can see that the total number of record comparisons which agree for each field is: 67 for last name, 237 for education, and 765 for sex. While ground truth is unknown for this dataset, at most 34 record pairs can correspond to true matches under one-to-one matching. This suggests that among non-matching record pairs agreement on education may be more frequent than agreement on last name, which takes many more

unique values. In such a scenario we would expect to see a larger weight assigned to matching on last name than matching on education level, the result yielded by the penalized likelihood estimator.

Among the marginal weights for disagreements (shown in brown) there is less consistency between the algorithms, and a stronger dependence on θ for the penalized likelihood estimator. With the exception of the education field the EM-base estimates are typically much closer to zero (although still negative), estimating that disagreement on the field comparison occurs at a similar rate within both the matches and non-matches. The effect of varying θ is also more clearly seen, as it increases the algorithm becomes more stringent in the types of disagreements it views as disqualifying. For $\theta = 0$ agreement on two fields is sufficient for a record pair to be considered for matching, for $\theta > 0.24$ the record pair is only allowed to disagree on sex or education level, and for $\theta > 2.23$ only disagreement on sex is allowed. Finally, for $\theta > 3.04$ only record pairs which agree on all three fields are considered for matching. This is consistent with θ defining a prior over the link structure which places an increasingly large prior probability on link structures with fewer matches. In the right panel of Figure 2.5 we plot the number of matches against θ (in green) as well as the expected number of matches under the prior (black). The estimated number of matches varies from 31 when $\theta = 0$ to 24 when $\theta > 3.04$ (and $\theta < 6.21$). The shaded grey region shows the range from the 0.5% quantile of the prior to the 99.5% quantile. The fact that the green line stays within this region over a range of θ values suggests that there is considerable uncertainty over the true number of matches. In the absence of ground truth observations the right panel of Figure 2.5 suggests that the estimated link structure is sensitive to the value of θ.

2.3.3 Robustness examples

To evaluate the performance of the penalized likelihood estimator we employ synthetic data from Sadinle (2017) using a data generator developed by Christen and Pudjijono (2009); Christen and Vatsalan (2013). From each of 100 data files we estimate the link structure between two sets of 500 records ($n_A = 500$, $n_B = 500$). Each record contains four fields: given name, family name, age and occupation categories. We run simulations for each data file where errors are introduced into 1, 2 or 3 of the 4 fields for each record and the share of records which are linked is 100%, 50% or 10%, resulting in a total of 900 simulations. Given name and family name are compared based on a Levenshtein similarity measure with similarity level bins of: exact agreement, (0%, 25%] mild disagreement, (25%, 50%] moderate disagreement, and (50%, 100%] extreme disagreement. Age and occupation categories are coded as either matching or non-matching. Following Sadinle (2017) we use a flat Dirichlet distribution as a prior on similarity levels for all fields for both the matched and non-matched record pairs, that is both the M component and the U component of the mixture model. See Sadinle (2017) for further details. As a pre-processing step we randomly permute the records in data file A as, by construction, all matches occur on the diagonal within the original data. As

such an artificial structure may be exploited by some assignment algorithms causing them to appear more efficient on the synthetic data than could be expected in real-world settings.

We beginning by fitting the penalized estimator for each of the 900 simulations setting θ to 0, 2, 4, 6, and 8. We fit estimator varying the algorithm used to solve the assigning problems within the penalized likelihood estimator. This does not affect the performance of the estimator, but allows us to compare the computational performance of each assignment algorithm. The assignment algorithms considered are: (1) the standard Hungarian algorithm, (2) first running graph clustering and then applying the Hungarian algorithm to each component (as described in Section 2.2.1), (3) a standard Auction algorithm, (4) first running graph clustering and then applying the Auction algorithm to each component, and (5) an Auction algorithm using price updating as suggested in Section 2.2.2[§].

The resulting runtimes are shown in Figure 2.6, with runtimes shown on a log-scale. The runtimes for the three estimation procedures that use an Auction algorithm return fairly similar runtimes across all simulation parameters. As discussed previously, there is reason to believe that Auction algorithms may, to an extent, implicitly apply clustering as rows and columns in different graph components will not compete against each other in the auction bidding process. So it is unsurprising that addition a graph clustering step does not substantially improve the runtime, and may increase it slightly, relative to a plain implementation of the Auction algorithm. It is somewhat more surprising that adding cost updating between the iterations of the penalized likelihood estimator does not decrease the runtime of the procedure. But, this may be a case where the additional overheard required by such a step outweighs the performance gains, at least for problems of this size.

In contrast procedures that rely on an Auction algorithm, the relative performance of the procedures which employ a Hungarian algorithm vary strongly with the value of θ as well as the overlap between the two files. In the 100% overlap simulations the standard Hungarian algorithm performs only marginally worse than the Auction algorithm based procedures. While, for sufficiently large values of θ, the additional of a graph clustering step with the Hungarian algorithm achieves performance comparable to that of the Auction algorithm based procedures. This result, that adding the graph clustering step to the Hungarian algorithm achieves performance comparable to the Auction algorithm based procedures, but only for larger values of θ, holds across simulation parameters. Indeed, we observe this pattern even in the 10% overlap simulations, a setting were we might expect the small number of true matches, which we expect to be high weight, to result in a graph that readily separates into distinct components. However, in practice we see that for $\theta = 0$ and $\theta = 2$ combining graph clustering with the Hungarian algorithm yields little benefit, suggesting that the graph clustering may be yielding a largest component that contains nearly all of the record pairs. In contrast, for $\theta = 4$, $\theta = 6$ and, $\theta = 8$ the Hungarian algorithm combined with graph clustering procedure achieves

[§]Both the Hungarian and Auction algorithms were implemented from in the Julia programming language by the author. Code and additional implementation details are available here: https://github.com/brendanstats/AssignmentSolver.jl

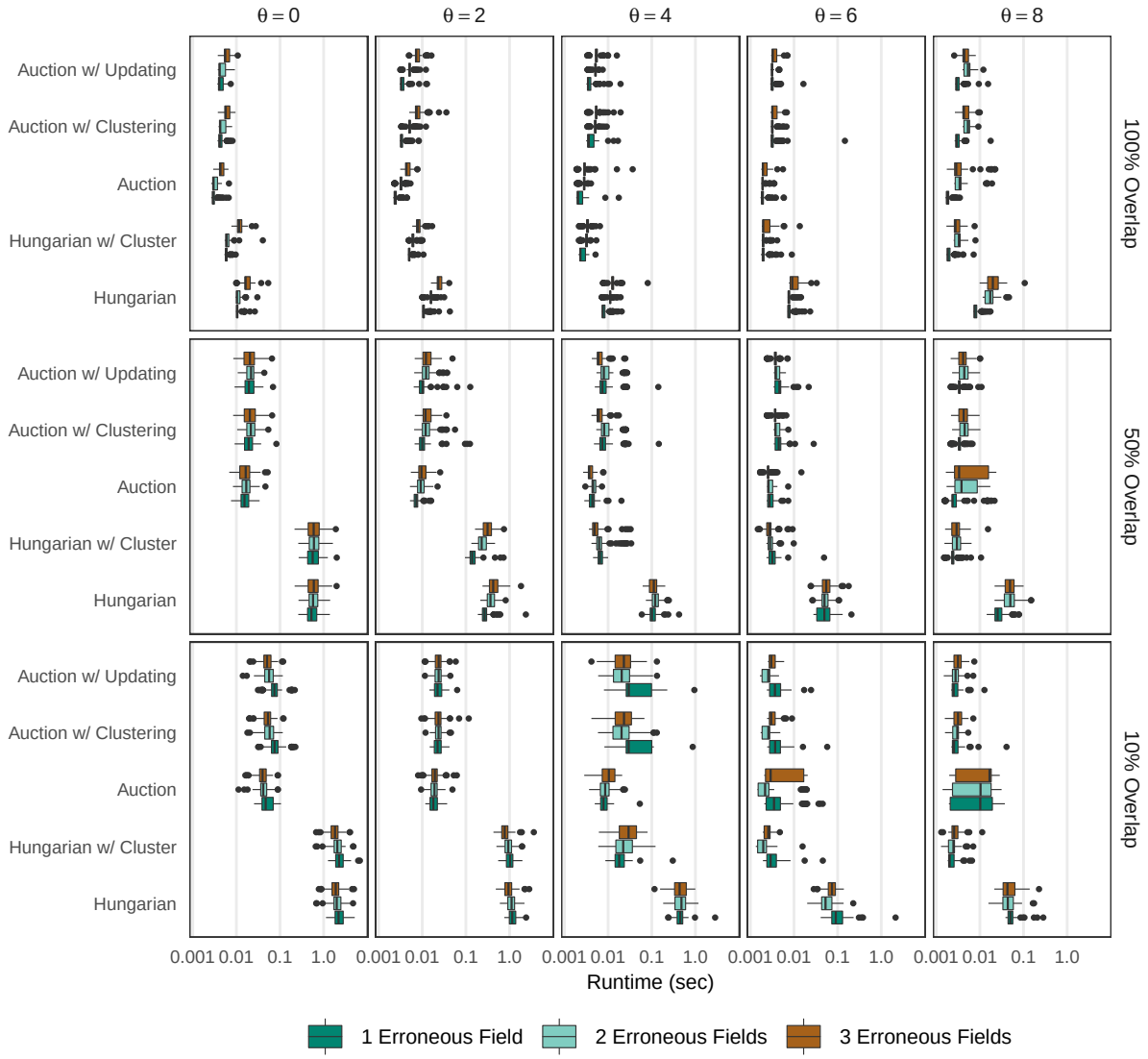


Figure 2.6: Timing of penalized likelihood estimate varying algorithm used to solve the assignment problem. Across the board there is little difference between the three methods which employ an Auction algorithm. In contrast the performance of the Hungarian algorithm is significantly improved by the inclusion of a graph clustering step, but only for larger values of θ.

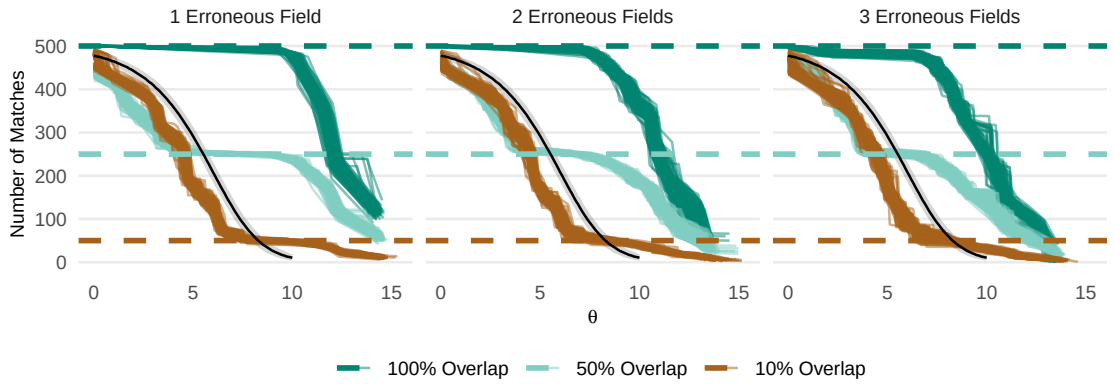


Figure 2.7: Number of matches estimated for each dataset as a function of θ. The black line (with a 99% coverage interval shown in grey) marks the expected value of the corresponding prior. Dashed lines plot the true number of matches for each overlap level.

performance comparable to the Auction algorithm based procedures. This suggests that, for low values of θ, across all simulation scenarios, the penalized weights $\tilde{w}$ estimated by the penalized likelihood estimator are positive for a non-trivial fraction of the non-matching record pairs. We also note that the number of error-free fields seems to have little effect on the runtime of the algorithm, this is perhaps unsurprising since the model we estimate remains consistent with the underlying data generating process.

As in the Italian Census example from Section 2.3.2, the number of links identified by the penalized likelihood procedure is strong influenced by the choice of penalty. To efficiently explore a range of penalty values we first compute the penalized likelihood estimator with $\theta = 0$, generating estimates of C, the m-parameters, and the u-parameters. We then iteratively increase θ, selecting the new value such that, at each iteration, the new value is larger than the smallest weight of the previously estimate to C. This ensures that the value of C estimated in the next iteration will differ from the current estimate. As in the preceding example we repeat this procedure for each of the 900 simulation scenarios from Sadinle (2017).

The number of matches found within each scenario is shown in Figure 2.7 plotted against the corresponding value of θ with the true number of matches shown as a dashed line. We also plot the expected number of links (with a 99% coverage interval shown in grey) under the distribution defined by taking θ as a parameter for (1.19), the prior defined by Green and Mardia (2006). Across all scenarios the estimated number of matches falls as θ is increased. Encouragingly, we note that the estimated number of matches aligns closely with the true number of matches when the prior places significant weight near the true number of matches. Indeed for the 50% and 10% overlap scenarios we can see that the estimated number of matches is close to the true number of matches for a range of θ values roughly center on the value of θ for which the prior expected number of matches equals the true value. Unsurprisingly, this range appears to be wider in the scenarios with fewer erroneous fields, corresponding to higher signal generative processes.

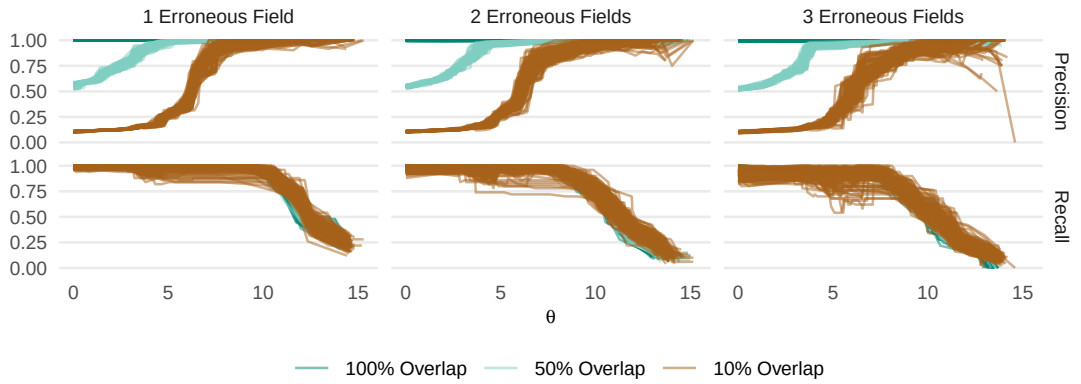


Figure 2.8: Precision and recall as functions of θ. Lower values of θ typically result in near perfect recall but poor precision with the opposite occurring with larger values of θ

Just because the correct number of matches is estimated does not guarantee that the correct matches have been identified. This is particularly true when there is less than 100% overlap between the datasets. We therefore plot the precision and recall of the set of estimated matches in Figure 2.8. What we see is entirely consistent with the results in Figure 2.7. For low values of θ, when too many matches may be identified, the recall is near 1, meaning that all true matches are correctly identified, but precision is significantly below 1. As θ is increased precision increases with no cost to recall, until θ is raised to such a value that fewer than the true number of matches is identified. Past this point precision remains close to 1, indicating that nearly all estimated matches are made correctly, but recall falls with the number of estimated matches. For values of θ near that at which the prior expected number of matches is near the true value both precision and recall are at or near 1. Although it appears that in the scenario with three erroneous fields the signal to noise ration may be such that perfect precision and recall cannot be achieved simultaneously for any value of θ.

Finally, we examine the estimated m parameters, which correspond to the probabilities of observing an exact match on the first and last name fields for a matching record pair. We select these parameters because it is easy to determine what the true value is from the description of the generative model provided in Sadinle (2017). We plot the estimated values as a function of θ in Figure 2.9 with the true values, which are independent of the level of overlap, shown as dotted black lines. As observed in Figures 2.7 and 2.8 there is general a range of θ values, coinciding with the prior expected number of matches is closely aligned with the true value, in which the estimated values are close to the true value. For lower values of θ, under which too many matches are estimated, the parameter tends to be underestimated. This is consistent with many non-matching record pairs, which contain fewer exact agreements, are included in the M component. For large values of θ, when very few matches are estimated, the behavior appears to be much less stable. In such cases only a subset of the true matches are identified as such (as shown in Figure 2.8) but it is unclear how this

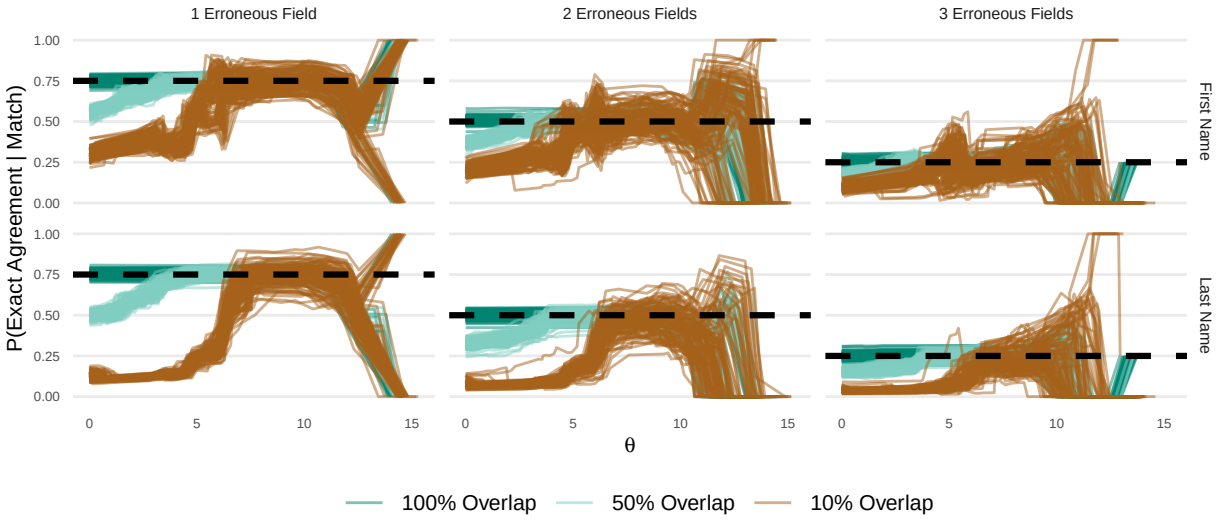


Figure 2.9: Estimated probability of exact agreement on first name and last name comparison as a function of θ. True values from the generative model are shown in the dashed black lines.

subset is selected, and for extreme values there may be multiple modes. In such a simulated setting, where all true matches are expected to correspond to an equally strong signal, this processes is probably influenced by incidental correlations between the distribution of comparisons within the M and U components. We see for example that with 1 erroneous field, for large values of θ, the estimated probability of agreement on first name typically approaches one, while the probability of agreement on last name approaches zero. Yet in a non-trivial number of simulations the opposite happens. However, we note that for $\theta = 10$ 99% of the prior probability is placed on between 4 and 19 matches occurring, far fewer than in even the 10% overlap case. Thus, as long as some prior information on the expected number of matches is available such extreme values of θ can be avoided.

2.3.4 Discussion

In this chapter we have made several contributions to the question of generating a point estimate of the link structure under one-to-one matching. In particular we have reexamined the use of LSAPs to resolve a set of weights into an estimate of the link structure. We have shown that using either thresholded or penalized versions of these weights, as defined in (2.4) and (2.9) respectively, leads to both a better estimate of the link structure and, through induced sparsity, a more tractable optimization problem. We then outline several theoretical reasons why Auction algorithms may be particularly well suited to solving the resulting LSAPs. Finally, we use the computational advances to develop a new penalized likelihood estimator which requires many more LSAPs to be solved, but allows the one-to-one matching constraint to be enforced throughout

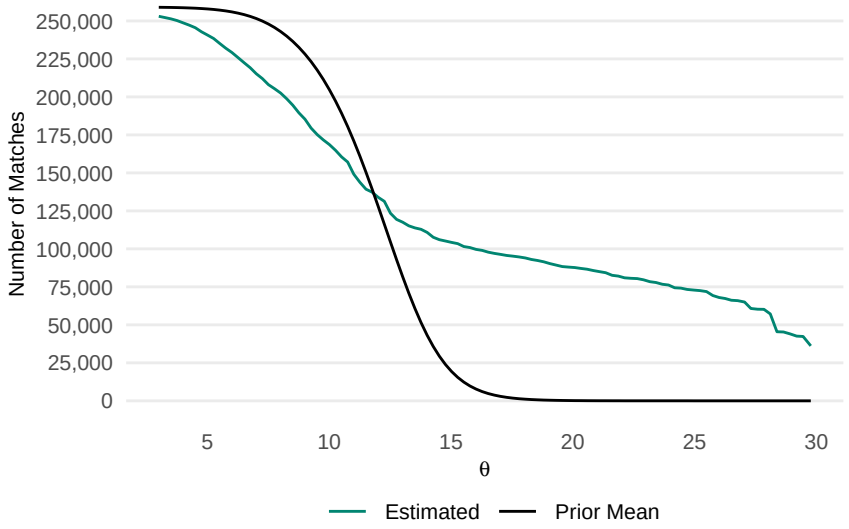


Figure 2.10: Number of matches estimated in Alameda county voter data as a function of θ. The total runtime of the estimation is under an hour.

the estimation of the matching parameters. However, we have yet to provide evidence that the penalized likelihood estimator can be applied practically to truly large record linkage problems. Furthermore, it is clear from the results that the performance of the penalized likelihood estimator is highly dependent on the value of the penalty parameter θ. Yet, we have not provided guidance on the selection of this crucial parameter. We address the first of these questions applying to penalized likelihood estimator to a larger record linkage problem, the details of which will be discussed further in Chapter 3. For the second we suggest strategies which may be employed to select θ depending on the level of evaluation which can be applied to the problem.

While the examples presented in this chapter are all relatively small, 500×500 or smaller, the algorithmic advances presented in Section 2.2 make it possible to apply the penalized likelihood estimator to significantly larger problems. In Figure 2.10 we plot the estimated number of matches from the penalized likelihood estimator for two voter registrations files from Alameda county, data which we discuss in more detail in Chapter 3. The two files to be matches contain approximately 260,000 and 290,000 record respectively, nearly three orders of magnitude larger than the problems presented in this section. Yet, the total runtime of the estimation on this problem is under an hour, similar to the amount of time required to compute the comparison vectors. Finally, we note that, for even larger problems, if the runtime appears intractable the level of sparsity (and hence the ease of solving) can always be increased by initialing the procedure at larger θ value. Indeed, although not examined here, such an approach may be a reasonable method of initializing the parameters before computing the estimator with a lower value of θ.

We next consider the problem of selecting an appropriate value of θ. In general we recommend, as a first step, always computing the penalized likelihood estimator across a range of values for θ. Performing such a

sensitivity analysis is straight forward and yields additional information. At a minimum it allows a sensitivity analysis to be performed not just on the estimated link structure but on any down stream quantities. If these appear relatively stable over a range of θ values, as seen in the precision plotted in Figure 2.8 for the 100% overlap simulations, then it may be the case that the downstream estimation is not particularly sensitive to the exact value of θ selected. If however, the quantity is more sensitive to the value of θ, such as the number of estimated links shown in Figure 2.10 then the selection of θ can be aided by either the incorporation of prior information on the expected number of links, or by the considering labeled data.

If for reasons of cost or difficulty of implementation no labeling of data is possible then we recommend the selection of an appropriate value for θ by interpreting the penalized likelihood estimate as a MAP estimate implied by the value of θ. This approach assumes a prior distribution over the link structure of the form introduced by Green and Mardia (2006) (and defined in (1.19)). Under this framework one can, a prior, set an expected number of links, which will in turn correspond to a value of θ. We plot this value in Figure 2.10 as a black line. If no prior information on the expected number of links is available then an alternative approach is to set θ to the value at which the expect number of links under the corresponding prior equals the estimated number of links (where the lines cross). In the simulation study presented in Section 2.3.3 this approach appears to perform well. However, we caution over interpreting these results as within the simulation study the generative model closely aligns with the fitted model, whereas in real world examples it is common from some model assumptions to be violated, particularly the conditional independence assumption.

In the absence of prior knowledge about the expected number of matches within a dataset we recommend selecting a value of θ through the use of training data. If such data is already available then θ can be selected be setting a loss function which will evaluate the cost of false matches (more likely under lower values of θ) vs. false non-matches (more likely under larger values of θ). In the frequent scenario where labeled data is not available, but can be generated at some cost then we recommend a more ad-hoc approach where instead of evaluating all matches only the marginal matches are evaluated. Consider again Figure 2.10, at $\theta = 10$ approximately 175,000 matches are estimated, at $\theta = 15$ 110,000, and at $\theta = 20$ 90,000 matches. Furthermore, it is likely that nearly all of the 90,000 matches found with $\theta = 20$ are also included in the 110,000 estimated matches under $\theta = 15$. We can exploit this by labeling a sample of the 20,000 matches estimated when $\theta = 15$ but not when $\theta = 20$. If this set is found to contain mostly matches then we can conclude that this likely true of the 90,000 matches we have not examined, indeed the set of 90,000 matches identified under a larger penalty is likely to be higher quality. We may thus conclude that we should set θ to 15 or lower. If however, we find that the labeled sample contains many non-matching record pairs then we can instead conclude that we should set θ to some value larger than 20. We can repeat this processes until we find a value of θ above which the marginal set of links is composed of the desired share of matches and below which the marginal set contains an unacceptably high share of non-matches. Such an iterative procedure bears some similarity to the one suggested by Larsen and Rubin (2001).

A final shortcoming of the penalized likelihood estimator is that it fails to identify cases where a single record is part of multiple record pairs which appear to be matches (correspond to high weights). As shown in Table 2.1, there are a total of 25 record pairs in which the records match across all three fields. Yet, only 24 of them are linked, because the 25th contains a record that is also included in one of the other 24, so that the two record pairs cannot be matched simultaneously under a one-to-one matching constraint. Because it reports only a single assignment, for each value of θ, the penalized likelihood estimator is unable to identify this uncertainty and simply links one of the record pairs, essentially at random (unless otherwise constrained by the link structure). In contrast Bayesian methods, which estimate a posterior over the link structure, will characterize this uncertainty but have thus far been intractable for large record linkage problems. In the next chapter we show how the penalized likelihood estimator can be used in an initial pre-processing step in a procedure which allows fully Bayesian models to be estimated for large record linkage problems.

Chapter 3

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

PRL is inherently computationally expensive; with files of size n_A and n_B there are $n_A \times n_B$ record comparisons to be made. We saw in Chapter 2 that the introduction of sparsity, by excluding record pairs weight low weights can make PRL problems significantly more tractable. Unfortunately, the penalized likelihood estimator we introduce in Chapter 2 fails to include an estimate of the uncertainty in the link structure, a serious shortcoming. Bayesian PRL, as introduced in Section 1.6, can easily estimate such uncertainty via a posterior distribution. However, in the absence of a high-quality blocking field, Bayesian PRL is generally not computationally tractable for files containing more around a thousand records. In this chapter we introduce a method for estimating Bayesian PRL models which, by excluding low-quality record pairs as done in Chapter 2, allows an approximate posterior distribution for the link structure to be estimated for significantly larger files than was previously possible. Crucially, this method does not assume the files contain high-quality fields suitable for use as blocking-key.

3.1 Post-Hoc Blocking for Bayesian PRL

Given their computational complexity, Bayesian implementations of PRL can benefit from stricter blocking or indexing than other methods. Stricter indexing increases the risk of false non-matches, so it is important

that the indexing be as efficient as possible by admitting plausibly matching record pairs while excluding clearly non-matching pairs. We propose constructing a high-quality blocking key from the available fields which separates most of the obviously non-matching pairs across blocks while keeping plausible matches within the same block. This is difficult to do via traditional indexing *a priori*, especially in the absence of labelled matches, so we suggest a data-driven approaches to choose the blocks (with or without labelled matching and non-matching pairs) – hence the name *post-hoc blocking*.

Post-hoc blocking is straightforward: First, perform traditional blocking or indexing only to the extent necessary to make computing 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 relatively high weight to plausible matching pairs and low weight to true non-matching pairs; they need not be well-calibrated probabilities or proper likelihood ratios. Third, conduct an additional blocking pass using the estimated weights to construct the blocking key, reducing the number of record pairs just enough to make running an MCMC algorithm feasible. With the post-hoc blocks in hand, we run an MCMC algorithm as usual, restricting the proposal distribution to only consider matches within post-hoc blocks.

Figure 3.1 illustrates the process of post-hoc block generation. The rows and columns of the heatmaps correspond to records from file A and file B, respectively. Panel (a) 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 a low value w_0. Panel (b) 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 (shown below the heatmap); an edge is present between records a_i and b_j if 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 Panels (c) and (d). Post-hoc blocking significantly reduces the number of candidate pairs while identifying a block of records that appear to have multiple plausible configurations (post-hoc block 1, in blue). After a first pass, if any of the remaining post-hoc blocks are too large, we increase w_0 and apply this procedure recursively within the large blocks.

The procedure for sampling from an approximate posterior distribution for C employing post-hoc blocking is summarized in Algorithm 2 below; implementation details follow.

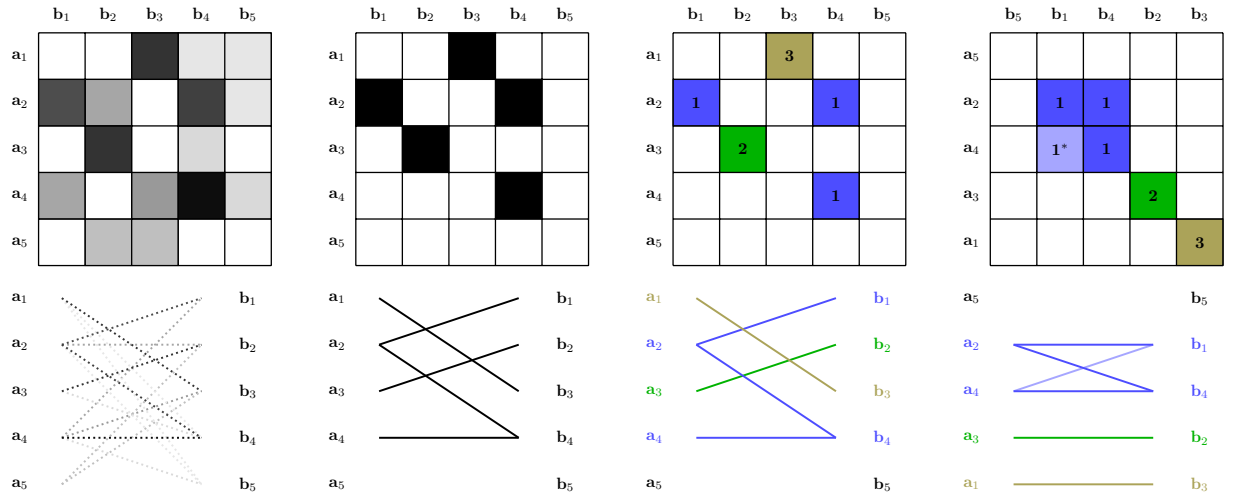
Weight estimation. Clearly the performance of post-hoc blocking will depend on the quality of the weights. However, compared to using the weights to *identify* truly matching pairs, for the purposes of post-hoc blocking we can tolerate lower quality weights. What is essential is that they give high weight to truly matching and ambiguous record pairs, while giving low weight to clearly non-matching pairs (so the blocks are compact). It is less important that the weights give a good rank ordering of the truly matching/ambiguous pairs – as

Algorithm 2 Post-hoc Blocking with Restricted MCMC

Input: Comparison vectors Γ for a set of record pairs, initial weight threshold w_{min}, maximum post-hoc block size N_c

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 $e_{ab} = \mathbb{1}(\hat{w}_{ab} > w_0)$ with $w_0 = w_{min}$
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. For post-hoc blocks larger than N_c repeat 2. and 3. with a threshold $w_0' > w_0$. Apply recursively on any resulting post-hoc blocks larger than N_c
5. Run a standard MCMC algorithm, fixing $C_{ab} = 0$ for all record pairs outside of the post-hoc blocks.



(a) Weights **(b)** Edge Matrix **(c)** Connected Components **(d)** Post-hoc Blocks

Figure 3.1: An example of post-hoc blocking: the top figure of each panel shows an edge matrix and the corresponding bipartite graph is shown in the bottom figure. (a) shows an example of estimated weights with darker 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.

long as they end up in the same post-hoc block, the Bayesian model and MCMC algorithm will treat them appropriately.

If labelled true matching and non-matching record pairs are available we could use these to predict matching probabilities for the remaining pairs using standard classification methods. These predicted probabilities will often fail to be well calibrated; for example, when a record in file A has multiple plausible candidates in file B they may all receive high matching probabilities with a classifier trained treating record pairs as iid observations. Regardless, these records will be gathered into the same post-hoc block and the uncertainty in the matching structure will be accurately represented in the posterior distribution.

Alternatively, in the absence of labelled record pairs we could use EM estimates of the Fellegi-Sunter weights (1.5) as post-hoc blocking weights. This can work well in settings where there are many attributes available for matching and where most of the records in A also appear in B (Winkler, 2002). 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 (Winkler, 2002; Tancredi et al., 2011; Sadinle, 2017). More reliable weights can be obtained by getting coarse estimates of $m-$ and $u-$ probabilities while accounting for the one-to-one matching constraint. We outline a method for obtaining such weights using a novel penalized likelihood procedure in Section 3.2.1; this is how we generate the post-hoc blocking weights for our application in Section 3.3.1.

Obtaining 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 computationally efficient solutions (Tarjan, 1972; Gazit, 1986).

Selecting the maximum block size N_c. Choosing the maximum block size N_c requires balancing statistical accuracy (the quality of our posterior approximation) against computational efficiency. Smaller values of N_c are more likely to exclude true matching pairs, increasing the false non-match rate. Excluding truly non-matching pairs which are not *obviously* non-matches also risks misrepresenting posterior uncertainty. On the other hand, selecting a larger N_c decreases bias by admitting more record pairs and yields a smaller number of post-hoc blocks of larger size. Larger post-hoc blocks lead to increased computation time, as the most significant computational gains accrue when a large fraction of the post-hoc blocks are small. Given these considerations we should choose the largest N_c that leads to a computationally feasible MCMC algorithm. What constitutes a "feasible" problem will naturally be context dependent.

Implementing restricted MCMC algorithms. Post-hoc blocking can achieve massive reductions in scale relative to traditional blocking schemes. 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 possible a restricted MCMC algorithm which mixes much more efficiently than standard approaches.

For very small blocks we perform Gibbs updates by enumerating all possible values of the corresponding submatrix of C and sampling proprtional to their unnormalized posteriors. Other implementations of

Bayesian PRL have taken advantage of this enumerability when a large number of high-quality traditional blocking fields are available (e.g. Gutman et al. (2013)). However, post-hoc blocking is more likely to produce a large number of small blocks than traditional blocking, especially in the absence of one or more high-quality blocking keys.

For moderately-sized blocks, informative locally balanced Metropolis-Hastings proposals can be used instead of simple add/drop/swap proposals (Zanella, 2019). Zanella (2019) 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 containing 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 (2019) mitigated this issue by including random sub-block generation as part of the locally balanced proposal. But as the file sizes increase random sub-blocks, if held at a fixed size, are increasingly unlikely to capture all or even many of the plausible candidates for each record in the block, increasing mixing time. The alternative, allowing the size of the sub-blocks to grow with the file size, will result in a quadratic growth in the cost of performing the update. In contrast, our post-hoc blocks are specifically constructed to capture all the plausible candidates for a given record in the same compact 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, as shown in Section 3.2.2. In large problems, an approximation of some sort seems unavoidable – it is infeasible to run any MCMC algorithm over datasets with hundreds of thousands of records generating hundreds of millions or billions of candidate record pairs (after indexing) sufficiently long to mix properly. The result is that in a practical MCMC run the vast majority of those entries we fix at zero would have posterior probabilities estimated at or near zero anyway. With post-hoc blocking and restricted MCMC using locally balanced proposals we are able to hone in on areas of non-negligible posterior uncertainty, and spend more of our time sampling in these regions.

3.1.1 Post-hoc Blocking Versus Traditional Blocking/Indexing/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

*At the cost of significant additional bookkeeping it is possible in principle to include the post-hoc blocking thresholds as part of a proposal distribution similar to Zanella (2019); we leave this extension for future work.

model fitting. The same is typically true under filtering – the record pairs that are filtered *after* a complete comparison have been made are ignored during model fitting, even though their comparison vectors are available.

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 are used to estimate model parameters. We take a similar approach to record pairs excluded by the initial blocking/indexing scheme – although their comparison vectors are not available, we can compute the relevant summary statistics exactly under a conditional independence model (see Section 3.4.1 for details). This avoids some of the more pernicious bias-inducing effects of blocking, indexing, and filtering on subsequent parameter estimation described by Murray (2016). In the model introduced in Section 1.5.2 this amounts to adding additional record pairs directly to the U component, so we call this step a U-correction. We examine the effect of the U-correction further in Section 3.4.1.

3.2 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 provided one example of this in Section 2.3.2. 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.

3.2.1 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

Last	Sex	Edu	Count	EM Weight	Maximum Weight
1	1	1	25	5.27	6.21
1	0	1	8	3.77	3.04
1	1	0	13	-0.94	2.23
0	1	1	126	2.68	0.24
1	0	0	21	-2.45	-1.03
0	0	1	78	1.18	-3.14
0	1	0	601	-3.53	-3.81
0	0	0	658	-5.04	-7.19

Table 3.1: Maximum weights used for post-hoc blocking, and EM weights for comparison

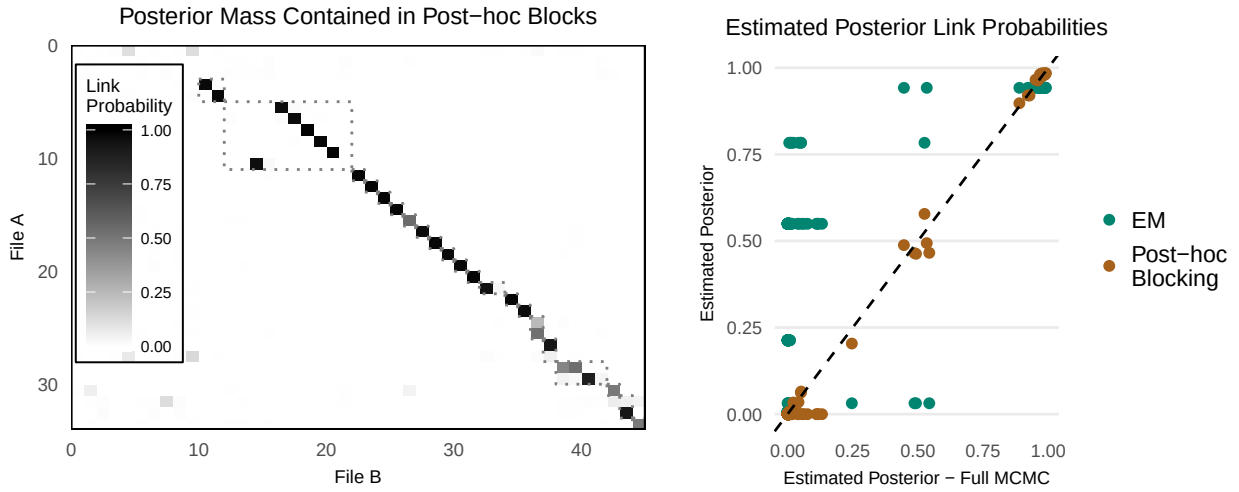
pair as the post-hoc blocking weight. This obviates the need to calibrate θ and assigns relatively high weight to any record pair that is a plausible match candidate for *some* value of θ.

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, which we outlined in Section 2.3.3 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. As discussed in detail in Chapter 2, Auction algorithms can provide an efficient means of updating the estimate of the link structure as the penalty is increased. Furthermore, since the weights need only be approximately correct, if the computational cost of computing the sequence of weights becomes too high the tolerance of the estimated link structure (controlled by the ϵ parameter of the Auction algorithm) to small deviations from optimality can be increased.

3.2.2 Illustrations of Post-Hoc Blocking and Restricted MCMC

We again consider the small scale Italian Census data introduced in Section 2.3.2 to illustrate the performance of post-hoc blocking with maximal weights. We consider the same model as in Chapter 2 but instead of applying the penalized likelihood estimator we apply a full Bayesian model. The prior over the linkage structure is set to a Beta-bipartite distribution with $\alpha = 1.0$ and $\beta = 1.0$, which is uniform over the expected proportion of matches (Fortini et al., 2001, 2002; Larsen, 2005, 2010; Sadinle, 2017). We again assume a conditional independence model for $m-$ and $u-$probabilities as in (1.9). Each vector of conditional probabilities is assigned a Dirichlet prior distribution. We assume that $m_j \sim \text{Dir}(1.9, 1.1)$ and $u_j \sim \text{Dir}(1.1, 1.9)$ 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 estimate post-hoc blocking weights using the maximal weight procedure in Section 3.2.1. The resulting weights for each possible comparison vector are shown in Table 3.1, along with EM weights for comparison. Notable discrepancies are in gray. The maximum weights are simply an aggregation of the different weights



(a) Post-hoc Blocks

(b) Posterior Match Probabilities

Figure 3.2: (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 match probability considering all record pairs.

estimated by the penalized likelihood estimator presented in Table 2.1. Therefore we do not repeat our discussion of the discrepancies but simply provide the table for reference.

Given the small size of the problem we select only a single post-hoc blocking threshold w_0 to implement the restricted MCMC. In our post-hoc blocking procedure we limit the size of the largest post-hoc block to fewer than 100 record pairs. The resulting post-hoc blocks contain only 94 of the 1530 possible record pairs. These are spread across 21 separate post-hoc blocks. Of the 21 post-hoc blocks, 14 contain only a single record pair, 4 contain 2 record pairs, the remaining three contain 4, 8, and 60 record pairs respectively.

We then run both a MCMC algorithm containing all 1530 record pairs and our restricted MCMC under identical model specifications. Results from both models are displayed in Figure 3.2(a), 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 3.2(b) we compare the posterior match 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 0.12 – 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 3.4 seconds for the full MCMC algorithm versus 0.32 seconds when employing

post-hoc blocks. The order of magnitude reduction in runtime 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 probabilities 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 3.1. But the failure to account for one-to-one matching, discussed extensively in Chapter 2, seems to play a larger role. In general we would expect omitting 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.

In the next section we demonstrate the application of our post-hoc blocking method to the PRL problem of linking voter registration files. These files are orders of magnitude larger than those presented so far, containing hundreds of thousands of records. Yet, we are able to link them in a matter of hours. Furthermore, we then use the estimated posterior distribution to propagate uncertainty in the link structure into estimates derived from the linked data. The ease of uncertainty quantification is a significant advantage of Bayesian PRL which we have yet to discuss fully.

3.3 Linking the California Great Registers

Beginning in 1900, California counties were required to publish a typeset copy of their voter lists in each election year (Spahn, 2017), known as the California Great Registers, which contain the name, address, party registration and occupation of every registered voter. This 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 dataset that follows this change at the individual level, provided we can link individual voters over time.

One quantity of interest to historians and political scientists is the frequency with which voters changed party affiliation from 1932 to 1936, during Roosevelt's first term as president (Erikson and Tedin, 1981; Andersen, 1979). Decades of surveys conducted since 1948 have shown that voters rarely switch parties. But this earlier period, before modern polling, featured the most dramatic and rapid change in partisanship in the twentieth century, making individual-level panel data from this period especially interesting. In particular, individual-level panel data would enable a more detailed study of party switching behavior by demographic

groups (Sundquist, 1983; Corder and Wolbrecht, 2016; Norpoth, 2019). To construct such a panel, we link records from the across the 1932 and 1936 registers based on the recorded name, address and occupation.[†]

Though the structure of the data is relatively simple, transferring it from the printed page into digital format is challenging. Ancestry.com scanned and performed optical character recognition (OCR) on the Great Registers, enabling use of the data by their subscribers for genealogical research. Since the quality of the scan as well as the original organization of the page can make the OCR fail to produce recognizable text or mistranscribe certain words and letters, much of the data was digitized imperfectly. Once digitized, the data require further processing and standardization which is also subject to error. These errors, coupled with natural variation (such as address changes and inconsistent recording of name variants and occupations), make linking challenging.

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 49% of the time), making quality matches – and accounting for uncertainty in which records match – is essential to robustly estimating the party-switching rate. In addition to linking these two files, we also estimated false match rates to understand the performance of linkage algorithms and adjust our estimates, and compared the distribution of variables in the linked dataset to cross-sections of the Great Registers to help gauge whether differential false non-matching might threaten the representativeness of our linked sample.

3.3.1 Application to Alameda County

In our study of the Great Registers we link 1932 and 1936 voter registration files for Alameda county. We chose this location and period because the data quality in Alameda is relatively good, prior work suggests that the party switching rate over this period is relatively high, and we suspect that registers from presidential election years have a higher degree of overlap than those from adjacent election years. Data preprocessing details appear in Appendix 3.4. After cleaning and parsing the records from each year, we are left with 259,162 records from 1932 and 288,087 records from 1936.

We present two estimates of the links between the 1932 voter file and the 1936 voter file. The first are based on the Bayesian model described in Section 1.5.2 with post-hoc blocking and restricted MCMC. The second estimates are from a benchmark analysis using methods described in Enamorado et al. (2019), as implemented in the fastLink R package (Enamorado et al., 2018). FastLink utilizes a Fellegi-Sunter based model with parameter estimates computed via EM with specialized routines for blocking and post-estimation imposition of one-to-one matching constraints. We present error rate comparisons between the estimated links in Section 3.5.1 and differences in estimated party switch rates in Section 3.6.

[†]While party might be an informative field in making a match, we withheld it from the matching process so that our estimate of the key quantity of interest – the party switching rate – is not biased toward stability.

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

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

Table 3.2: String similarity to ordinal mapping. Jaro-Winkler string similarity (left) and zero-padded Levenshtein string similarity (right).

3.4 Data processing details

Before constructing comparison vectors we undertake a number of pre-processing steps. The suffix field is coded as missing so frequently that we are forced to discard it entirely. Name 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 construct 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 code the occupation variable as missing for housewives, as chance agreement on occupation for housewives is very common.

We split the given name field into separate first name and middle name fields. We also 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 re-coded (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. We discarded records missing two or more of the first name, surname, occupation, and street name fields.

3.4.1 Bayesian Model: Implementation Details

Before estimating the Bayesian model we reduced the set of potential matches via indexing by disjunctions of blocking keys. A record pair was included as a potential match if the first three characters of the given name or the first three characters of the surname matched exactly.[‡] This left about 850 million record pairs as potential matches.

Construction of comparison vectors

We used the Jaro-Winkler similarity score to compare first name, surname, occupation, and street name fields. We compared the street number field with a Levenshtein distance, using zero-padding to ensure the

[‡]Women's first names beginning with "mar" are exceedingly common in this period, so for records pairs with both first names beginning with "mar" we used four characters of the first name.

distance is calculated between strings of equal length. The resultant string similarities were converted to comparison vectors by binning (Table 3.2). We compared our constructed female indicator and the street type using exact matching (coded as 2 for a match and 1 for a non-match).

Modeling the similarity in the middle name field required a more nuanced approach because in many cases only a middle initial was recorded. We considered three cases: two full middle names present, two middle initial initials present, and a full middle name present in one record and only a middle initial in the other record. When two full names were present we used the same string comparison and cutoffs as for comparing first and surnames. For two initials, or one full name and one initial, we used exact matching between the first letter of the full name and the initial. This resulted in 10 possible similarity levels for middle name, the six in Table 3.2 for two full middle names, two for comparisons between middle initials, and two for comparisons between a middle initial and a full middle name.

Computing comparisons outside of the indexing for the "U-correction" on Bayesian models

It is well known that failing to account for the blocking or indexing scheme can result in bias in the estimated u-parameters (Murray, 2016). This bias is introduced because, under any reasonable indexing or blocking scheme, the distribution of even non-matching comparison vectors will differ substantially between record pars within the scheme and those excluded from it. For example, in the indexing scheme employed for our Bayesian model (Section 3.4.1) we consider only record pairs which match on the first three digits of first or last name. By design the scheme excludes record pairs which are dissimilar on both first name and last name, the overwhelming majority of record pairs. This means that record pairs displaying low levels of similarity on first name or last name will be underrepresented, potentially massively so, relative to what would be observed if all comparison vectors were computed.

Setting $C_{ab} = 0$ for all record pairs outside of the blocking scheme then, under a conditional independence assumption, the missing comparisons need only be generated marginally. That is, it is not necessary to calculate the full comparison vector for each record pair outside of the indexing or blocking scheme, only to determine frequency with which each similarity level would occur for each feature. This is a result of the fact that, under a conditional independence assumption, the likelihood factors in such a way that only the marginal frequencies with each group (match and non-match) are necessary as can be seen in (1.9).

Computing these marginal frequencies is much more tractable as it can be done by computing similarities only for observed unique values and weighting appropriately. For example, the first name john occurs 8,173 times in the 1932 data and the first name william occurs 8,349 times in the 1936. Hence, there will be a total of 68,236,377 record pairs (8,173×8,349) in which the first name in 1932 is john and the first name in 1936 is william but the string similarity need only be computed once.

Field	Unique Values 1932	Unique Values 1936	Required Comparisons
First Name	16,496	15,045	248,182,320
Middle Name	6,489	6,039	39,187,071
Middle Initial	37	32	1,184
Surname	55,648	54,715	3,044,780,320
Female	2	2	4
Occupation	17,164	18,115	310,925,860
Street Number	9,071	10,098	91,598,958
Street Name	15,256	8,316	126,868,896
Street Type	12	12	144

Table 3.3: Unique values observed within each year for each record field. Due to ORC errors numbers are sometimes observed in the middle initial field causing the number of unique observed values to be greater than 26.

In Table 3.3 we show the number of unique values observed for each field for each year. In total computing the full set of comparisons requires approximately 5 billion different comparisons. We note that while this is larger than the number of comparison vectors we generate for the analysis because multiple comparisons are required for each comparison vector even computing the full set of comparisons is less computationally costly than generating the comparison vectors. Furthermore, this is less than 1% of the nearly 675 billion comparisons that would be required if all possible comparison vectors were generated individually. While significantly more efficient approaches exist for generating the full set of comparison vectors (e.g. the implementation in Enamorado et al. (2019) using a hash table), at large scales other constraints, such as memory constraints, often become binding. Finally, while in our case we compute the marginal frequencies exactly it is possible to approximate them closely by computing similarities for record pairs generated via weighted random sampling of the unique values (with weights corresponding to observed frequencies). We refer to the inclusion of these frequencies as making a "U-correction" since the main effect is on the estimates of the u-parameters. We find that this correction is extremely important in practice.

Prior distributions, post-hoc blocking and restricted MCMC

Our prior specification for the model in Section 1.5.2 began with a Beta-bipartite prior on C with $\alpha = 1.0$ and $\beta = 1.0$. For first name, surname, occupation, and street name, $m_j \sim \text{Dir}(10, 6, 2, 1, 1, 1)$. For middle name $m_j \sim \text{Dir}(10, 5, 3, 6, 2, 1, 1, 1, 1, 1)$, where the weights of 5 and 3 correspond to exact matching between two initials and exact matching between an initial and the first letter of a full middle name, respectively. For street number $m_j \sim \text{Dir}(10, 6, 2, 1, 1)$, and for female and street type we set $m_j \sim \text{Dir}(5, 1)$. The prior distribution for all u_j vectors was uniform; these parameters are well-estimated from the data since most pairs are non-matches.

We fit the model via restricted MCMC using post-hoc blocks based on the maximal weights procedure detailed in Section 3.2.1 of the supplemental material. We set N_c, the maximum post-hoc block size, to

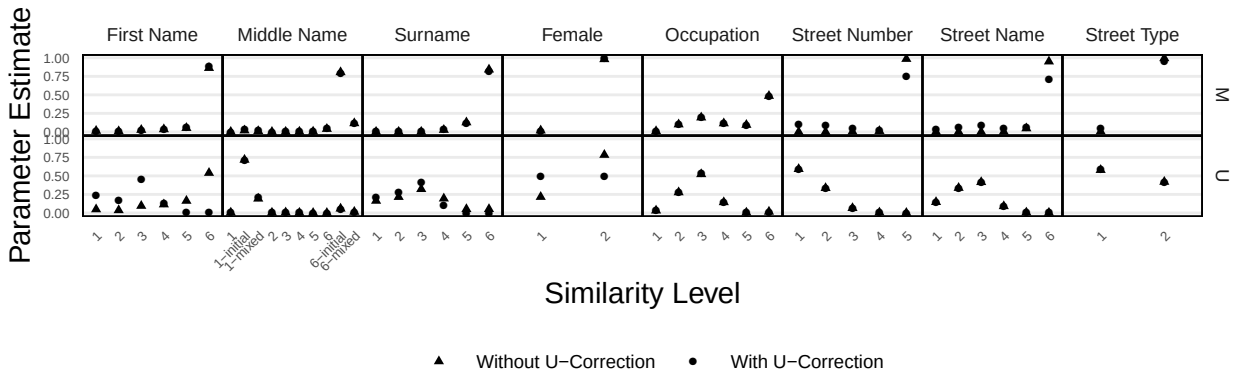


Figure 3.3: Posterior means of m-parameters (top) and u-parameters (bottom) with and without the U-correction.

250,000 record pairs. The resulting set of 94,997 distinct post-hoc blocks contains approximately 820,000 of the record pairs, a reduction of 99.9%.

On a 2014 Linux workstation, constructing the comparison vectors took about 2 hours, while estimating the weights and finding the post-hoc blocks took approximately 90 minutes. We ran the restricted MCMC algorithm for 25,000 steps, where each "step" comprises an update to the $m-$ and $u-$ paramaters in addition to a Metropolis-Hastings proposal within *every* post-hoc block (i.e., one step of the MCMC algorithm constitutes nearly 95,000 add/delete/swap updates to C, one within each post-hoc block). The restricted MCMC algorithm took 3.7 hours to run. In total it took under 7.5 hours to link these two files.

Effect of making the "U"-correction

We show the effect making the U-correction has on the parameter estimates in Figure 3.3. Examining the u-parameters in the bottom row it is clear the largest effect is on first name, one of the indexing fields, and the female indicator, which is likely correlated with first name. We see a smaller effect on surname, the other indexing field. Interestingly we also see changes in the estimated m-parameters for the address fields street number, street name, and street type. Here the observed change is a *decrease* in the estimated probability of observing an exact agreement in the address fields, conditional on the record pair corresponding to a match. This suggests that the model with the U-correction is matching a larger number of record pairs which differ on address, corresponding to the mover category discussed in Sections 3.5.1 and 3.6. The differences in parameter estimates also results in a larger number of estimated links.

Posterior distributions for the number of links with and without the U-correction are shown in Figure 3.4. The posterior distribution for the model with the U-correction (center) indicates both a larger mean number of estimated links, approximately 137,600 with the correction and only 87,200 without, as well as more dispersion in the number of links. The right panel shows the number of matches that would be returned

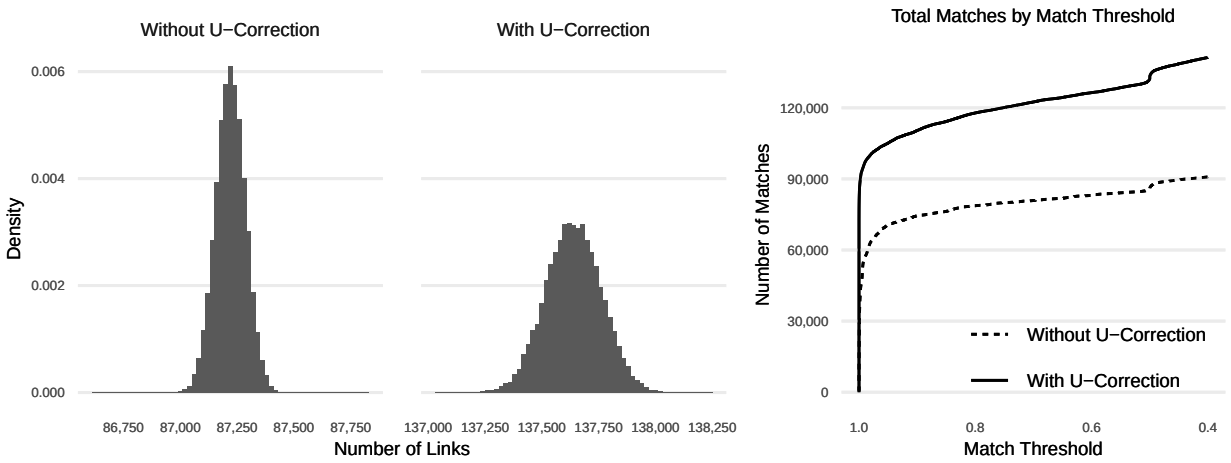


Figure 3.4: Posterior distribution of number of links for our Bayesian model with and without U-correction (left).

by each model as a function of the posterior probability threshold for declaring a record pair a match. In addition to identifying more matches, for all match thresholds, the larger slope of the U-correction line indicates that the model identifies many more record pairs with a non-trivial level of uncertainty about the match status, perhaps better characterizing the matching uncertainty. An inspection of a set of links assigned a substantially high link probability by the model with the U-correction suggests that most of these additional links correspond to true matches. In particularly, essentially all of the mover matches identified by the Bayesian model discussed in Section 3.5.1 are assigned a non-trivial link probability only by the model with the U-correction.

3.5 fastLink implementation

We relied on internal fastLink functions to block on first name, since fastLink was unable to use indexing by disjunctions by design. We used fastLink's built-in clustering function to generate blocks of maximum size 10,000 by 10,000 – much larger than our post-hoc blocks – resulting in 123 distinct blocks containing 1.1 billion record pairs. To generate comparisons with fastLink we used the Jaro-Winkler similarity score on all fields except for female and street type, for which we rely on exact matching. We were limited to three categories for the string comparisons by the fastLink package, so we followed recommendations in Winkler (1990); Enamorado et al. (2019) to set string similarity cutoffs; similarity above 0.92 corresponded to an "exact" match, between 0.88 and 0.92 a partial match, and below 0.88 a non-match on the field.[§].

[§]In very recent versions of the fastLink package the default 0.92 similarity threshold for an "exact" match was increased to 0.94.

Enamorado et al. (2019) suggest declaring matches by thresholding estimated posterior probabilities (derived from (1.10)) at a value between 0.75 and 0.95. Based on early inspection of the results we chose a cutoff of 0.9. Like other EM-based methods, fastLink does not incorporate a one-to-one matching constraint during estimation, so we relied on the fastLink's deduplication procedure to produce a set of record pairs consistent with the one-to-one matching assumption. Roughly, the fastLink deduplication procedure limits declared matches to those record pairs for which the estimated posterior probability is the observed maximum for both records across all possible record pairs involving one of two records. In the case where multiple record pairs achieve the maximum, ties are broken by sampling uniformly among the candidates.

3.5.1 Comparing the Bayesian model and fastLink

Before estimating party switching rates we compared the links made by each method. In particular, we focus on estimating the overall match rate and the rate of false matches. Overall, we find that the Bayesian model makes many more true matches at a significantly lower false match rate. Since choices about string similarities and blocking procedures do not explain the gap in results between the Bayesian model and fastLink; we discuss some important remaining differences between the methods below, and how they might impact performance.

Common versus distinct parameters and the $U-$correction. fastLink estimates a distinct Fellegi-Sunter model (1.4) via regularized EM within each block, yielding 123 separate estimates for each model parameter. Since fastLink does not make a $U-$correction this model makes some sense, as we would expect the at least some of the $u-$probabilities – which roughly measure the probability that two randomly selected records within a block will agree or partially agree by chance – to vary across blocks. (It is less clear why the $m-$ probabilities, which capture measurement/recording or reporting error, should vary by blocks defined by the first name field.) Figure 3.5 shows histograms of these parameter estimates along with the posterior means from the comparable Bayesian model (labelled "Comparable"). We see the expected variability of fastLink's first name $u-$parameters across blocks. We see similar variability in the $u-$parameter for female, since the distribution of gender varies across fastLink's blocks. There is marked variability in the fastLink estimated $m-$probabilities across blocks, which is more difficult to explain by anything other than sampling variability (even though many of the blocks are large, the proportion of matching pairs is small and fastLink does not borrow information about the $m-$probabilities across blocks).

Constraints on the parameter estimates. fastLink imposes the following constraints on the parameter estimates: $m_{match} \geq m_{partial-match} \geq m_{non-match}$ and $u_{match} \leq u_{partial-match} \leq u_{non-match}$ (Enamorado et al., 2019). It does not appear that this assumption is reasonable in our application. Consider the case where random agreement on a field is relatively common but transcription errors are either uncommon or, more plausibly, result in the a comparison falling into the lowest similarity bucket. In this scenario partial

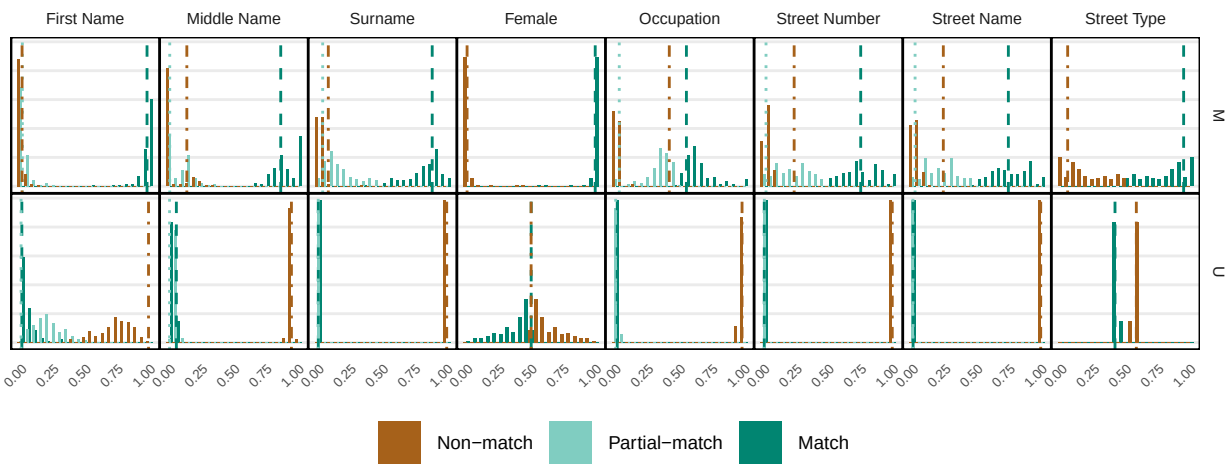


Figure 3.5: Distribution of fastLink parameters across blocks. Vertical lines show posterior means of parameters for the Comparable model.

agreements may be observed less frequently than either matching comparisons among non-matched record pairs (due to random agreement) or non-matching comparisons among matched record pairs (due to parsing or transcription errors). This would violate the constraints imposed by fastLink and result in parameter estimates on the boundary.

Concretely we see these effects in the Bayesian $m-$probability estimates for occupation and the address fields. Conditional on the record pair being a true match, it is most likely that the street name, number and occupation agree, but the next most likely outcome is *disagreement*, not partial agreement. This makes sense, as occupations are often inconsistently recorded in a fashion leading to low similarity scores (a "stevedore" in 1932 might report his occupation as a "dockworker" in 1936) and addresses are subject to seemingly random failures in the OCR and parsing. We might correct some of these effects with better pre-processing (e.g. more intensive standardization of occupations), but both addresses and occupations are subject to change over time. If these changes occur at a higher rate than minor typographical or OCR errors leading to partial agreement then the constraint would still be violated.

Finally, note that as long as partial matches are *relatively* more common among truly matching record pairs than non-matching pairs then an observed partial match will still, other features held constant, indicate that the record pair is more likely to be a true match. Thus, we might expect to observe monotonicity among the *ratios* of the parameters, $m_{match}/u_{match} \geq m_{partial-match}/u_{partial-match} \geq m_{non-match}/u_{non-match}$ when the comparison under consideration is ordinal (but not, in general, among the parameters themselves). In our application we do not impose this constraint although our parameter estimates, with U-correction, satisfy it approximately.

Imposing the one-to-one constraint during or post-estimation.

						ID	Year	First	MI	Last	Female	Occupation	Street Number	Street Name	Street Type
						a_1	1932	william	e	brown	0	carpenter	3200	high	street
						a_2	1932	william	e	brown	0	carpenter	6401	outlook	avenue
						a_3	1932	william	j	broun	0	carpenter	3026	shattuck	avenue
fastLink			Bayesian Model												
a_1	a_2	a_3	a_1	a_2	a_3										
1.00	**1.00**	0.00	0.00	**1.00**	0.00	1936	william	e	brown	0	carpenter	6401	outlook	avenue
1.00	1.00	0.00	0.00	0.00	0.00	1936	william	e	brown	0	title examiner	6000	romany	road
0.97	0.97	0.39	0.00	0.00	0.00	1936	william	—	brown	0	musician	—	—	—
0.47	0.47	0.02	0.00	0.00	0.00	1936	william	—	brown	0	laborer	—	decoto	—
0.21	0.21	0.01	0.00	0.00	0.00	1936	william	h	brown	0	carpenter	—	san lean	—
0.03	0.05	0.00	0.00	0.00	0.00	1936	william	—	brown	0	clerk	25	linda	avenue
0.03	0.05	0.00	0.00	0.00	0.00	1936	william	—	brown	0	clerk	2210	tenth	avenue
0.01	0.01	**1.00**	0.00	0.00	**1.00**	1936	william	j	brown	0	carpenter	3026	shattuck	avenue
0.01	0.01	0.00	**0.64**	0.00	0.00	1936	william	m	brown	0	carpenter	2205	woolsey	street
0.00	0.00	0.99	0.00	0.00	0.00	1936	william	j	brown	0	laundry businese	1037	oakland	avenue
0.00	0.00	0.99	0.00	0.00	0.00	1936	william	j	brown	0	shoe salesman	2200	grant	street
0.00	0.00	0.99	0.00	0.00	0.00	1936	william	j	brown	0	candy maker	2311	fifth	street
0.00	0.00	0.99	0.00	0.00	0.00	1936	william	j	brown	0	laborer	888	fiftysecond	street
0.00	0.00	0.99	0.00	0.00	0.00	1936	william	j	brown	0	clerk	9223	holly	street

Table 3.4: Example comparison of estimated posterior match probabilities from fastLink before deduplication and those estimated by the Bayesian model. Posterior match probabilities in bold indicate record pairs which are selected by the deduplication (fastLink) or contained in the Bayes estimator (Bayesian model).

There are gains to enforcing the one-to-one constraint during estimation rather than post-hoc. For example, consider Table 3.4. The first three rows are "William Brown"s found working as carpenters in 1932. The common names and occupation makes these individuals difficult to link. The remaining rows are records from 1936 found to have non-negligible matching probability to one of these.

The Bayesian model has no problem identifying the two exact matches to a_2 and a_3. Compare this to the fastLink estimated probabilities, which assigns significant matching probability between a_2 the first three 1936 candidates, despite the fact that there is an exact match present and the other two candidates differ on occupation, and one is missing an address and middle name. While fastLink's deduplication procedure makes the correct links here, the estimated matching probabilities are clearly nonsensical. And importantly, if the William Brown on Outlook Ave. had failed to register in 1936 then fastLink would have happily linked him to one of the other two records, as removing a single record pair has almost no influence on the estimated model parameters.

Estimating False Match Rates

We compared the false match rate of fastLink and Bayesian estimators by manually confirming matches declared by one or both methods, blind to which method actually made the match. We pre-registered the design of this comparison [¶]. For this exercise we needed to reduce the full Bayesian posterior over linkage structures to one set of declared links. We used a simple point estimate, classifying any record pair with a posterior match probability of greater than 0.5 as a match. This is the Bayes estimate under squared

[¶]Available at http://egap.org/registration/5452.

	Mover					Non-Mover					Overall				
Stratum	FM	TM	ND	Labeled	Total Matches	FM	TM	ND	Labeled	Total Matches	FM	TM	ND	Labeled	Total Matches
Intersection	2	88	10	100	14,276	4	96	0	100	38,968	6	184	10	200	53,244
Bayesian Only	12	118	20	150	18,525	26	121	3	150	60,562	38	239	23	300	79,087
fastLink Only	102	33	15	150	21,636	105	44	1	150	4,348	207	77	16	300	25,984

Table 3.5: Hand-coding results from mover (left) and non-mover (center) matches and overall (right). Each matched record pair is labeled as either a false match (FM), a true matches (TM) or no determination (ND), when insufficient information is available.

error or balanced misclassification loss functions (Tancredi et al., 2011). For the fastLink estimate we use a conservative threshold of 0.9, which is within the recommended range of 0.75 to 0.95 (Enamorado et al., 2019). These two thresholds are not directly comparable – fastLink's estimated matching probabilities are conditional on the block and are not proper posterior probabilities, since they fail to enforce the one-to-one constraint and often sum to values larger than one. By contrast, the posterior matching probabilities from our Bayesian model are proper probabilities and are not conditional on the indexing scheme due to our U-correction. We begin by comparing the point estimates, which guided our hand labelling, before turning to a more apples-to-apples comparison in the next subsection.

The two estimated sets of matches define three disjoint sets: record pairs classified as a match by both models; Bayesian only matches (record pairs classified as a match by our Bayesian model but not by fastLink); and fastLink only matches (record pairs classified as a match by fastLink but not by our Bayesian model). We further subdivided these sets of matches into two strata: "mover" and "non-mover" matches. We suspected that links made between individuals at different addresses would have higher error rates regardless of method. We defined a matched record pair as a "mover" if the string similarity between the street names was less than 0.85 or the similarity between the street numbers was less than 0.5[‖]. Otherwise the match was classified as a non-mover, including cases where the address information is missing for one or both records.

For both mover and non-mover matches we drew a stratified sample, sampling 100 matches from the intersection stratum and 150 matches from each of the Bayeisan only and fastLink only strata. This yielded 800 total pairs for labelling (400 mover matches and 400 non-mover matches). Each record pair was then labeled as either a false match (FM), a true match (TM), or no determination (ND) (for record pairs where there was not enough information to classify the record pair as either a match or a non-match with a reasonable level of confidence). The labeller was presented with the record pair under question, as well as similar records from each file, but was blind as to which method(s) had made the match. Results of the labeling are shown in Table 3.5.

Our pre-registered comparison excluded matches labelled ND. Removing these pairs we estimated the mover and non-mover false match rates for both our Bayesian model and for fastLink, computing estimates and standard errors using standard methods for stratified samples (Rice, 2006). These appear in Table 3.6.

[‖]These similarity thresholds correspond to a similarity level of less than 5 for street name and less than 3 for street number as listed in Table 3.2.

	ND Excluded			ND as Non-match		
	Mover	Non-Mover	Overall	Mover	Non-Mover	Overall
Bayesian Model	0.062 (0.031, 0.093)	0.123 (0.083, 0.164)	0.108 (0.077, 0.139)	0.173 (0.126, 0.219)	0.133 (0.092, 0.175)	0.143 (0.110, 0.176)
fastLink	0.464 (0.419, 0.509)	0.107 (0.071, 0.142)	0.269 (0.240, 0.297)	0.518 (0.470, 0.565)	0.107 (0.072, 0.142)	0.293 (0.264, 0.322)
Absolute Difference	0.402 (p<0.0001)	0.017 (p = 0.45)	0.161 (p<0.0001)	0.345 (p<0.0001)	0.026 (p = 0.24)	0.150 (p<0.0001)

Table 3.6: Estimated false match rates with 95% confidence intervals excluding ND record pairs (left) and counting ND record pairs as false matches (right) by model.

	ND Excluded			ND as Non-match		
	Mover	Non-Mover	Overall	Mover	Non-Mover	Overall
Intersection	0.022 (0.000, 0.053)	0.040 (0.002, 0.078)	0.035 (0.006, 0.065)	0.120 (0.056, 0.184)	0.040 (0.002, 0.078)	0.061 (0.029, 0.094)
Bayesian Only	0.092 (0.043, 0.142)	0.177 (0.115, 0.239)	0.157 (0.108, 0.206)	0.213 (0.148, 0.279)	0.193 (0.130, 0.257)	0.198 (0.147, 0.249)
fastLink Only	0.756 (0.683, 0.828)	0.705 (0.631, 0.778)	0.747 (0.685, 0.809)	0.780 (0.714, 0.846)	0.707 (0.634, 0.780)	0.768 (0.711, 0.824)

Table 3.7: Estimated false match rates with 95% confidence intervals excluding ND record pairs (left) and counting ND record pairs as non-matches (right) by stratum.

The overall estimated false match rate is 0.11 for the Bayesian model and 0.27 for fastLink, a difference of 0.16 ($p < 0.0001$). The difference is driven primarily by fastLink's high error rate in the mover stratum (a difference of 0.40 $\pm$0.05, $p < 0.0001$). For non-movers, the difference between the two methods was not statistically significant, and the Bayes estimate captured many more matches (Table 3.5).

A more conservative approach would count all the "ND" labelled record pairs as false matches. While this is likely to overestimate the true false match rate, since some ND matches correspond to true matches, it furnishes something of an upper bound for the true false match rate. We repeated the analysis counting ND record pairs as false matches. This gives higher estimated false match rates across the board, but results in the same conclusions as when ND record pairs are excluded from the analysis (Table 3.6).

Examining the estimated false match rates for the individual strata, as shown in Table 3.7, helps to explain the differences in error rates. As expected, the estimated false match rate in the intersection stratum is lower than the Bayesian only or fastLink only strata for both movers and non-movers. However, the false match rate in the fastLink only stratum is much larger than in the Bayesian only stratum. In fact, for both mover and non-mover matches, we estimate that the *majority* of the matches in the fastLink only strata are false matches. In contrast, the estimated overall error rate in the Bayesian only stratum is nearly five times lower than the fastLink only stratum when excluding ND pairs ($p < 0.0001$), and over four times lower when counting ND pairs as true non-matches ($p < 0.0001$).

False Match Rates as a Function of Linked Sample Size

One limitation of the estimates reported in Table 3.7 is that they correspond to only a single threshold for each model, and for reasons described above these two thresholds are not directly comparable. This is a common problem when evaluating record linkage methods; to mitigate this issue Hand and Christen (2018) suggest comparing methods by plotting error rates or other performance metrics as a function of the number of matches made as thresholds vary. Figure 3.6 shows these curves for the overall false match rate and the

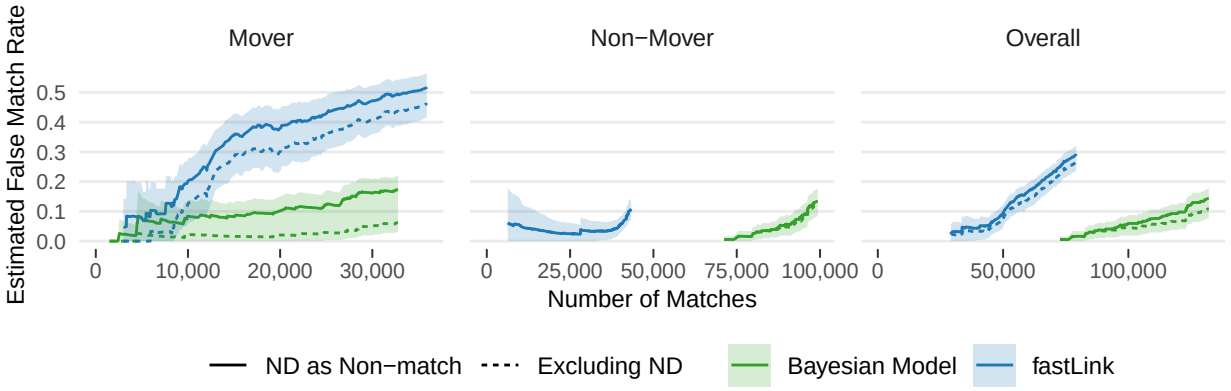


Figure 3.6: Estimated false match rates for mover matches (left), non-mover matches (center), and all matches (right) for deduplicated fastLink (blue) and Bayesian model (green). Solid lines count ND ("no determination") record pairs as true non-matches, dashed lines exclude such pairs. Bands are the union of 95% confidence intervals counting ND pairs as non-match and excluding ND pairs.

false math rate in mover/non-mover strata. Overall the Bayesian method makes significantly more matches than fastLink at any given false match rate. This is due mostly to the higher match rate in the non-mover stratum, although a similar pattern can be observed in the mover stratum.

3.6 Party Switching Rates in Alameda County

Political 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 (Spahn, 2017). Based on cross-sectional data, before the realignment (in 1928) all demographic groups had a Democratic registration rate of about 20% (Spahn, 2017). 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 21 percentage points more likely to be registered as a Democrat in 1936 than in 1932. Among white collar men, the change was just 14 points. In Alameda county, women and men moved about the same amount, each increasing their support for the Democrats by a bit less than 20 percentage points.

However, cross-sectional data can only tell a limited story about party switching – for example, it is unable to disentangle the effects of party switching from differential turnout. A more nuanced picture can be drawn if individuals can be linked over time. Since name prefix and occupation are identified for individuals in the Great Registers, and gender can be inferred with a fairly high degree of confidence, given a set of links it is easy to estimate both the cross-sectional partisan composition and the party-switching rates for individuals registered in both years and aggregate up to groups in order to shed light on these theories.

In this section we consider links generated by our Bayesian model and fastLink. To simplify the presentation of results, voters that were registered as neither Democrat nor Republican (10.4% in 1932 and 7.5% in 1936) in either of the two election years are excluded from the analysis in this section. The largest possible number of links between such individuals is 232,106 (the number of voters with a major party affiliation in the smaller 1932 register). In reality this is a very loose upper bound, as it would only be attainable if everyone in registered in 1932 with a major party affiliation also registered in 1936 in the same county (i.e. no drop-out, death, or out-migration).

Fitting the Bayesian model produced a posterior mean of about 117,000 individuals linked across the two files with major party affiliation in both years. FastLink returned fewer links, identifying around 69,000 record pairs as matches. The composition of the linked sets were also markedly different; for example, both methods linked approximately the same *number* of movers, so the proportion of fastLink's matches that are movers is about twice as high.

Party Switching Rates

We estimated the posterior distribution of party-switching rates from the Bayesian model by computing them for every MCMC sample of C, after discarding the first 2,500 iterations as burn-in. For fastLink we obtain a set of links by thresholding its estimated link probability at 0.9. Enamorado et al. (2019) recommends weighting various estimates computed over linked data by its estimated match probabilities to account for linkage uncertainty. However, when estimating the party switching rate it is necessary to account for uncertainty both in the individual links and in the total number of links. This setting does not seem to fit in the cases considered in Enamorado et al. (2019), and we are unaware of an adjustment that appropriately characterizes linkage uncertainty here using only the output provided by fastLink. Moreover, the fastLink estimated probabilities don't seem to be well-calibrated posterior probabilities (see Table 3.4 for an example), so any adjustment based on them is perhaps questionable. For the purposes of comparing the methods here we present only point estimates using the fastLink matches.

We also considered adjusting point estimates based on the estimated false match rate. We expect erroneous matches to inflate the estimated match rate: Even over this period party switching is not the norm, so false matches are more likely to show a switch in parties than true matches. Considering the distribution of party affiliations in the two files linking two records at random will show a party-switch about half the time. Assuming false matches occur completely at random, the set of estimated matches is composed of a mixture of false matches with proportion π_F, the false match rate, and true matches with proportion $(1 - \pi_F)$, which have a switching rate of ρ_T (our target of inference). The observed switch rate is related to ρ_T by

$$\rho_{observed} = 0.5\pi_F + \rho_T(1 - \pi_F).$$

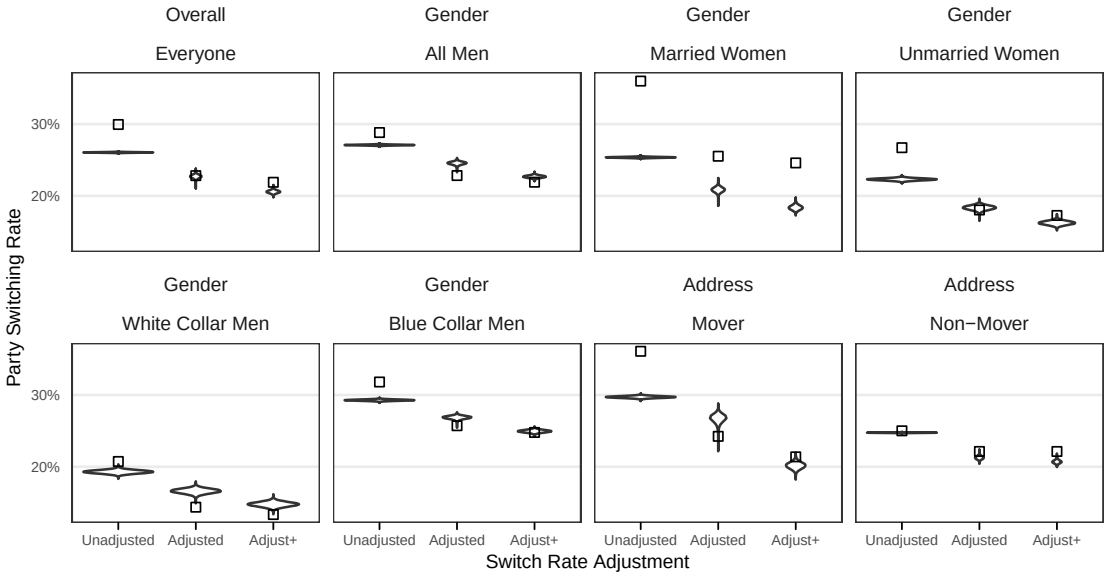


Figure 3.7: Posterior distributions of party switching rate for interesting subgroups across samples of record-pairs for the Bayesian model. The square points show the point estimates from fastLink. The bias-adjusted switch rate ("Adjusted") and the bias-adjusted switch rate treating indeterminate matches as false matches ("Adjust+") are also plotted.

Using this formula and an estimate of the false match rate we can convert the observed switch rate to an estimate of the true switch rate using the relationship

$$\rho_T = \frac{\rho_{observed} - 0.5\pi_F}{(1 - \pi_F)}.$$

Using the labeled data from Section 3.5.1 we estimate false match rates both overall and within subgroups (adjusting the stratum proportions to match those of the subgroups in the latter case). For the Bayesian model we apply this adjustment to each MCMC sample to obtain the posterior of the adjusted switching rates. We consider two estimates of the false match rate: "Adjusted" estimates use a false match rate in which ND labelled pairs are excluded while the "Adjust+" estimates count ND labels as true non-matches (errors)**. We suspect that these error rates under- and over-estimate the true error rates respectively (for reasons discussed in Section 3.5.1), so it seems reasonable to regard them as providing sensitivity bounds for the true party switching rate.

The distribution of the mean party-switching rate overall and for interesting subgroups is displayed in Figure 3.7. With the exception of non-movers, fastLink consistently shows higher unadjusted rates of party-switching than the Bayesian model, save for the overall non-mover stratum where the two are nearly identical.

**In estimating the false match rate for the Bayesian model record pairs falling into none of the strata sampled for labeling, because they were assigned a low posterior link probability by every algorithm, are assumed to have either the maximum estimated false match rate across strata (Adjusted) or a false match rate of 1 (Adjust+).

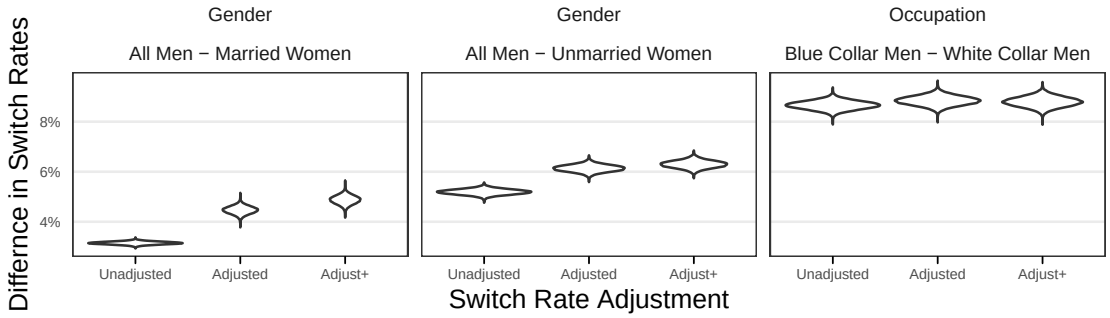


Figure 3.8: Posterior distribution of the difference in mean switch rates between subgroups. Posteriors of the bias-adjusted switch rate ("Adjusted") and the bias-adjusted switch rate treating indeterminate matches as false matches ("Adjust+") are also plotted.

This is consistent with our findings in Section 3.5.1, where fastLink's higher overall false match rates were driven largely by its behavior in the mover stratum.

After adjustment most of the estimates exhibit better agreement between fastLink and the Bayesian model. So why do we prefer the Bayesian model over fastLink? First, the Bayesian method returns a much larger matched set – about 70% larger, or 48,000 links – with lower overall false match rate, and provides uncertainty intervals via the posterior distribution. Second, the estimates are not always brought in line by adjustments, for example in the case of married women in Fig 3.7. It seems reasonable to put more trust in the Bayesian model with its lover overall false match rate here, especially because it estimates a lower switching rate (consistent with making fewer false matches). Third, the adjustment is imperfect. It assumes that false matches occur at random and that the only variation in false match rates by demographic subgroup is due to the varying proportions of movers/non-movers within that group, both of which are likely oversimplifications. Relaxing these assumptions would require at minimum a much more extensive labelling exercise. Therefore we recommend choosing a method with low overall estimated false match rates, and then applying false match rate adjustments as feasible, treating this as a form of sensitivity analysis. We do not consider estimates from fastLink any further.

Demographic differences in party switching rates

To draw out the differences in switch rates based on gender and class we plot the difference in various switch rates in Figure 3.8. The rightmost panel shows a stark difference in party switching rate between blue and white collar men (the rate is estimated at about 8.5%, regardless of bias adjustment). 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 particular fact has been known at the group level (Ladd and Hadley, 1975), but has never been demonstrated with individual-level data.

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. The leftmost panels of Figure 3.8 show that men switched parties at a considerably higher rate than women, which at first glance seems contrary to what one would expect based on Corder & Wolbrecht's analysis. We return to this in the next subsection, where we disaggregate total party switching into flows to and from Democratic party, providing a slightly more nuanced picture.

Relative Party Switching Rates

A high party switching rate does not in and of itself guarantee a large flow from one party to the other, as it could be the case that large numbers of voters are switching parties but the flows roughly cancel in aggregate. Figure 3.9 shows the fraction estimated fraction of party switchers that switch from the Republican party to the Democratic party, indicating that the vast majority of switching was from Republicans to Democrats, as expected.

As in the previous section we adjusted these estimates for potential false matches. Given the share of voters registered to the Democratic and Republican parties in 1932 and 1936, randomly linked record pairs which differ on party would appear to switch from the Republican party to the Democratic party approximately 70% of the time. This proportion is nearly identical when computed separately for men and women. We modify the adjustment introduced in Section 3.6 and compute the adjusted fraction as $\tau_{adj} = (n_{R2D} - 0.7 n_{switch}\pi_F)/n_{switch}(1-\pi_F)$ where n_{R2D} is the observed number of voters switching from the Republican to the Democratic party, n_{switch} is the observed number of voters switching party, and π_F is the false match rate. The false match rates used for the"Adjusted" and "Adjusted+" estimates remain the same.

While Figure 3.8 shows that men switched parties at a considerably higher rate than women we see a more mixed picture in Figure 3.9. While all groups switch to the Democratic party at a much higher rate than the Republican part, the group with the most lopsided switching is married women (followed by blue collar men). Thus, Figure 3.9 offers a picture more consistent with the findings of Corder & Wolbrecht's: Unmarried women appear to favor the Democratic party less than any other group except for white collar men.

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 the same age. The gap is not explained by different base rates of Democratic affiliation – in 1932,

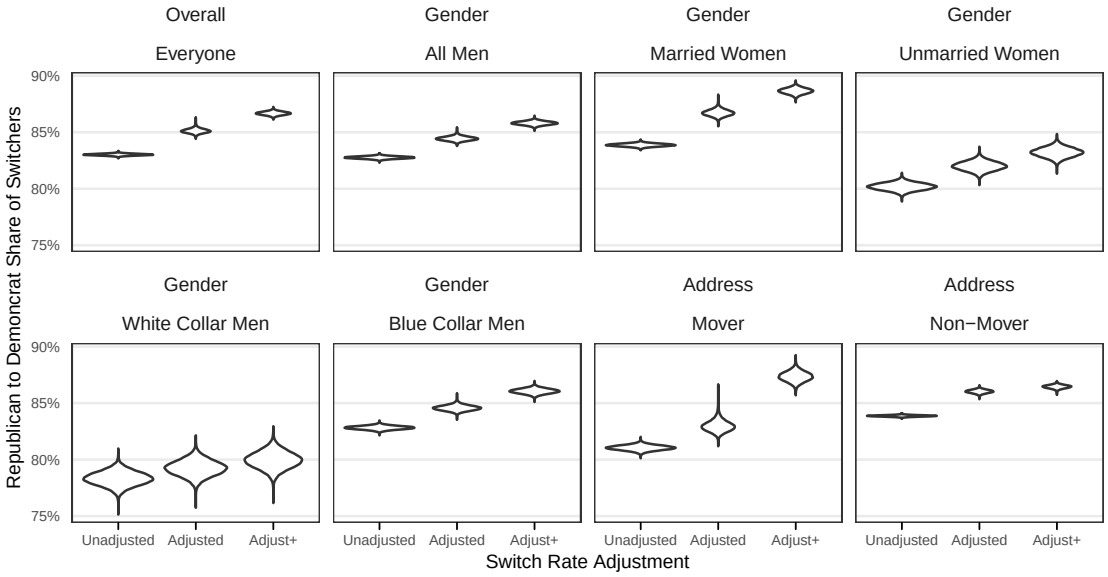


Figure 3.9: Posterior distributions of the fraction of party switchers who switch from the Republican party to the Democratic party. The overwhelming move towards to Democratic party is readily apparent among all subgroups but especially among married women and blue collar men.

the unmarried and married women posterior mean rate of Democratic registration of 26.5% and 26.2%, respectively.

Of course, it may also be the case that our linked sample is somehow less representative for unmarried women than for other groups – in general, over this time period women are more difficult to link than men. However, for the estimands considered here, this differential non-linkage is only consequential if non-linkage *within* the stratum of unmarried women is correlated with party-switching. (This is why we focus on estimating rates rather than totals.) This seems relatively unlikely, although not impossible. Finally, we note that our analysis cannot provide a complete picture of party affiliation for women over this period, as our analysis of women excludes those who marry between 1932 and 1936. The vast majority of these women would have changed their name and moved to a new address, and they are simply unlinkable using only the Great Registers.

To further understand the implications of potential differential false non-match rates, we compared the distribution of key demographics in 1932 and 1936 for the linked subsample to the distribution across the entire register in each year (Section 3.7). The distributions are rather close, particularly for marital status of women and occupation of men. Some observed differences are expected; our linked sample skews slightly male, likely due in part to reasons discussed above. Indeed, while we expect these distributions to be grossly similar, we would not expect all of them to match exactly even if linkage was done perfectly because of differential voter turnout (including drop-out and new registrations). For example, based on the increase in

the total number of voters in 1936 and the increase in Democratic registration over 1932 we might expect the distribution of new registrants in 1936 to skew Democratic, which would tend to make a perfectly linked subsample (all of whom were registered in 1932) proportionally more Republican than the registered population in 1936. In fact, our linked sample skews slightly Republican by three percentage points in 1932 and five in 1936 (possibly due to compound effects of differential turnout and Republicans being on average older, wealthier, and more stable, and therefore easier to link).

3.7 Comparing linked and cross-sectional populations

The lack of ground truth makes determining the representativness of the linked sample challenging to evaluate. In particular, if false non-matches are concentrated among particular subgroups then bias may be introduced into the analysis of the linked data. To examine this possibility we compare the marginal distributions of different subgroups observed in observed record pairs with those in the linked sample. We report marginal distributions for gender and female marital status in Table 3.8. Table 3.9 contains the marginal distributions for occupation and party membership. For the observed records counts and proportions are reported while for the linked sample we report the posterior mean estimated by our Bayesian model.

The tables indicate that there is a high degree of similarity between the observed marginal distributions and those in the linked sample. Table 3.8 shows that in the linked sample men are over represented relative to women and married women are over represented relative to unmarried women. In both cases the findings are unsurprising as the over represented groups, men and married women, would generally be thought easier to link, relative to all women and unmarried women respectively. However, the discrepancy is a slight, with a maximum discrepancy between population shares is 0.03 in Table 3.8. The difference between observed and linked marginal distributions of white collar and blue collar workers is even smaller, just 0.01, as shown in the left panel of Table 3.9. The right panel of Table 3.9 shows the marginal distributions in party membership. The differences between marginal distributions of party membership in the observed and linked samples are somewhat larger, up to 0.05, than in other subgroups. However, we note that from 1932 to 1936 saw both an overall shift towards to Democratic party and a substantial increase in the number of registered voters. By definition any new voters will be excluded from the linked sample since they are not present in the 1932 records. If these new voters are heavily skewed towards the Democratic party, as it likely the case, then we would expected a representative linked sample to contain a higher share of Republican voters than the voter registrations overall. This is exactly what is observed with the observed and linked distributions of party membership for 1932 matching more closely than those for 1936. Thus, while there are minor compositional differences between the observed and linked data in general the share of different subgroups display a surprisingly high level of agreememnt.

Year	Record Count	All Men	All Women
Observed 1932	259,162	0.53	0.47
Observed 1936	288,087	0.51	0.49
Linked 1932	137,640	0.54	0.46
Linked 1936	137,640	0.54	0.46

Year	Record Count	Married Women	Unmarried Women
Observed 1932	122,596	0.82	0.18
Observed 1936	142,034	0.82	0.18
Linked 1932	63,553	0.83	0.17
Linked 1936	63,922	0.84	0.16

Table 3.8: Marginal distribtuion of gender (left) and marital status for women (right) in observed and linked sample. For the linked sample posteriors means from our Bayesian model are reported. Both men and married women are slightly over represented in the linked sample indicting that these records are somewhat easier to link.

Year	Record Count	Blue Collar Men	White Collar Men
Observed 1932	63,619	0.92	0.08
Observed 1936	66,921	0.92	0.08
Linked 1932	35,474	0.91	0.09
Linked 1936	33,501	0.91	0.09

Year	Record Count	Democrat	Republican
Observed 1932	232,106	0.33	0.67
Observed 1936	266,465	0.53	0.47
Linked 1932	125,890	0.30	0.70
Linked 1936	128,508	0.48	0.52

Table 3.9: Marginal distribtuion of male occupation (left) and party (right) in observed and linked sample. For the linked sample posteriors means from our Bayesian model are reported. Both white collar men and republicans are slightly over represented in the linked sample.

3.8 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 by our analysis of the Great Registers.

Linking our extract from the Great Registers provides new insight into party switching over one of the largest political realignments in American history. However, these insights are necessarily limited by the information available on the registration rolls. In particular we are missing important demographics like age and education, and variables like gender, marital status, and occupation type have to be reconstructed from attributes available on the file. It may be possible to obtain a more fine-grained picture of party switching by linking the registers to adjacent Census years to pick up more detailed demographic information about voters, provided these additional links can be made accurately. Similarly, using additional data sources like marriage records to help bridge the gap between election years could reduce error rates for difficult subpopulations, such as women who marry during the "off" years.

Clearly it would be useful to adapt our methods to link multiple files simultaneously, both for linking in the Great Registers and for many other applied problems. The computational challenges inherent in record linkage grow rapidly as we consider linking multiple files; see e.g. Sadinle (2013); Steorts et al. (2015, 2016) for some examples and discussion of model-based deduplication and multiple file linking. Adapting post-hoc blocking to the multiple file setting (and to files with duplicates) is a promising area for future work. We expect that the simplest multiple-file/de-duplication version of post-hoc blocking – stacking the multiple files

and proceeding as in Section 3.1 as though we were matching this "meta" file to itself – should work well, provided adequate post-hoc blocking weights can be constructed.

A second outstanding question is pertains to the construction of a post-hoc blocking scheme. While we have discussed the construction of maximal weights for post-hoc blocking (Section 3.2.1), we have not justified the choice of maximum size of a post-hoc block. We first note that this choice should be made almost entirely on computational grounds, if computational constraints are not binding then post-hoc blocks need not and should not be used. Thus in general the largest (i.e. including the most record pairs) post-hoc blocking scheme for which a posterior distribution can be estimated in a reasonable amount of time should be adopted. Under the locally balanced moves for updating the link structure outlined by Zanella (2019) the cost of performing an update is linear with the number of record pairs included within the post-hoc blocks. This is therefore a reasonable measure of the overall size of the scheme. We further note that in our experience the total runtime of the MCMC algorithm can be extrapolated reasonably well from running a few updates within each post-hoc block. It is thus possible to select a post-hoc blocking scheme (given by defining a maximum component size) and estimate the corresponding runtime of the restricted MCMC sampler with a reasonable degree of accuracy, which can aid in the selection of the blocking scheme. If an automatic stopping rule is to be used for the MCMC sampler then this is somewhat harder to apply and we suggest first running with a post-hoc blocking scheme which includes fewer record pairs. Using results from this scheme can provided a basis for selecting the final set of post-hoc blocks.

Chapter 4

An Informed Prior for Record Linkage with One-to-one Matching

In this chapter we introduce an informative prior for use in Bayesian PRL. Existing priors for Bayesian PRL can incorporate information about the expected number of links, but do not allow for additional information about the expected link structure. In addition to prior information about the expected number of links, our prior allows for the inclusion of expectations about which subregions of the link structure are mostly to contain the estimated links. Specifically, we provide a framework for the incorporation into the prior of one or more pre-specified blocking rules, which can be used to place significantly larger prior link probabilities on the subset of the record pairs contained within the blocking scheme. For example, if the records correspond to individuals, we might expect that two records are far more likely to match (correspond to the same person) if they agree on both first name and last name. In contrast, a record pair which disagrees strongly on first name, even if it matches on many other fields such as address and surname, is unlikely to be a match. Importantly, because such rules can be specified prior to seeing the record comparisons our approach does not violate the consistency of our Bayesian framework. We introduce a sequential approach for constructing such a prior, while enforcing one-to-one matching constraint on the link structure. The approach we outline is general and can easily be applied to incorporate alternative structural constraints on the link structure.

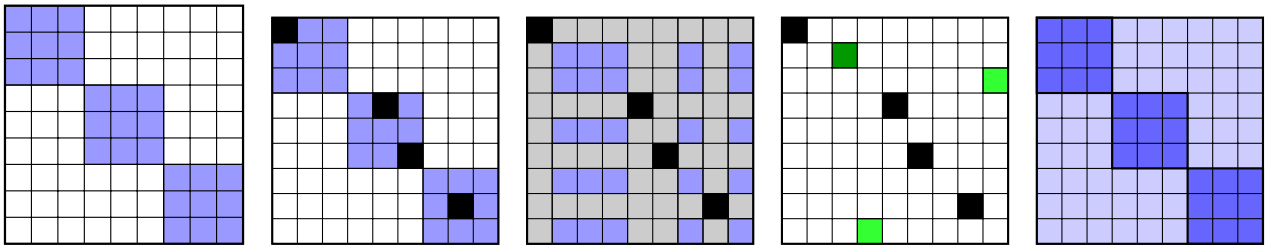
In Section 1.6.1 we introduced several existing priors for Bayesian PRL that enforce one-to-one matching. These priors all have two features in common (1) they are informative on the number of links L and (2) conditional on the number of links L, they place a uniform distribution over the possible values of C which contain exactly L links. As a result, if C and C' are two possible values of the link structure, which contain exactly L links, then $\pi(C) = \pi(C')$ under every prior introduced in Section 1.6.1. The second condition,

that of uniformity, is extremely restrictive and prevents the inclusion of any additional information within the prior.

While prior information can also be incorporated via priors on the matching parameters these prior encode an entirely different type of information, which is not directly informative about the expected link structure. Specifically, they encode prior beliefs about the expected comparisons distributions within the M and U components. Consider an example where we wish to place a higher prior probability of matching on record pairs which display a high level of similarity on both first name and surname relative to other record pairs. We might consider placing a larger prior density on values of the m-parameters that correspond to a high similarity level on the first name and surname comparisons. Such a prior encodes the expectation that matching record pairs are highly likely to show a high level of similarity on comparisons for these fields. This is equivalent to a prior belief that the error rates in these fields are expected to be low. Alternatively, we could place a prior on the u-parameters that assigns a low prior density to parameter values that correspond to agreement on first name and surname within the U component. The information this encodes is that random agreements on these fields are expected to be rare. Neither of these expectations need to be true for it to be the case that most record pairs that show a high level of similarity on both first name and surname correspond to matches. Indeed it is entirely plausible that all three of the following are true: (1) of record pairs which show a high level of agreement on first name and surname a large portion are matches (2) due to higher error rates a relatively large portion of matching record pairs fail to display a high level of similarity on both first name and surname, and (3) random agreements on first and last names are at least somewhat frequent. In order to incorporate the type of prior information given by (1) a method for placing a more informative prior over C is needed.

It is common in record linkage problems to have some prior information on the number of expected matches, or the overlap between the two files. However, a large amount of additional information, particularly about what types of comparisons are likely to correspond to matches is typically also available. Indeed, the basis for applying blocking schemes to record linkage problems is the belief that the blocking scheme will capture all or nearly all of the truly matching record pairs. That is, the prior match probability *conditional* on a record pair being contained within a blocking scheme is significantly higher than for record pairs excluded from the blocking scheme. The construction of such blocks will often depend on the observed comparisons, and therefore incorporating specific blocks into the prior is challenging. However, a method for constructing blocks, or a blocking rule, can easily be defined prior to viewing the comparisons and therefore can be specified in a manner consistent with a Bayesian analysis. It is useful to note that incorporating this type of information leaves the prior invariant to permutation (Zanella, 2019, Supplement B) as a permutation of the records in either file will also permute the block memberships.

Incorporating this type of prior information is appealing from a theoretical perspective but challenging to do in practice. This is particularly true when other constraints, such as one-to-one matching, must also be



(a) Subregion **(b)** Sample for subregion **(c)** Remaining region **(d)** Final links **(e)** Resulting density

Figure 4.1: (a) Shows a set of blocks for which a set of links is initially samples, as shown in (b). Conditioning on the sampled links shown in (b) the region shown in blue in (c) is available for linking in the second stage while the region shown in gray is not, due to the one-to-one matching constraint. In stage two additional links are sampled as shown in (d) with the dark green link denoting a link that could have been drawn in the first stage while the light green links are links which could have only been sampled in the second stage. The overall density from this generating process is shown in (e).

incorporated into the prior. In the next section we outline a general procedure for constructing a generative model which can both enforce a one-to-one matching constraint and place a larger prior match probability on some subregions of C relative to others.

4.1 Generative Model

In constructing our model we take inspiration from a sequential procedure for identifying matching records. In such a sequential approach we might first consider a subset of the overall set of record pairs and attempt to find matches within this subset. Under a one-to-one matching assumption any records matched within the subset could then be removed from further consideration. The remaining record pairs, including those not considered in the initial subset, would then be reconsidered for matching, and additional matches possibly added.

A simple two-stage example of this processes in shown in Figure 4.1. Panel (a) shows a set of blocks within which we draw a set of links consistent with one-to-one matching. These sampled links are shown as black squares is shown in panel (b). Then under a one-to-one match assumption, conditioning on the sampled links, means that any record pairs that contains an already linked record is no longer available for linking. In the link matrix this means that all record pairs contained in a row or column with a link from the first stage are no longer available for linking. This is shown in panel (c), with gray squares corresponding to record pairs which can no longer be linked as a result of the one-to-one matching assumption. The remaining blue squares corresponding to record pairs which can still be linked without violating one-to-one matching. Under a reordering of the record pairs the record pairs shaded in blue would correspond to a 6×5 rectangular subregion. Thus, the remaining records pairs can be treated as a single block, and a second set of links sampled for this region. These links are shown in green in panel (d). The dark green link corresponds

to a record pair that is included in the first stage blocking and thus could have been linked in the first draw but wasn't, while the lighter green links correspond to record pairs which are not within the blocks shown in panel (a). Finally, in panel (e) we shown the overall link densities which would result if this processes were repeated many times. Because the record pairs within the blocks are available for linking in both draws (unless excluded by one-to-one matching) the final link probability is higher (darker blue) in panel (e) than in panel (a) for these record pairs. Conversely, for record pairs outside of the first stage blocks the final link probability is lower (lighter blue) in panel (e) than in panel (c). This dynamic occurs because in many instances they will be unavailable for linking due to the one-to-one matching constraint.

By construction this sampling approach will link record pairs within the first stage blocks with a higher probability than those excluded from the blocks. There is however, no need to limit this processes to only two stages. Multiple sets of blocks can be considered. Within each stage we condition on the links made in earlier stages, excluding already linked records from consideration. We then sample additional links within the blocks corresponding to the current stage. The final set of links is then the union of the links sampled across all stages. Such an approach will tend to place a higher sampling density on links contained in blocks available for linking in earlier stages, and lower sampling densities on record pairs that can only be linked in later stages.

We next define some notation for our multi-stage sampling procedure. To begin, assume that we are given a sequence of T functions $g^{(1)}, \ldots, g^{(T)}$ which will return a subset of the full set of records pairs when applied to the data. That is $g^{(t)}(A, B) \mapsto H^{(t)}$ where $H^{(t)} \subseteq A \times B$, the full set of record pairs for sets of records A and B. Next, let $C^{(t)}$ define the link structure sampled in stage t. In Figure 4.1 the shaded region in panel (a) corresponds to $H^{(1)}$, the links shown in panel (b) to $C^{(1)}$, $H^{(2)}$ is simply the full set of record pairs, and the green links shown in in panel (d) to $C^{(2)}$. For sets of links across multiple stages, let $C^{(i:j)}$ denote the set of links sampled in stages i through j:

$$C^{(i:j)} = \sum_{t=i}^{j} C^{(t)}. \tag{4.1}$$

We define the sets of linked records from A and B for a value as C as:

$$\begin{aligned} A_C+ &= \left\{ a : a \in A, \sum_{b \in B} C_{ab} = 1 \right\} \\ B_C+ &= \left\{ b : b \in B, \sum_{a \in A} C_{ab} = 1 \right\}. \end{aligned} \tag{4.2}$$

We denote the sets of remaining records available for linking in A and B as:

$$A \setminus A_C + \qquad \text{and} \qquad B \setminus B_C + . \tag{4.3}$$

Under a one-to-one matching restriction if a record is linked in an earlier stage it cannot be linked in a subsequent one. Thus, given values for $C^{(1)}, \ldots, C^{(t-1)}$ the records from A and B which are still available for linking in stage t are the sets $A \setminus A_{C^{(0:t-1)}+}$ and $B \setminus B_{C^{(0:t-1)}+}$ respectively. Finally, let $\pi^{(1)}, \ldots, \pi^{(T)}$ be densities that we use to sample $C^{(1)}, \ldots, C^{(T)}$ from $H^{(1)}, \ldots, H^{(T)}$ respectively. Where $\pi^{(t)}$ depends on parameters $\theta^{(t)}$. We can thus write the probability of sampling a specific sequence of link structures $C^{(1)}, \ldots, C^{(T)}$:

$$\pi\left(C^{(1)}, \ldots, C^{(T)} \mid g^{(1)}, \ldots, g^{(T)}, \theta^{(1)}, \ldots, \theta^{(T)}\right) = \prod_{t=1}^{T} \pi^{(t)}\left(C^{(t)} \mid g^{(t)}(A \setminus A_{C^{(0:t-1)}+}, B \setminus B_{C^{(0:t-1)}+}), \theta^{(t)}\right) \tag{4.4}$$

where we take $C^{(0)}$ to be a $n_A \times n_B$ matrix of zeros. In theory a density $\pi(C \mid g^{(1)}, \ldots, g^{(T)}, \theta^{(1)}, \ldots, \theta^{(T)})$ can be derived from (4.4) by marginalizing, (i.e. summing (4.4) over all sets of $C^{(1)}, \ldots, C^{(T)}$ for which $C^{(1:T)} = C$). In practice this is extremely difficult unless T is small. Fortunately, we are generally more interested in the prior probability places on aggregate quantities, such as the total number of link L, or the number of links contained within a specific subset of the record pairs, the distribution of which can be estimated via simulation.

The remaining challenge is to specify a reasonable set of g's and π's . In particular, the subregions specified by the g's must be consistent with the densities defined by the π's. Alternatively, the π's must be able to draw samples from the subregions defined by the g's that are consistent with one-to-one matching and can accommodate the removal of some record pairs as a result of linking in earlier stages. If we restrict the subregions to be sets of disjoint blocks, such as those defined by a traditional blocking scheme, then existing priors for one-to-one matching can be used for such a sampling method. In the next section we derive an informative prior based on the Beta-bipartite prior introduced in Section 1.6.1.

4.1.1 An Informative Prior Under Traditional Blocking

Going forward we assume that all functions g for restricting the set of record pairs to a subregion take the form of tradition blocking. Thus, when applying g to datasets A and B, the generated subregion H will be a set of K disjoint blocks. Where block H_i corresponds to the product set of record pairs $A_i \times B_i$ for sets of records A_i and B_i, subsets of A and B respectively. These sets correspond to the records for which the blocking key takes a specific value. Because the blocks are disjoint it follows that $A_i \cap A_j = \emptyset$ and $B_i \cap B_j = \emptyset$ for $i \neq j$. We let $n_{a_i} = |A_i|$ and $n_{b_i} = |B_i|$. Furthermore we do not require that g map all records to a block and thus, $\sum_{i=1}^{K} n_i \leq n_{a_i}$ and $\sum_{i=1}^{K} n_{b_i} \leq n_B$. For each block H_i, we define $q_i = \min(n_{a_i}, n_{b_i})$ and $r_i = \max(n_{a_i}, n_{b_i})$ and let quantities $Q = \sum_{i=1}^{K} q_i$ and $R = \sum_{i=1}^{K} r_i$.

We next consider sampling the link structure C such that all links are contained within a set of blocks H. If we restrict links to only occurring within blocks then let C_i denote the set of links occurring within the set of record pairs H_i. Since all record pairs outside of the blocks are unavailable for linking we define overall link structure as the union of these subregions and hence $C = \sum_{i=1}^{K} C_i$. With the number of links L defined as $\sum_{i=1}^{K} L_i$ where L_i is the number of links contained within C_i denoted L_i. Equivalently, $L_i = \sum_{ab \in H_i} C_{ab}$. A sample of links for C can then be drawn either independently for each C_i or jointly across all of them simultaneously.

One method sampling from a single block is given by the Beta-bipartite distribution introduced in Section 1.6.1 (Sadinle, 2017). This distribution first places a Beta-binomial distribution over the number of links and then, conditional on the total number of links, draws uniformly from the possible bipartite matchings. This density was defined in (1.15) but for convenience we re-define it for C_i, using the notation introduced aboveL

$$\pi(C_i \mid q_i, r_i, \alpha, \beta) = \frac{(r_i - L_i)!}{r_i!} \frac{\mathrm{B}(L_i + \alpha, q_i - L_i + \beta)}{\mathrm{B}(\alpha, \beta)}. \tag{4.5}$$

Under this distribution $L_i \sim$ Beta-binomial(q_i, α, β).

We present a generalized version of the Beta-bipartite distribution which can be used to define a distribution jointly over blocks. As with the Beta-bipartite prior we first define a link probability $p \sim$Beta(α, β). With, p fixed to the same value for all blocks H_i within H. Then, conditional on p L_i, the number of links within block i, follows a Binomial distribution and the link structure C_i is then drawn uniformly from the possible bipartite matchings with L_i links. The resulting destiny is given by:

$$p(C_1, \ldots, C_K \mid q_1 \ldots, q_K, r_1, \ldots, r_K, \alpha, \beta) = \left(\prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} \right) \frac{B(L + \alpha, Q - L + \beta)}{B(\alpha, \beta)} \tag{4.6}$$

We refer to this as the *Beta-bipartite with Blocking* distribution. Within a Beta-bipartite with Blocking distribution the number of links within each block follows a Beta-binomial distribution, as in Beta-bipartite distribution, but the shared match probability parameter p introduces correlation between the number of links across blocks. The result is that the total number of links L also follows a Beta-binomial distribution with $L \sim$Beta-Binomial(Q, α, β). Thus, it follows that:

$$\mathbb{E}[L] = \frac{\alpha}{\alpha + \beta} Q \text{ and } \mathbb{V}[L] = \frac{\alpha\beta(\alpha + \beta + Q)}{(\alpha + \beta)^2(\alpha + \beta + 1)} Q \tag{4.7}$$

We show a more careful proof this fact in Appendix A.2.

Assuming a Beta-bipartite with Blocking density for each stage we can re-write (4.4) as:

$$\prod_{t=1}^{T}\left(\prod_{i=1}^{K^{(t)}}\frac{\left(r_i^{(t)}-L_i^{(t)}\right)!}{r_i^{(t)}!}\right)\frac{B\left(L^{(t)}+\alpha^{(t)},Q^{(t)}-L^{(t)}+\beta^{(t)}\right)}{B\left(\alpha^{(t)},\beta^{(t)}\right)} \tag{4.8}$$

where $\theta^{(t)}=\{\alpha^{(t)},\beta^{(t)}\}$. We refer to this density as an *Iterated Beta-bipartite with Blocking*. We note that the conditioning of later stages on earlier stages given by (4.5) is contained in the $Q^{(t)}$ and $r_i^{(t)}$ parameters, which are functions of the earlier stages. That is $Q^{(t)}$ and $r_1^{(t)},\ldots,r_{K_t}^{(t)}$ are functions of $C^{(1:t-1)}$.

As with a Beta-bipartite with Blocking density we focus our analysis on the behavior of the number of links L. It is more likely that a researcher will have some expectation for this quantity. It is possible to derive closed form expressions for the distribution of L in some simple scenarios. We consider the simple case of only two blocking rulesL $g^{(1)}$ and $g^{(2)}$, with the further restriction that $g^{(2)}$ simply place all records into a single block. For this scenario it is possible work work out the $\mathbb{E}[L]$ and $\mathbb{V}[L]$ in closed form, which we do in Appendix A.3. Although, due to interactions between successive sets of blocks it is hard to determine the behavior of even this quantity. In general we recommend evaluating the prior distribution over L via simulation as we discuss further in Section 4.4.1.

4.2 MCMC Sampler

Sampling from the Iterate Beta-bipartite with Blocking prior is more complex than for the Beta-bipartite prior employed in Chapter 3 as it requires monitoring a latent *stage* parameter. Tracking this parameter is equivalent to maintaining T separate link structures: $C^{(1)},\ldots,C^{(T)}$ over which the Iterate Beta-bipartite with Blocking distribution is defined. The MCMC sampler employed in Chapter 3 relies heavily on locally balanced updates for the link structure introduced in Zanella (2019). Because the post-hoc blocks are disjoint the extension of locally balanced updates to post-hoc blocks is straightforward, a separate update can simply be performed for each post-hoc block. Because the link structures in each stage are not disjoint, some record pairs will be contained in multiple stages, the application of locally balanced moves to include mixing over the stage parameter is more challenging. We outline a modified version of the Metropolis-within-Gibbs sampler introduced by Zanella (2019) that includes a locally balanced update over the blocks within each stage of C as well as a Gibbs update of the stage of each link within C.

In our model we assume the same prior distributions and record linkage model as in Chapter 3, with the only difference in models being that we now place an Iterated Beta-bipartite with Blocking prior over C. The assumed set of priors are therefore:

$$
\begin{aligned}
\pi(m_j) &\sim \text{Dirchlet}(\alpha_{mj}) \\
\pi(u_j) &\sim \text{Dirchlet}(\alpha_{uj}) \\
\pi(C) &\sim \text{Iterated Beta-bipartie with Blocking}(g^{(1)}, \ldots, g^{(T)}, \alpha_C, \beta_C)
\end{aligned} \tag{4.9}
$$

where α_{mj} and α_{uj} are vectors of length k_j for $j = 1, \ldots, d$. In (4.9) α_C and β_C are vectors of length T defining the parameters of the Beta-bipartite with Blocking distribution for each stage of the Iterated Beta-bipartite with Blocking distribution.

As in Chapter 3 it is possible to update the m and u-parameters using a Gibbs step. Let n_{mji} and n_{uji} be the number of record pairs with agreement level i for feature j within the M and U components respectively*:

$$
\begin{aligned}
n_{mji} &= \sum_{ab} C_{ab} \mathbb{1}(\gamma^j_{ab} = h) \\
n_{uji} &= \sum_{ab} (1 - C_{ab}) \, \mathbb{1}(\gamma^j_{ab} = h).
\end{aligned} \tag{4.10}
$$

We define $n_{mj} = \{n_{mj1}, \ldots, n_{mjk_j}\}$ and $n_{uj} = \{n_{uj1}, \ldots, n_{ujk_j}\}$. Then, as n_{uj} and n_{mj} follow a Multinomial distributions, conditional on C:

$$
\begin{aligned}
m_j \mid \Gamma, C &\sim \text{Dirchlet}\left(\alpha_{mj} + n_{mj}\right) \\
u_j \mid \Gamma, C &\sim \text{Dirchlet}\left(\alpha_{uj} + n_{uj}\right).
\end{aligned} \tag{4.11}
$$

We therefore update these parameters via a draw from the distributions given in (4.11).

4.2.1 Mixing Over Iterated Link Structure

The challenging part of designing an appropriate MCMC sampler is ensuring efficient mixing over the link structure. Under an Iterated Beta-bipartite with Blocking prior this mixing must occur in a way that samples different sets of links within a stage *and* mixes of the stage parameter assigned to individuals links. While in theory a proposal scheme could be designed which proposes joint updates to both the link structure and stage of a link simultaneously updating the parameters $C^{(1)}, \ldots, C^{(T)}$ we have found such schemes to be excessively complicated and challenging to implement. We instead introduce a MCMC sampler which first updates the set of links within a stage, updating $C^{(1)}, \ldots, C^{(T)}$ separately, and then separately updates the stage parameter of the existing links. With the update to the stage parameter allowing links to move from $C^{(t)}$ to $C^{(t')}$ while holding $C^{(1:T)}$ fixed.

Proposing updates to the links within each stage separately creates a problem: the proposed update must not violate the one-to-one matching assumption when links in other stages are considered. We therefore

*If a field is recorded as missing then it is excluded from the count. This is a consistent with a missing at random assumption (Sadinle, 2017)

update the link structure within each stage holding all links assigned in other stages constant. Let $C^{(-t)} = C^{(1:T)} - C^{(t)}$, the set of linked record pairs in all stages *but* t. Then for a block $H_i^{(t)} \in H^{(j)}$ we remove all linked records contained in the block, and define the remaining set:

$$H_i^{(t-)} = \left(A_i^{(t)} \setminus A_{C^{(-t)}+}\right) \times \left(B_i^{(t)} \setminus B_{C^{(-t)}+}\right). \tag{4.12}$$

We then propose an update only to the link structure of $H_i^{(t-)}$, ensuring that the update will not violate our one-to-one matching assumption. While this will exclude records linked in a stage other than t, it will include records linked in stage t and all unlinked records. Relatively efficient mixing over the link structure within a stage t can then be accomplished by using locally balanced proposals (Zanella, 2019) for $H_i^{(t-)}$, as was done in Chapter 3. We perform such an update for each block $H^{(t)}$, and then repeat the processes for each stage. Since all linked record pairs are assigned a stage this ensures that the status of each record pair is considered in at least one update.

While a MCMC sampler relying only on the procedure for updating the link structure described in the previous paragraph, with no additional updates to the stage parameter, will mix over both the link structure and the stage parameter we would not expect the mixing to occur quickly. Consider the updates to the link structure required for a link to switch from stage t to stage t'. The record pair must be deleted from stage t and then, in a subsequent update, added to stage t'. If the record pair corresponds to a very high likelihood link then removing the link from stage t may require many iterations of the MCMC sampler. Such a sampler may therefore be extremely inefficient and slow to mix. We therefore add a step to our sampler which updates the stage of each linked record pair. Because the stage parameter appears in the prior distribution, but not in the likelihood, such an update need consider only the change in prior probability. Importantly, updates to the stage of a link cannot introduce a violation of one-to-one matching and therefore the prior can be evaluated based only on the block sizes and number of linked record pairs. Indeed, we can construct a Gibbs update which samples from the full set of possible stages for which a record pair may be linked. This set is of size at most T for each linked record pair, but may be smaller if a linked record pair is not allowed (included in a blocking scheme) for all T stages.

The effect on the prior density of moving a link one stage later (from stage t to $t+1$) is equivalent to making the following three changes: (1) sampling one fewer link in stage t, (2) sampling one additional link in stage $t+1$, (3) having the sampling in stage $t+1$ occur within a block with one additional row and one additional column relative to the link existing in stage t. By examining (4.8) we can observe that each of these changes results in a single multiplicative term. The ratio between the prior densities for these two terms is thus easily computed (additional details are given in Appendix A.4). Starting by factoring out the density when the link occurs with $t = 1$ (or the earliest stage in which the link is allowed) it is straightforward to construct the required probabilities for the Gibbs sampler. Adding such an update ensures that the MCMC

sampler mixes much more quickly over the latent stage parameter. This does not preclude the introduction of a method for jointly updating the stage parameter along with the link structure but we leave such a sampling scheme as an avenue for future work.

4.2.2 Algorithm

Combining the updates for the link structure, the stage parameter, and the matching parameters yields the sampling algorithm given by Algorithm 3. This formulation also makes it clear that Algorithm 3 is simply a generalization of the sampling procedure used in Chapter 3. In place of the update for each post-hoc block done in Chapter 3, an update is performed on each stage-block pair, while holding links in different stages constant. Then the other parameters, only the matching parameters in Chapter 3, but both the matching parameters and the stage parameter here are updated via individual Gibbs steps, conditional on the link structure.

Algorithm 3 Metropolis-within Gibbs Update

Input: $C^{(1)}, \ldots, C^{(T)}$, m, and u
Output: Updated values $C'^{(1)}, \ldots, C'^{(T)}$, m', and u'

1. Update the set of links contained in C:
2. **for** stage $t = 1$ to T **do**
3. **for all** block $i = 1$ to $K^{(t)}$ **do**
4. Perform a locally balanced update to $H_i^{(t-)}$.
5. **end for**
6. **end for**
7. **for all** $(a, b) \in C^{(1:T)}$ **do**
8. Update the stage of (a, b) via a Gibbs update over the allowed stages.
9. **end for**
10. Update the m and $u-$parameters via a Gibbs update.

While sufficient for fitting a Bayesian model for a moderately sized record linkage problem Algorithm 3 does not scale well to larger PRL problems. To achieve this we must combine Algorithm 3 with the post-hoc blocking approach introduced in Chapter 3. The key to understanding how Algorithm 3 can be modified to account for post-hoc blocking is to observe that adding the restriction that links can occur only within post-hoc blocks means that we need only look at the intersections between the blocking scheme(s) defined by the Iterated Beta-bipartite with Blocking prior and that defined by the post-hoc blocks.

Consider a set of post-hoc blocks $H^{(phb)}$ containing $K^{(phb)}$ distinct blocks. If for stage t of our prior we have defined a set of blocks $H^{(t)}$ then all allowed linked record pairs for stage t must occur within *both* a block $H_i^{(t)}$ from stage t and a post-hoc block $H_j^{(phb)}$ for $i \in \{1, \ldots, K^{(t)}\}$ and $j \in \{1, \ldots, K^{(phb)}\}$. Thus there are, at most $K^{(t)} \times K^{(phb)}$ distinct blocks we must consider. While this number may be large, in practice

$H_i^{(t)} \cap H_j^{(phb)} = \emptyset$ for many pairs of blocks (i, j). In fact the total number of record pairs considered in stage t is upper bounded by the number of record pairs contained within the post-hoc blocks. Since the cost of performing a locally balanced update to the link structure of a block scales linearly with the number of record pairs contained in the block the problem will remain tractable unless a large number of stages, which each contain many record pairs, are defined. Therefore the adoption of the more flexible prior does nothing to reduce the ability of post-hoc blocks to scale Bayesian models for PRL.

Algorithm 4 Metropolis-within Gibbs Update with Post-hoc Blocking

Input: $C^{(1)}, \ldots, C^{(T)}$, m, and u
Output: Updated values $C'^{(1)}, \ldots, C'^{(T)}$, m', and u'

1. Update the set of links contained in C:
2. **for** stage $t = 1$ to T **do**
3. **for** block $i = 1$ to $K^{(t)}$ **do**
4. **for** block $j = 1$ to $K^{(phb)}$ **do**
5. Perform a locally balanced update to $H_i^{(t-)} \bigcap H_j^{(phb)}$.
6. **end for**
7. **end for**
8. **end for**
9. **for all** $(a, b) \in C^{(1:T)}$ **do**
10. Update the stage of (a, b) via a Gibbs update over the allowed stages.
11. **end for**
12. Update the m and $u-$parameters via a Gibbs update.

This allows us to introduce Algorithm 4 for performing an update when post-hoc blocking is employed. In the next section we use this MCMC sampler to re-analyze the data from Alameda county introduced in Chapter 3 under a more informative prior.

4.3 Application

To demonstrate the ability of the Iterated Beta-bipartite with Blocking prior to incorporate relevant prior information we re-analyze the voter data from Alameda county introduced in Chapter 3. We adopt the same specification as our preferred Bayesian model introduced in Chapter 3, varying only the prior over the link structure. Holding the set of comparisons constant between the models means that, unlike with the comparison with the fastLink model, a direct comparison between model parameters is possible. This includes setting identical priors over the matching parameters. Because the prior over C does not impact the estimation of the post-hoc blocks we use estimate these in the same manner as in Chapter 3 as well. We

retain the prior over the link structure, a Beta-bipartite$(1.0, 1.0)$, as our "base prior". We compare these results to those under a Iterated Beta-bipartite with Blocking prior, our "informative prior" .

For the informative prior we construct a two-stage Iterated Beta-bipartite with Blocking distribution. The first stage is defined by selecting record pairs which displayed a level of similarity greater than 0.92 using a Jaro-WInkler string similarity on both first name *and* last name. The resulting record pairs are then treated as edges in a bipartite graph and the resulting connected components used to define the blocks in the first stage of the prior. This processes is similar to the construction of post-hoc blocks, with the exception that the threshold is applied directly to a similarity measure rather than to an estimated weight.

The parameters for the Beta-bipartite distribution with Blocking used in the first stage were $\alpha_1 = 1.0$ and $\beta_1 = 2.0$ while for the second stage we set $\alpha_2 = 1.0$ and $\beta_2 = 1.8$. While the β parameters are both greater than the $\beta = 1.0$ used in our base prior the multiple stages mean that the expected number of links is similar under both the base prior and the informative prior. The parameters in the informative prior were selected to ensure this. This was accomplished by using Monte Carlo draws to estimate the expected number of links under different parameter values for the informative prior.

As in Chapter 3 both algorithms were run for a total of 25,000 iterations of the MCMC sampler. Under the base prior an update was performed for each post-hoc block within each iteration, followed by an update to the matching parameters via a Gibbs step. For the informative prior an update was performed for each post-hoc block for each stage for each MCMC iteration. This is followed by a Gibbs update to the stage of each link as described in Algorithm 4. The total runtime of the MCMC sampler under the base prior was approximately 4 hours. Under the informative prior the runtime, which performed significantly more updates overall, was approximately 6 hours. As in the previous analysis the first 2,500 iterations of the MCMC sampler were removed as burn-in.

4.3.1 Estimated Parameters

We plot the posterior over the number of estimated links in the left panel of Figure 4.2. The estimated posterior distribution of the number of links under the informative prior identifies significantly more links than under the base prior. This is the behavior we would expect if the prior was placing a larger prior density on regions of C that contain record pairs which the model subsequently identifies a large number of high quality matches. Additionally, the right panel of Figure 4.2 shows that this is not only a result of a number of low probability links being assigned somewhat higher posterior probabilities. Instead separation between the dotted line (informative prior) and solid line (base prior) between match thresholds of 0.9 and 0.7 indicates that under the informative prior a higher posterior match probability is assign to record pairs with a broad range of posterior match probabilities under the base prior.

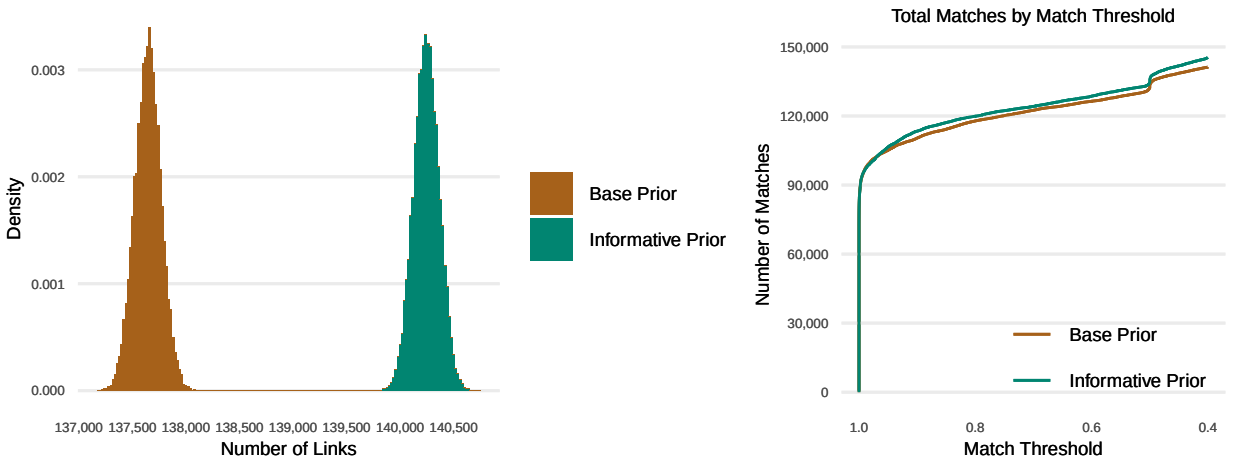


Figure 4.2: Posterior of the number of links (left) under the base and informative priors. The number of links above the posterior match threshold (right) indicates that the shift in the posterior over the number of links is not due simply to an increase among low probability links.

To better understand the effects of the informative prior we examine the change in estimated pairwise posterior link probabilities in Figure 4.3. The left column of Figure 4.3 labeled "Similar Name" corresponds to record pairs contained in the first stage blocks of the informative prior. The right column, "Dissimilar Name" contains the record pairs excluded from the first stage blocks and thus, can only be linked in the second stage. Additionally, Figure 4.3 separates mover record pairs (top row) and non-mover record pairs (bottom row) using the same definition pf mover as in Chapter 3. Given the log color scale of Figure 4.3 the dark red in the bottom left and top right of most of the panels suggests that the estimated pairwise match probabilities display a high degree of similarity under the different prior specifications for the majority of record pairs.

Examining the Dissimilar Name column we note that the estimated match probabilities under the informative prior are generally lower than under the base prior, as we would expect. Although we note the faint line on the diagonal within the non-mover panel indicates that, for a subset of these record pairs, the two models estimate a similar posterior match probability. The agreement of the model under different priors on these record pairs suggests that the posterior match probabilities may be at least partially determined by the one-to-one matching constraint. A simple example of this can be see in the denser region close to the point $(0.5, 0.5)$. This may corresponds to cases where there are two records in one file which closely match either a single record or a pair of records in the other file. In this case the algorithm will be uncertain of the appropriate configuration and will therefore frequently switch back and forth between them, splitting the posterior match probabilities and assigning approximately 50% match probability to each configuration. Under appropriate likelihood ratios a similar dynamic could result in an 80/20 or 70/30 split. This ratio will be only minimally affected by a change in prior if the prior match probability is changed identically for

both record pairs (as it will be for two record pairs in the same name similarity group) yielding the region of density on the diagonal. A shift in prior may however affect the probability on of a no link configuration (i.e. neither possible link is made). Thus, we do not observe the higher density region on the diagonal among the record pairs which are thought to correspond to movers with dissimilar names as are likely to have a substantial posterior probability placed on being non-matches.

Among the Similar Name record pairs, shown in the left column, the picture is somewhat more complicated. We again see a non-trivial density on the diagonal, which again likely corresponds to record pairs where the posterior match probabilities are structurally constrained. We see a large number of mover record pairs (top left panel) with an increased posterior match probability under the informative prior. As noted in our description of Figure 4.3 this appears to occur across a range of posterior probabilities under the base prior. We also see slight curves below the diagonal among both left panels indicating record pairs that are avaiable for linking in the first stage of the informative prior but for which a lower posterior probability is estimated under the informative prior than under the base prior. This result is somewhat surprising but may be due to the shifts in the matching parameter values which are shown in Figure 4.4.

The posterior means of the matching parameters are extremely similar under the two priors as shown in Figure 4.4. The estimated u-parameters are essentially identical, as we might expect given the extremely large number of comparisons contained within the U component. The m-parameters estimates are also extremely similar, with some small shifts: a slight increase in the probability of exact agreement (similarity level 6) on first name and surname under the informative prior, as well as a slight *decrease* in the probability of exact agreement on the three street fields. While the first shift is unsurprising given the prior specification the second change suggests a change in the composition of matches, potentially affecting mover and non-mover matches differently.

4.3.2 Labeling

To evaluate the effect of the informative prior on link accuracy we undertake a hand-labeling exercise similar to that performed in Section 3.5.1. As between we divide record pairs into mover and non-mover categories as the accuracy of the link estimates may differ significantly between these two groups. Within each category we examine, under each prior, the set of links contained in the Bayes estimate of C, this corresponds to all record pairs assigned a posterior link probability of greater than 0.5. We further divide the record pairs into three groups: those classified as link under both priors, the "intersection", those classified as links only under the base prior "base prior only", and those classified as links only under the informative prior "informative prior only". For both mover and non-mover matches we label 100 record pairs each, in the intersection group, and 150 each from the sets of record pairs included in the Bayes estimate of C under only one of the priors. This results in a total of 800 records pairs which are hand-labeled. In addition to the three labels

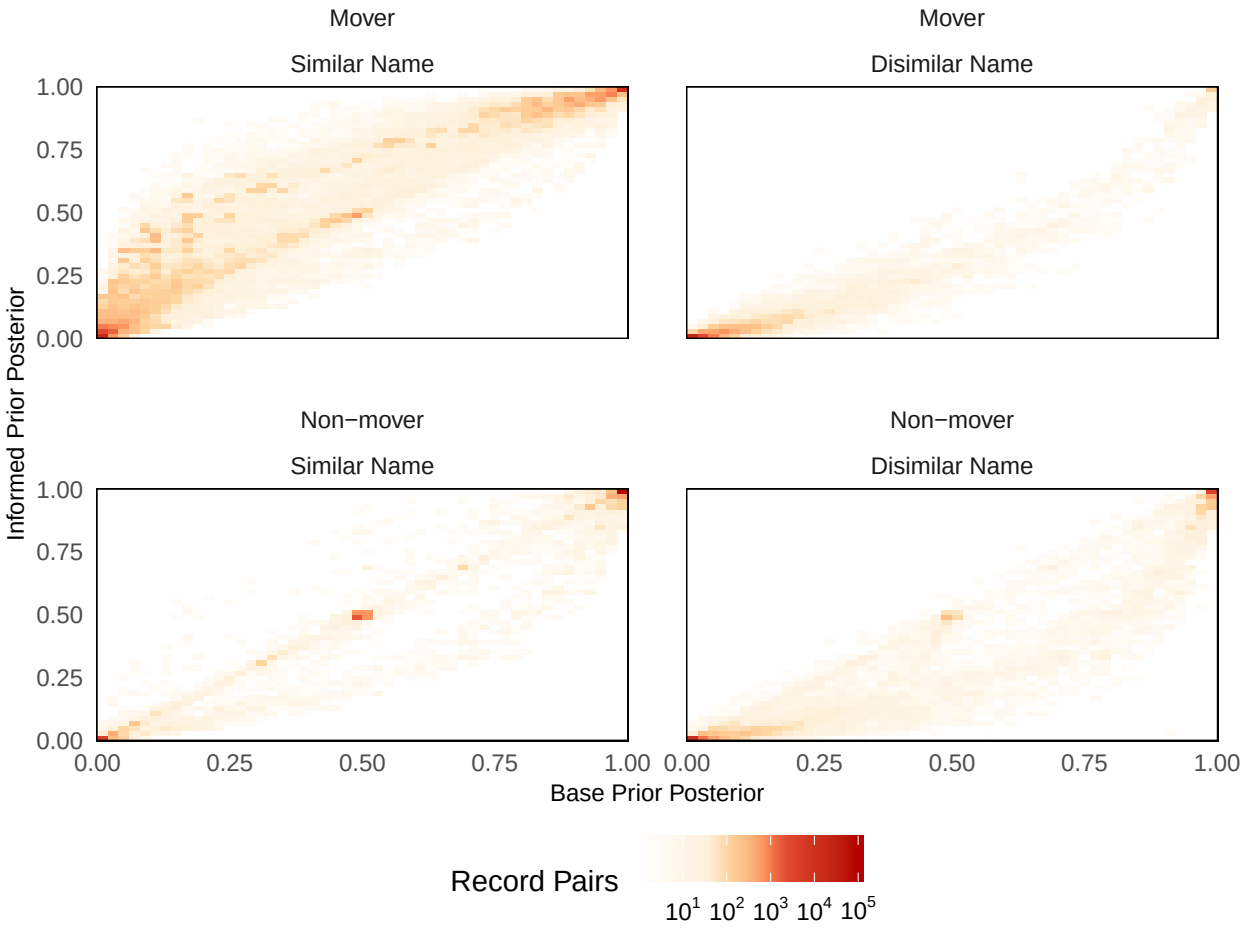


Figure 4.3: Comparison between estimated pairwise posteriors between the base prior (x-axis) and the informative prior (y-axis). The left column (similar name) contains record pairs which are available for linking in the first stage of the informative prior while those in the right column (dissimilar name) are available for linking only in the second stage.

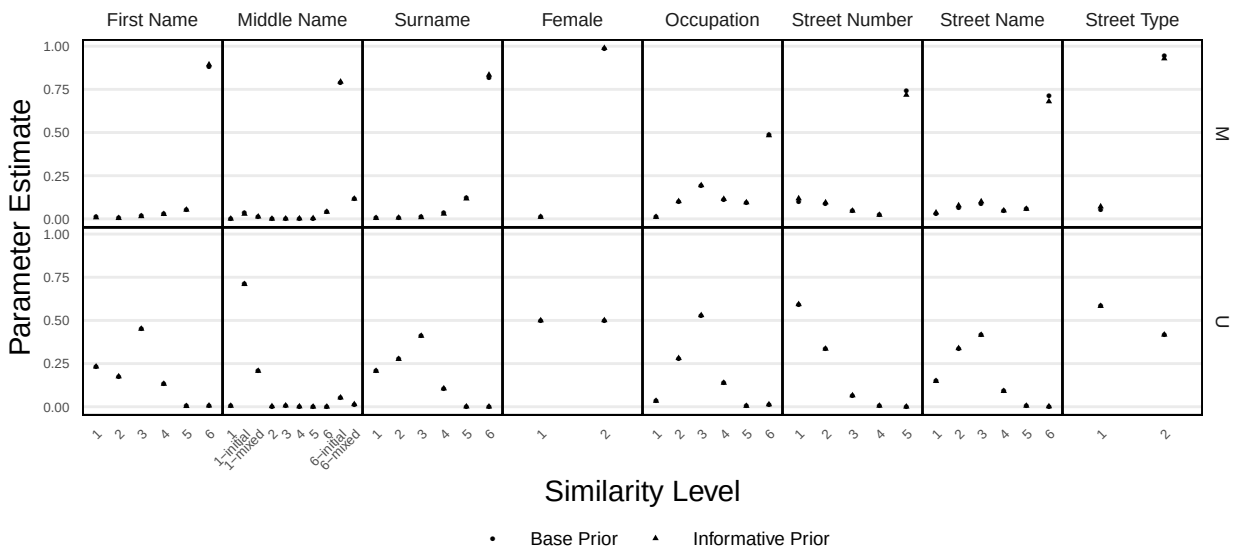


Figure 4.4: Posterior means of estimated matching parameters under base prior and informative prior. There is essentially no difference in the estimated means in the U component and only a very minimal difference within the M component.

which we applied in Section 3.5.1: false match (FM), true match (TM), no determination (ND) we add a fourth label category: duplicate (DU).

While in theory each person should appear at most once in each voter register in practice it appears that in some cases updates to a voter's registration may have been included in the file along with the original registration. This causes records pertaining to a single voter to appear multiple times within the register for a single year[†]. In general these duplicates appear responsible for the relatively large number of record pairs we see with estimated match probabilities very close to 0.5. As in the scenario where there are two perfect, or near perfect, matches for a single record, the posterior probability will generally be split between them. Allowing for a nonzero posterior probability of neither link being correct the true posterior match probabilities(i.e. setting Monte Carlo error to zero) for these record pairs are likely to be just under 0.5. However, because the threshold for inclusion as a link in the Bayes estimate of C is 0.5 this means that a small amount of Monte Carlo error in the pairwise match probabilities for these record pairs can result in a significant fraction of these record pairs being included in the Bayes estimator. In particular, if a pairwise match probability of 0.5 is estimated under both the informative and base priors it may be the case that the estimated probability is just over 0.5 under one prior and just under 0.5 the other. Thus, unless treated appropriately, these record pairs may contribute disproportionately to any perceived differences in Bayes estimates of C under different priors.

[†]This finding was made by Bradley Spahn and Jarred Murray based on an examination of scans of the original paper registers.

The existence of some duplicate records mayb be common in practice, and has been noted in previous work (Jaro, 1989) we are unaware of a generally accepted method for evaluating such record pairs, particularly for assessing method accuracy. Such record pairs correspond to true matches in the sense that both possible record pairs successfully link records referring to the same individual. However, as argued above, these record pairs may be included in the Bayes estimate of C only as a result of Monte Carlo error. This suggests that a conservative approach would classify any duplicate record pairs included in the estimate of C as false matches. Alternatively, duplicate record pairs could simply be excluded from the estimate of the error rates to avoid this issue. In our analysis, presented in Section 4.3.3, we therefore present estimates of the false match rate under all three treatments of duplicate labels (true match, false match, and exclusion).

4.3.3 Labeling Results

The results of the hand-labeling are shown in Table 4.1. Examining the counts in the total columns it is clear that the vast majority of record pairs fall into the intersection group for which the Bayes estimators agree under the different priors. Among mover matches the informative prior finds several thousand more matches, and most of these are labeled as true matches. In contrast, most of the mover matches found only by the base prior are labeled as false matches. A similar pattern is evident among the non-mover matches. Those found only by the informative prior are overwhelmingly labeled as duplicates, suggesting that the source of disagreement between the priors on these record pairs is Monte Carlo error. While a full two-thirds of the non-mover matches found only under the base prior are labeled as false matches.

	Mover						Non-Mover						Overall					
Strata	FM	TM	DU	ND	Labeled	Total	FM	TM	DU	ND	Labeled	Total	FM	TM	DU	ND	Labeled	Total
Intersection	2	87	2	9	100	31,476	3	96	1	0	100	96,741	5	183	3	9	200	128,217
Base Prior Only	85	53	4	8	150	1,321	100	10	37	3	150	2,779	185	63	41	11	300	4,100
Informed Prior Only	14	104	2	30	150	5,710	24	18	102	6	150	929	38	122	104	36	300	6,639

Table 4.1: Hand-coding results from mover (left) and non-mover (center) matches and overall (right). Each matched record pair is labeled as either a false match (FM), a true matches (TM), a duplicate (DU), or no determination (ND), when insufficient information is available.

Overall false match rates under the base and informative priors are shown in Figure 4.2. The left and right sets of columns examine the sensitivity of the results to the treatment of the ND labels, with the ND

		ND Excluded			ND as Non-Match		
Duplicates		Mover	Non-Mover	Overall	Mover	Non-Mover	Overall
Non-Match	Base Prior	0.0674 (0.0269, 0.108)	0.0649 (0.0276, 0.102)	0.0655 (0.0357, 0.095)	0.1508 (0.0875, 0.214)	0.0649 (0.0276, 0.102)	0.0862 (0.0540, 0.118)
	Informed Prior	0.0577 (0.0208, 0.095)	0.0479 (0.0099, 0.086)	0.0506 (0.0213, 0.080)	0.1571 (0.1002, 0.214)	0.0480 (0.0099, 0.086)	0.0781 (0.0464, 0.110)
	Absolute Difference	0.0098 (p = 0.08139)	0.0170 (p<0.0001)	0.0149 (p<0.0001)	0.0063 (p = 0.37357)	0.0170 (p<0.0001)	0.0081 (p<0.0001)
Excluded	Base Prior	0.0464 (0.0166, 0.076)	0.0548 (0.0220, 0.088)	0.0527 (0.0270, 0.079)	0.1334 (0.0733, 0.193)	0.0549 (0.0221, 0.088)	0.0744 (0.0455, 0.103)
	Informed Prior	0.0372 (0.0097, 0.065)	0.0355 (0.0020, 0.069)	0.0359 (0.0105, 0.061)	0.1407 (0.0866, 0.195)	0.0360 (0.0025, 0.069)	0.0648 (0.0364, 0.093)
	Absolute Difference	0.0091 (p = 0.07782)	0.0194 (p<0.0001)	0.0168 (p<0.0001)	0.0073 (p = 0.29786)	0.0190 (p<0.0001)	0.0095 (p<0.0001)
Match	Base Prior	0.0452 (0.0161, 0.074)	0.0482 (0.0156, 0.081)	0.0474 (0.0219, 0.073)	0.1305 (0.0716, 0.189)	0.0483 (0.0158, 0.081)	0.0687 (0.0402, 0.097)
	Informed Prior	0.0365 (0.0095, 0.063)	0.0313 (-0.0018, 0.064)	0.0327 (0.0076, 0.058)	0.1382 (0.0851, 0.191)	0.0316 (-0.0015, 0.065)	0.0610 (0.0329, 0.089)
	Absolute Difference	0.0087 (p = 0.08866)	0.0169 (p<0.0001)	0.0147 (p<0.0001)	0.0076 (p = 0.27052)	0.0167 (p<0.0001)	0.0077 (p<0.0001)

Table 4.2: Estimated false match rates with 95% confidence intervals excluding ND record pairs (left) and counting ND record pairs as false matches (right) by model.

		ND Excluded			ND as Non-Match		
Duplicates		Mover	Non-Mover	Overall	Mover	Non-Mover	Overall
Non-Match	Intersection	0.044 (0.002, 0.086)	0.040 (0.002, 0.078)	0.041 (0.010, 0.072)	0.130 (0.064, 0.196)	0.040 (0.002, 0.078)	0.062 (0.029, 0.095)
	Base Prior Only	0.627 (0.547, 0.706)	0.932 (0.891, 0.973)	0.834 (0.796, 0.871)	0.647 (0.570, 0.723)	0.933 (0.893, 0.973)	0.841 (0.804, 0.878)
	Informed Prior Only	0.133 (0.073, 0.194)	0.875 (0.821, 0.929)	0.237 (0.184, 0.290)	0.307 (0.233, 0.380)	0.880 (0.828, 0.932)	0.387 (0.323, 0.451)
Excluded	Intersection	0.022 (0.000, 0.053)	0.030 (0.000, 0.064)	0.028 (0.002, 0.055)	0.112 (0.050, 0.175)	0.030 (0.000, 0.064)	0.050 (0.021, 0.080)
	Base Prior Only	0.616 (0.535, 0.697)	0.909 (0.855, 0.963)	0.815 (0.770, 0.859)	0.637 (0.559, 0.715)	0.912 (0.859, 0.964)	0.823 (0.780, 0.867)
	Informed Prior Only	0.119 (0.060, 0.177)	0.571 (0.422, 0.721)	0.182 (0.128, 0.236)	0.297 (0.224, 0.371)	0.625 (0.488, 0.762)	0.343 (0.277, 0.409)
Match	Intersection	0.022 (0.000, 0.052)	0.030 (0.000, 0.063)	0.028 (0.002, 0.054)	0.110 (0.049, 0.171)	0.030 (0.000, 0.063)	0.050 (0.020, 0.079)
	Base Prior Only	0.599 (0.518, 0.679)	0.680 (0.605, 0.756)	0.654 (0.597, 0.711)	0.620 (0.542, 0.698)	0.687 (0.612, 0.761)	0.665 (0.609, 0.721)
	Informed Prior Only	0.117 (0.059, 0.174)	0.167 (0.106, 0.228)	0.124 (0.074, 0.174)	0.293 (0.220, 0.366)	0.200 (0.136, 0.264)	0.280 (0.217, 0.344)

Table 4.3: Estimated false match rates with 95% confidence intervals excluding ND record pairs (left) and counting ND record pairs as non-matches (right) by stratum.

labels either excluded from the calculation (left) or counted as false matches (right). While in the sets of rows the treatment of the matches labeled as DU is varied. Across the board the informative prior performs significantly better both overall (combining mover and non-mover matches) and when only the non-mover matches are examined. When the ND labels are excluded from the analysis the estimated false match rate for movers is lower under the informative prior while the base prior performs better when no determination labels are counted as false matches. In neither case is the difference statistically significant at a 95% level of confidence. Counting the ND labels as false matches is likely to be conservative, while excluding them is likely to be anti-conservative. It is therefore reasonable to conclude that performance among the mover matches under the two priors is similar. However, since the informative prior finds additional matches within this group, without significantly increasing the false match rate, we prefer the results under the informative prior for the mover matches as well.

We further examine the error rates within the different labeling strata in Table 4.3. Across all sensitives the intersection strata performs significantly better (has a much lower estimated false match rate) than either the base prior only or informative prior only strata. This result is not unexpected as we would expect this stratum to contained essentially all record pairs which about which it is make a positive link determination with the choice of prior affecting the link decision only in more uncertain cases. Therefore, we focus on comparing the estimated error rates in the base prior only and informative prior only strata. Here we find that the estimated error rate is significantly lower in the informative prior only stratum. There is no overlap in the 95% confidence intervals for the estimated error rate, across nearly all sensitivities. The one exception is among non-mover matches when matches labeled as duplicates are treated as non-matches. In this scenario, particularly conservative for the informative prior only as it finds a much greater share of duplicates which we have argued is primarily due to Monte Carlo error, the informative prior only error rate is still found to be lower but the difference is not statistically significant. This pattern holds regardless of how the no determination labels are treated. Perhaps more importantly, the error rates in the base prior only strata are large in an absolute sense, exceeding 60% in almost all of the sensitivities. In contrast the error rates in the informative prior only strata are typically much lower.

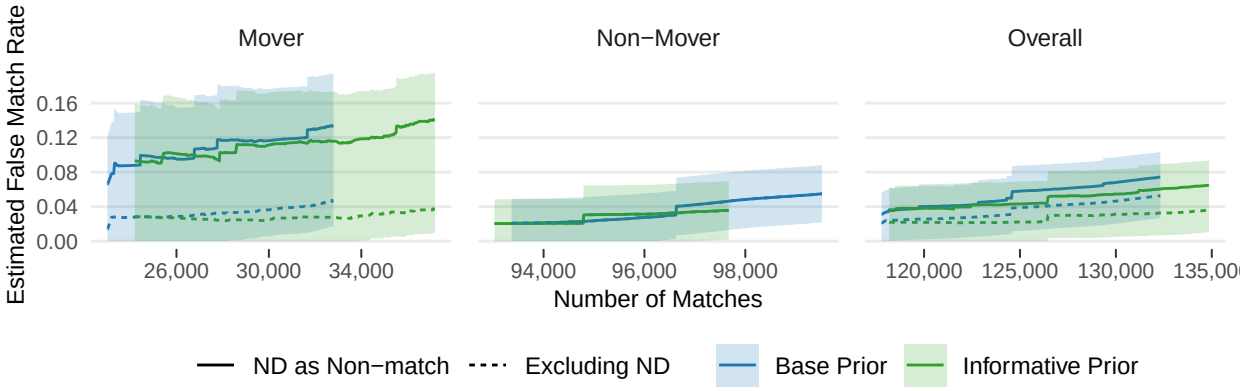


Figure 4.5: Estimated false match rates for mover matches (left), non-mover matches (center), and all matches (right) for deduplicated fastLink (blue) and Bayesian model (green). Solid lines count ND ("no determination") record pairs as true non-matches, dashed lines exclude such pairs. Bands are the union of 95% confidence intervals counting ND pairs as non-match and excluding ND pairs.

As a final comparison of the estimated false match rates under the base and informative priors we plot the total number of record pairs classified as matches against the estimated false match rate as shown in Figure 4.5. The lines shown are produced by classifying all record pairs with a posterior match probability above a given threshold as matches, similar to Figure 4.2, with each threshold corresponding to both a false match rate and a number of matches. To produce Figure 4.5 we vary the match threshold from ≥ 1 to > 0.5. The estimated match rates are nearly identical under the two priors for both mover and non-mover matches. We do however see that the informative prior identifies significantly more mover matches overall, and that at around 32,000 matches it appears that the (more conservative) estimated false match counting ND matches as non-matches under the informative prior is roughly equal to that for the base prior excluding ND matches. In general we can clearly see that among the non-movers substantially more matches are identified under the informative prior without a significant increase in the estimated false match rate, suggesting that many more true matches are identified by the informative prior. Thus, we expected that the false *non-match rate*, which we do not estimate, is significantly lower under the informative prior. We note that overall the uncertainty in the false match rate dominates any difference between the estimates under the two priors. This is not inconsistent with the results in Table 4.2, which show that the informative prior performs significantly better as our labeling scheme was designed to have high power when examining the difference between the false match rate under the two priors.

4.3.4 Party Switching Results

To examine the effects of the informative prior on our downstream estimates we recreate Figures 3.7 and 3.9 from Chapter 3 to compare the estimated party switching rates and party flows respectively. We make the same false match correction described in Section 3.6 to yield our adjusted and adjusted+ estimates of the

of party switching rate as shown in Figure 4.6. It is clear from Figure 4.6 that estimated unadjusted party switching rate under the informative prior is lower across essentially all subgroups. Given that a higher false match rate will tend to inflate the estimated party switching rate this suggests that the false match is lower among the matches found under the informative prior. Interestingly, we see this even among the mover matches, a group for which we estimated that the overall false match rate was similar between the two priors. We focus on the mover and non-mover categories for our examination of the adjusted and adjusted+ switching rates as the estimated false match rates are computed separately for these groups allowing a more straight forward comparison. Within the non-mover category the adjusted and adjusted+ party switching rates are extremely similar. This is consistent with the findings shown in Table 4.1, which shows that among the non-movers very few TM record pairs are found outside of the intersection stratum. We see a somewhat different picture among the non-movers, with the estimated party switching rate under the two priors differing somewhat under both the adjusted and adjusted+ estimates. From Table 4.1 we can see that the informative prior matches nearly 4,000 additional TM record pairs not found by the base prior (the base prior matches less than 500 TM record pairs not found under the informative prior) so it is possible that these differences are due to compositional differences in the matched record pairs. However, if we consider that the adjusted estimate of the party switching rate is probably somewhat anti-conservative, while the adjusted+ estimate is probably conservative then we might expect the correct value to lie somewhere between the two. Under both priors there is substantial overlap in this range so we are hesitant to draw any firm conclusions about differences in the overall estimated party switching rates.

We observe similar shifts in the estimated share of party switches flowing from the Republican party to the Democratic party as shown in Figure 4.7. Across all groups we again see a higher unadjusted rate under the informative prior. Because false matches that appear to be party switches will be relatively evenly split distributed in terms which party they appear to be switching to the inclusion of false matches will tend to make the estimated flow appear closer to 50%. The observed increase in the share flowing to the Democratic party, the party we know most voters switched to, under the informative prior is consistent with the prior identifying fewer false matches. We again observe this shift even among the mover matches, a group for which the informative prior also identifies significantly more matches. Interestingly, we also observe a higher estimated flow under the informative prior among the movers after adjusting for the estimated false match rate. As with Figure 4.6 this is not sufficiently strong statistical evidence to conclude that the estimated share of party switches is higher under the informative prior, even among mover matches, but the results are suggestive.

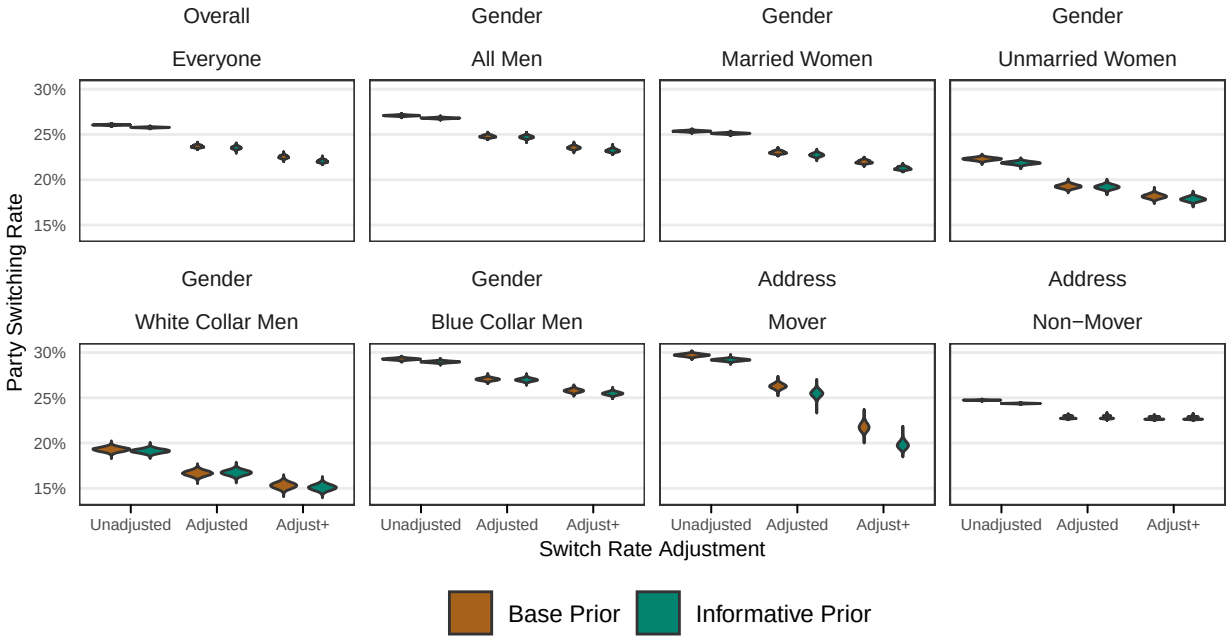


Figure 4.6: Posterior distributions of party switching rate for interesting subgroups across samples of record-pairs for the base and informative priors. The bias-adjusted switch rate ("Adjusted") and the bias-adjusted switch rate treating indeterminate matches as false matches ("Adjust+") are also plotted.

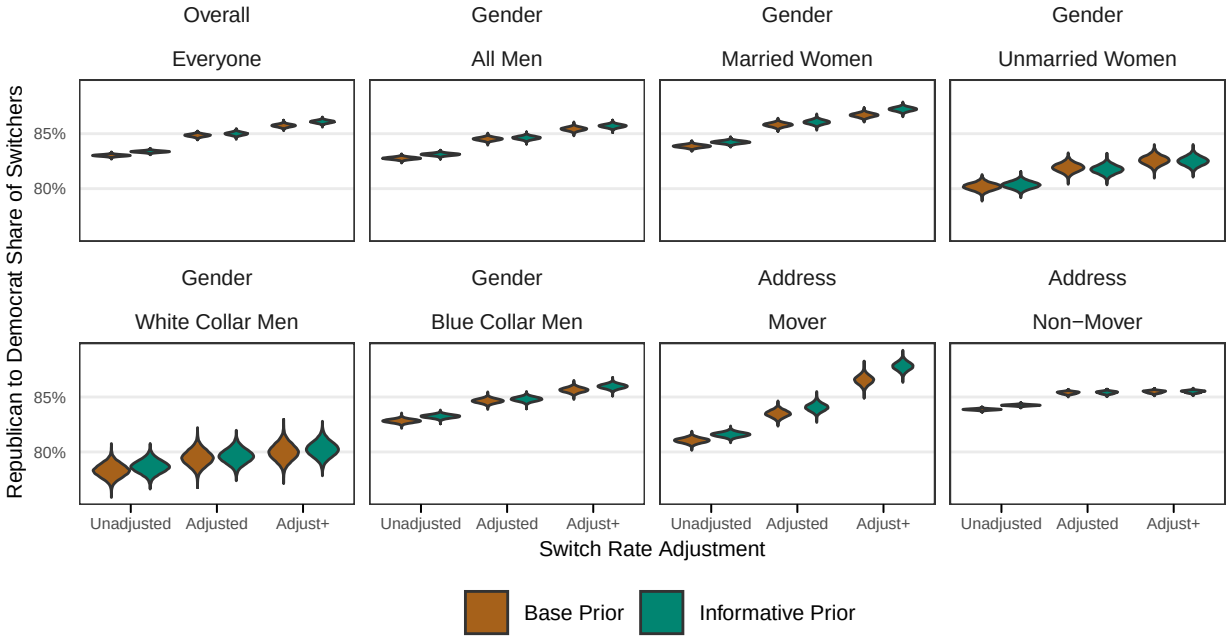


Figure 4.7: Posterior distributions of the fraction of party switchers who switch from the Republican party to the Democratic party. The larger move towards the Democratic party under the informative prior, particularly in the unadjusted estimate suggest that this model identifies a smaller share of false matches.

4.4 Discussion

We have introduced a framework for constructing an informative prior for Bayesian PRL under a one-to-one matching assumption. This contribution is twofold: first we clarify the fact that in many record linkage problems a large amount of prior information as to which fields matching record pairs are expect to show a high level of similarity. Indeed, this is the basis on which the application of blocking methods to PRL has been successful. Importantly, this information is different that simply providing an informative prior over the matching parameters, which model the error rates (m-parameters) and the chance of random agreements (u-parameters). Second, we provide a framework, inspired by a sequential approach to matching, for constructing a prior consistent with one-to-one matching, which can incorporate this information, placing a larger prior match probability on some record pairs relative to others. We derive a specific example of such a prior, the Iterated Beta-bipartite with Blocking distribution. Finally, we show how to integrate this approach with post-hoc blocking, introduced in Chapter 3. This combination makes it feasible to apply priors of this type to large record linkage problems.

We then re-analyze the voter registration data for Alameda county, introduced in Chapter 3. We find that a more informative prior distribution, that places greater density on record pairs which agree on both first name and last name, significantly improves model performance. In particular, we identify a larger number of mover matches without a significant increase in the false match rate among this group. Simultaneously, we significantly reduce the estimated false match rate among estimated non-mover matches. We stress however that our approach for prior construction is general and thus is likely to be easily applied to a variety of other Bayesian record linkage problem. In particular the sequential approach to defining a generative model can easily be applied to construct a prior consistent with constraint on the link structure other than one-to-one matching.

4.4.1 Prior Parameter Selection

While we have introduced a new, more flexible, and therefore more informative prior for Bayesian PRL. However, this additional flexibility comes with increased difficulty of construction. In general we find that it is easiest to understand the densities placed on particular sets of links by sampling from the prior, as done to select the parameters for the prior in Section 4.3. In general we find that the overall distribution of the number of links in each stage is a useful quantity to examine. When considering these quantities it is help to assign links, not to the stage in which the link was sampled, but rather to the earliest stage that the linked record pair could have been linked. This allows the density for links first available for linking in a stage t, a stylized version of which is shown in Panel (e) of Figure 4.1. This may be an easier overall quantity to interpret.

We also provide the expected value and variance of the Iterated Beta-bipartite with Blocking distribution in a simple two-stage setting in Appendix A.3. The generalization of these quantities to additional stages seems straight forward, although it requires some additional assumptions as to the structure of the blocks in each stage. If these assumptions are met then examining these quantities may prove useful as well. We acknowledge that a more straightforward interpretation of these quantities would be highly desirable and this is an area for future work. Of particular use in developing such interpretations would be additional PRL problems about which prior information is available. Through the encoding of additional types of information, via different stage blocking schemes, and observing the effect on the posterior distribution it may be possible to develop a more tractable method of prior construction.

Chapter 5

Conclusions

5.1 Contribution

In this dissertation we have made several significant contributions to the set of available methods estimating PRL models under a one-to-one matching constraint. In Chapter 2 we re-examined the problem of resolving a link structure consistent with one-to-one matching from set of estimated weights. We outlined methods for solving the corresponding LSAPs significantly faster than has been reported in the existing record linkage literature and developed a new penalized likelihood estimator, which performs significantly better than existing methods. In Chapter 3 we introduced post-hoc blocking, a data driven method for constructing a blocking scheme which enables the estimation of an approximate posterior distribution over the link structure using a MCMC algorithm. The use of post-hoc blocks allows this estimation to remain tractable even when large files with noisy fields are being linked. Finally, in Chapter 4 we provided a method for constructing an informative prior under one-to-one matching which maintains invariance under permutations of the records contained within the files. In total we have developed methods for both significantly increasing the set of problems to which Bayesian models for PRL can be applied and significantly improved the performance of existing methods on large problems.

5.1.1 Incorporation of Structure into Assignment problems

We first re-examine the use of an assignment problem to resolve a set of weights into a link structure consistent with one-to-one matching as first suggested by Jaro (1989). We construct a modified version of the approach introduced by Jaro which directly incorporates the FS threshold into the assignment problem. This leads to a sparse assignment problem, for which a solution can be computed at significantly lower computational complexity, and maximizes an objective more closely aligned with the final set of links generated by the Jaro procedure. We suggest the use of Auction algorithms for solving the resulting

assignment problems, and outline several theoretical reasons why these might be expected to perform significantly better than traditional assignment algorithms, such as the Hungarian algorithm. We also provide a simulation study demonstrating the time improvements gained by applying Auction algorithms across a range of PRL scenarios. To our knowledge auction algorithms have not been applied to assignment problems in the record linkage literature previously.

Building on our computational advances we introduce a new penalized likelihood estimator, which uses alternating maximization to find a MAP estimator of the link structure under a prior introduced by Green and Mardia (2006). The penalized likelihood estimator maintains a one-to-one assignment throughout the estimation processes. This contrasts to existing approaches, such as those based on an EM algorithm, which enforce one-to-one matching only after estimating model parameters. The integration of the one-to-one matching assumption into the estimation appears to significantly improve the performance of unsupervised models. Additionally, we show how auction algorithms are uniquely suited to this type of iterative maximization, in which successive reward matrices are closely related.

5.1.2 Post-hoc Blocking

While our penalized likelihood estimator finds a MAP estimate of the link structure under a specific prior, it fails to characterize uncertainty in the link structure, as done by a full posterior distribution. Traditionally, the application of Bayesian methods to PRL has been limited to problems where either an extremely high quality blocking key is available, or the number of records to be linked is small. We introduce post-hoc blocking, a data driven method for constructing high quality blocks, which vastly expands the size and scope of record linkage problems to which Bayesian inference can be successfully applied.

We demonstrate the effectiveness of post-hoc blocking by linking two years, 1932 and 1936, of voter registration data from Alameda county, California. These files, which contain approximately 260,000 and 290,000 records respectively, were constructed by scanning the original paper registers and as a result contain numerous errors. The results yielded by the Bayesian model both identify significantly more matching record pairs and achieve a lower error rate for the identified matches relative to standard models. In addition they provide a straight forward approach to incorporating uncertainty in the estimated link structure into post-linking quantities of interest such as the overall party switching rate.

5.1.3 Informative prior

Our final contribution involves the introduction of an informative prior, which can be used to place a higher prior probability of linking on a subset of the record pairs. Our innovation is to specify into the prior not the specific subregions, but rules for defining such subregions. This allows the prior beliefs to be fully specified before observing the data, while allowing the distribution to appropriately model the observed data.

We derive this distribution from a generative model based on iterative sampling, which places greater link sampling density on regions available for sampling in earlier iterations. While we focus on integrating this with the existing Beta-bipartite distribution, which enforces one-to-one matching, in principle this approach can be combined with any distribution that enforces a one-to-one matching constraint. This approach fully integrates the intuition on which standard practices, such as blocking schemes, are built with a Bayesian modeling approach.

As with our other advances we demonstrate the performance gains from applying our more informative prior on real data covering registered voters from Alameda county, California. We find that applying a prior that places more weight on record pairs which display a high level of similarity on both first and last names significantly improves the performance of the method overall. Encouragingly, the improved performance comes in two ways: First we see a significant fall in the false match rate among record pairs with similar addresses which are classified as matches. Second, we identify a greater number of, harder to link, record pairs with dissimilar address. This increase to the number of matched record pairs occurs without a significant increase in the false match rate, suggesting that more truly matching record pairs are being identified. While a lower error rate is always desirable, it is the identification of additional matches which is perhaps more important. As we show with our party switching rate in Chapter 3, it is possible in many analyses to adjust for false matches if the error rate can be estimated. In contrast, in most record linkage problems a reasonable estimate of the false non-match rate is much harder to obtain as the set of true non-matches is so large that labels are needed for an enormous number of record pairs to estimate the false non-match rate with a reasonable level of confidence.

5.2 Future Directions

Taken together the advances introduced in this dissertation significantly expand the scope of problems to which Bayesian record linkage methods can be applied. Yet there remain significant improvements, many of them straight forward, that can fruitfully continue this trend. These fall into three categories: usability, improvements to existing algorithms, and new problems to which our methodological innovations can be applied.

5.2.1 Usability

Superior statistical performance is often an insufficiently compelling feature for many practitioners to adopt a new method. To be widely adopted a method must also be implement in such a way that it is easy and (relatively) fast to run. As part of this work we have built a package in the Julia programming language (Bezanson et al., 2017) implementing of all methods described. Yet, significant barriers remain to ease of

use, perhaps most importantly the automatic computation of comparison vectors. Wrapper functions that allow these methods to be run via the R and Python programming languages would also significantly expand the scope of potential practitioners.

There also exist several, relatively straight forward, adaptions of the existing methods which could significantly improve the runtimes, allowing for easier use on larger problems. For example, a distributed and parallel implementation of the auction algorithm employed throughout Chapter 2 could significantly reduce the runtime of the procedure while also allowing it to run on problems containing orders of magnitude more record pairs. The biggest theoretical hurdle of such an algorithm, a distributed version of the auction algorithm, is described in (Bertsekas and Tsitsiklis, 1989, 5.4) and need only be implemented. Similarly, it is clear that a straight forward application of parallel or distributed computing to performing locally balanced moves within or across the blocks generated by post-hoc blocking could significantly reduce the runtime of the MCMC sampler. While specific improvements would clearly depend on the details of both the problem and the computing architecture they would be necessary to apply the methods developed to many real-world problems.

Additional runtime improvements could come from modifying parts of the existing algorithms. We have found that while graph clustering can be performed relatively quickly (in $O(n)$ time with respect to the number of edges) performing the clustering many times can be quite costly. Thus, the runtime of Step 4 of Algorithm 2, which repeats Step 2 and Step 3, can be high if the weight threshold for retaining an edge is increased too slowly. This can arise if the procedure performs the graph clustering many times with a threshold too low to separate the problem into many components. This problem could be easily avoided by adopting an agglomerative approach to the clustering, starting with each record pair in its own cluster and then merging clusters. For larger problems such an approach could significantly faster runtimes, although it is unclear if an agglomerative algorithm would result in the same clusters (post-hoc blocks) as Algorithm 2.

A final area where a more efficient algorithm design could yield significant improvements are the restricted MCMC samplers described in Chapter 3 and Algorithm 4. In these samplers we perform a single locally balanced update on each post-hoc block for each update to the matching parameters. Yet mixing is likely to be much slower in the larger post-hoc blocks, it therefore seems sensible to update the link structure within these blocks more frequently than for the smaller blocks. Performing multiple updates to a single post-hoc block without updating the matching parameters can be done in a manner that is significantly more computationally efficient, linear time with respect to the number of records rather than the number of record pairs*. If these, straightforward, improvements to the computational aspects of Bayesian record linkage were to decrease the runtimes by a further order of magnitude it would constitute a significant advance in their practical application.

*This claim is based on a discussion the author had with Giacomo Zanella at the Bayes Comp 2020 conference

5.2.2 Deduplication

In this dissertation we have confined our focus to record linkage under a one-to-one matching constraint. However, it seems likely that pieces of this work are applicable to the other canonical record linkage problem: deduplication. In a deduplication problem a single file may contain multiple records that correspond to the same entity, with the goal being to cluster the records such that each cluster corresponds to a single unique entity. Often deduplication problems involve very large databases, particularly if one considers the case where a "single" file is created by simply stacking multiple files from different data sources.

The deduplication task is fundamentally a clustering problem, with clusters corresponding to each underlying entity. As such it should be possible to extend some of the methods used for constructing post-hoc blocks. For example, given a set of weights, it may be possible to combine our penalized likelihood and post-hoc blocking procedures to develop an estimate of the set of latent entities, or at least a good initialization for an MCMC sampler. As with other clustering problems choosing an appropriate place to cut the dendrogram would be a challenging. In the one-to-one matching setting we incorporate information from the prior to help set this cut-off within the penalized likelihood estimator. It may be possible to do something similar for deduplication problems, but significantly more research would be needed to see if this approach is feasible.

Bibliography

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*. 1

Amini, M. M. (1994). Vectorization of an auction algorithm for linear cost assignment problem. *Computers & industrial engineering*, 26(1):141–149. 38, 39

Andersen, K. (1979). *The creation of a Democratic majority, 1928-1936*. University of Chicago Press. 63

Belin, T. R. and Rubin, D. B. (1995). A method for calibrating false-match rates in record linkage. *Journal of the American Statistical Association*, 90(430):694–707. 25

Bertsekas, D. P. (1992). Auction algorithms for network flow problems: A tutorial introduction. *Computational optimization and applications*, 1(1):7–66. 38

Bertsekas, D. P. (1998). *Network optimization: continuous and discrete models*. Athena Scientific Belmont. 2, 38

Bertsekas, D. P. and Castanon, D. A. (1989). The auction algorithm for the transportation problem. *Annals of Operations Research*, 20(1):67–96. 38

Bertsekas, D. P. and Castanon, D. A. (1992). A forward/reverse auction algorithm for asymmetric assignment problems. *Computational Optimization and Applications*, 1(3):277–297. 28, 38

Bertsekas, D. P., Castanon, D. A., and Tsaknakis, H. (1993). Reverse auction and the solution of inequality constrained assignment problems. *SIAM Journal on Optimization*, 3(2):268–297. 38

Bertsekas, D. P. and Eckstein, J. (1988). Dual coordinate step methods for linear network flow problems. *Mathematical Programming*, 42(1-3):203–243. 38

Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ. 38, 39, 112

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 Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 1417–1425. Curran Associates, Inc. 39

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98. 111

Bijsterbosch, J. and Volgenant, A. (2010). Solving the rectangular assignment problem and applications. *Annals of Operations Research*, 181(1):443–462. 28, 39

Bilenko, M., Kamath, B., and Mooney, R. J. (2006). Adaptive blocking: Learning to scale up record linkage. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 87–96. IEEE. 8

Bilenko, M. and Mooney, R. J. (2003). On evaluation and training-set construction for duplicate detection. In *Proceedings of the KDD-03 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, pages 7–12, Washington, DC. 8

Bourgeois, F. and Lassalle, J.-C. (1971). An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Communications of the ACM*, 14(12):802–804. 28

Carpaneto, G. and Toth, P. (1983). Algorithm for the solution of the assignment problem for sparse matrices. *Computing*, 31(1):83–94. 37

Chambers, R. (2009). *Regression analysis of probability-linked data.* Statistics New Zealand. 24

Charras, C. and Lecroq, T. (2004). *Handbook of Exact String Matching Algorithms.* King's College Publications. 11

Chen, B., Shrivastava, A., and Steorts, R. C. (2018). Unique entity estimation with application to the syrian conflict. *Ann. Appl. Stat.*, 12(2):1039–1067. 1

Chipperfield, J. O., Bishop, G. R., Campbell, P., et al. (2011). Maximum likelihood estimation for contingency tables and logistic regression with incorrectly linked data. *Survey Methodology*, 37(1):13–24. 25

Chipperfield, J. O. and Chambers, R. L. (2015). Using the bootstrap to account for linkage errors when analysing probabilistically linked categorical data. *Journal of Official Statistics*, 31(3):397 – 414. 19, 25, 27

Christen, P. (2008a). Automatic record linkage using seeded nearest neighbour and support vector machine classification. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 151–159. ACM. 17

Christen, P. (2008b). Automatic training example selection for scalable unsupervised record linkage. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 511–518. Springer. 18

Christen, P. (2012a). *Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection*. Springer Science & Business Media. 4, 5, 7, 9, 10, 18

Christen, P. (2012b). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE transactions on knowledge and data engineering*, 24(9):1537–1555. 4

Christen, P. and Pudjijono, A. (2009). Accurate synthetic generation of realistic personal information. *Advances in Knowledge Discovery and Data Mining*, pages 507–514. 46

Christen, P. and Vatsalan, D. (2013). Flexible and extensible generation and corruption of personal data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1165–1168. ACM. 46

Cohen, W., Ravikumar, P., and Fienberg, S. (2003). A comparison of string metrics for matching names and records. In *Kdd workshop on data cleaning and object consolidation*, volume 3, pages 73–78. 10

Cohen, W. W. and Richman, J. (2002). Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 475–480. ACM. 7

Corder, J. K. and Wolbrecht, C. (2016). *Counting Women's Ballots: Female Voters from Suffrage through the New Deal*. Cambridge University Press. 64, 75, 79

Dalzell, N. M., Boyd, G. A., and Reiter, J. P. (2017a). Creating linked datasets for sme energy-assessment evidence-building: Results from the us industrial assessment center program. *Energy Policy*, 111:95–101. 1

Dalzell, N. M. and Reiter, J. P. (2018). Regression modeling and file matching using possibly erroneous matching variables. *Journal of Computational and Graphical Statistics*, 27(4):728–738. 23

Dalzell, N. M., Reiter, J. P., and Boyd, G. (2017b). File matching with faulty continuous matching variables. Working papers, U.S. Census Bureau, Center for Economic Studies. 20

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22. 15, 16

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). 1

Enamorado, T. (2018). Active learning for probabilistic record linkage. *Available at SSRN 3257638*. 18

Enamorado, T., Fifield, B., and Imai, K. (2018). *fastLink: Fast Probabilistic Record Linkage with Missing Data*. R package version 0.5.0. 27, 64

Enamorado, T., Fifield, B., and Imai, K. (2019). Using a probabilistic model to assist merging of large-scale administrative records. *American Political Science Review*, pages 353–371. 6, 14, 19, 27, 64, 67, 69, 70, 73, 76

Erikson, R. S. and Tedin, K. L. (1981). The 1928–1936 partisan realignment: The case for the conversion hypothesis. *American Political Science Review*, 75(04):951–962. 63

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210. 11, 15, 16

Fortini, M., Liseo, B., Nuccitelli, A., and Scanu, M. (2001). On bayesian record linkage. *Research in Official Statistics*, 4(1):185–198. 2, 14, 20, 21, 61

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. 2, 21, 23, 61

Frisoli, K. and Nugent, R. (2018). Exploring the effect of household structure in historical record linkage of early 1900s ireland census records. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 502–509. IEEE. 18

Fu, Z., Christen, P., and Boot, M. (2011). A supervised learning and group linking method for historical census household linkage. In *Proceedings of the Ninth Australasian Data Mining Conference-Volume 121*, pages 153–162. Australian Computer Society, Inc. 17

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. 58

Giang, P. H. (2015). A machine learning approach to create blocking criteria for record linkage. *Health care management science*, 18(1):93–105. 8

Goldstein, H., Harron, K., and Wade, A. (2012). The analysis of record-linked data using multiple imputation with data value priors. *Statistics in medicine*, 31(28):3481–3493. 25

Green, P. J. and Mardia, K. V. (2006). Bayesian alignment using hierarchical models, with applications in protein bioinformatics. *Biometrika*, 93(2):235–254. 22, 23, 35, 41, 49, 53, 110

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. 1, 20, 23, 59

Hand, D. and Christen, P. (2018). A note on using the f-measure for evaluating record linkage algorithms. *Statistics and Computing*, 28(3):539–547. 74

Herzog, T. N., Scheuren, F. J., and Winkler, W. E. (2007). *Data quality and record linkage techniques*. Springer Science & Business Media. 4, 9

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. 24

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. 37

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. 1, 2, 13, 14, 17, 18, 19, 20, 27, 28, 30, 35, 101, 109

Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340. 37

Jonker, R. and Volgenant, T. (1986). Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175. 40

Kejriwal, M. and Miranker, D. P. (2013). An unsupervised algorithm for learning blocking schemes. In *2013 IEEE 13th International Conference on Data Mining*, pages 340–349. IEEE. 8

Kelley, P. (1986). Robustness of the census bureau's record linkage system. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pages 620–624. 13, 16

Kelley, R. (1985). Advances in record linkage methodology: a method for determining the best blocking strategy. *Record Linkage Techniques*, pages 199–203. 4, 7

Kim, G. and Chambers, R. (2012). Regression analysis under probabilistic multi-linkage. *Statistica Neerlandica*, 66(1):64–79. 24

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics (NRL)*, 2(1-2):83–97. 28, 40

Ladd, E. C. and Hadley, C. D. (1975). *Transformations of the American party system: Political coalitions from the New Deal to the 1970s.* WW Norton. 78

Lahiri, P. and Larsen, M. D. (2005). Regression analysis with linked data. *Journal of the American statistical association*, 100(469):222–230. 24

Larsen, M. D. (2002). Comments on hierarchical bayesian record linkage. In *Proceedings of the American Statistical Association, Survey Research Methods Section*, pages 1995–2000. 3, 17

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. 20, 21, 23, 61

Larsen, M. D. (2010). Record linkage modeling in federal statistical databases. In *FCSM Research Conference, Washington, DC. Federal Committee on Statistical Methodology.* 2, 3, 21, 61

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. 1, 15, 18, 53

Lawler, E. L. (1976). *Combinatorial optimization: networks and matroids.* Courier Corporation. 40

Liang, H., Wang, Y., Christen, P., and Gayler, R. (2014). Noise-tolerant approximate blocking for dynamic real-time entity resolution. In Tseng, V. S., Ho, T. B., Zhou, Z.-H., Chen, A. L. P., and Kao, H.-Y., editors, *Advances in Knowledge Discovery and Data Mining*, pages 449–460, Cham. Springer International Publishing. 6

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. 1

Marchant, N. G., Steorts, R. C., Kaplan, A., Rubinstein, B. I., and Elazar, D. N. (2019). d-blink: Distributed end-to-end bayesian entity resolution. *arXiv preprint arXiv:1909.06039.* 6, 20, 23

McCallum, A., Nigam, K., and Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178. Citeseer. 6

Michelson, M. and Knoblock, C. A. (2006). Learning blocking schemes for record linkage. In *AAAI*, volume 6, pages 440–445. 8

Murray, J. S. (2016). Probabilistic record linkage and deduplication after indexing, blocking, and filtering. *Journal of Privacy and Confidentiality*, 7. 5, 17, 60, 66

Neter, J., Maynes, E. S., and Ramanathan, R. (1965). The effect of mismatching on the measurement of response errors. *Journal of the American Statistical Association*, 60(312):1005–1027. 17, 24

Newcombe, H. B. and Kennedy, J. M. (1962). Record linkage: making maximum use of the discriminating power of identifying information. *Commun. ACM*, 5:563–566. 11, 12, 39

Newcombe, H. B., Kennedy, J. M., Axford, S., and James, A. P. (1959). Automatic linkage of vital records. *Science*, 130(3381):954–959. 11

Norpoth, H. (2019). The american voter in 1932: Evidence from a confidential survey. *PS: Political Science & Politics*, 52(1):14–19. 64

Orlin, J. B. and Lee, Y. (1993). Quickmatch–a very fast algorithm for the assignment problem. 37, 38

Ramadan, B. and Christen, P. (2015). Unsupervised blocking key selection for real-time entity resolution. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 574–585. Springer. 8

Rice, J. A. (2006). *Mathematical Statistics and Data Analysis*, chapter 7. Belmont, CA: Duxbury Press, third edition. 73

Rokach, L. and Maimon, O. (2005). *Clustering Methods*, pages 321–352. Springer US, Boston, MA. 6

Sadinle, M. (2013). A bayesian framework for duplicate detection, record linkage, and subsequent inference with linked files. *Under review*. 1, 3, 82

Sadinle, M. (2017). Bayesian estimation of bipartite matchings for record linkage. *Journal of the American Statistical Association*, 112(518):600–612. 1, 2, 14, 16, 19, 21, 23, 46, 49, 50, 58, 60, 61, 90, 92

Samart, K. and Chambers, R. (2014). Linear regression with nested errors using probability-linked data. *Australian & New Zealand Journal of Statistics*, 56(1):27–46. 24

Scheuren, F. and Winkler, W. E. (1993). Regression analysis of data files that are computer matched. *Survey Methodology*, 19(1):39–58. 24

Scheuren, F. and Winkler, W. E. (1997). Regression analysis of data files that are computer matched-part ii. *Survey Methodology*, 23(2):126–138. 24

Smith, M. E. and Newcombe, H. (1975). Methods for computer linkage of hospital admission-separation records into cumulative health histories. *Methods of information in medicine*, 14(03):118–125. 13

Spahn, B. (2017). Before the american voter. *Available at SSRN: https://www.ssrn.com/abstract=3478473*. 63, 75

Steorts, R. C. et al. (2015). Entity resolution with empirically motivated priors. *Bayesian Analysis*, 10(4):849–875. 1, 20, 82

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. 1, 20, 82

Steorts, R. C., Ventura, S. L., Sadinle, M., and Fienberg, S. E. (2014). A comparison of blocking methods for record linkage. In *International Conference on Privacy in Statistical Databases*, pages 253–268. Springer. 4, 6

Sundquist, J. L. (1983). *Dynamics of the party system*. Brookings Institute Washington, DC. 64, 75

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. 20

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. x, 2, 16, 19, 20, 23, 24, 43, 58, 60, 73

Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160. 36, 58

Thibaudeau, Y. (1993). The discrimination power of dependency structures in record linkage. *Survey Methodology*, 19:31–38. 15, 17

Ventura, S. L. and Nugent, R. (2014). Hierarchical linkage clustering with distributions of distances for large-scale record linkage. In *International Conference on Privacy in Statistical Databases*, pages 283–298. Springer. 7, 18

Winkler, W. (1991). Error model for analysis of computer linked files. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pages 472–477. 1

Winkler, W. E. (1985). Exact matching lists of businesses: Blocking, subfield identification, information theory. *Alvey and Kills, editors, Record Linkage Techniques. US Internal Revenue Service*. 13

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*, pages 667–671. 1, 15, 16

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pages 354–359. 9, 14, 69

Winkler, W. E. (1993). Improved decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods, American Statistical Association*, pages 274–279. 1, 15, 16

Winkler, W. E. (1995). *Matching and Record Linkage*, chapter 20, pages 353–384. John Wiley & Sons, Ltd. 15

Winkler, W. E. (2002). Methods for record linkage and bayesian networks. Technical report, Statistical Research Division, US Census Bureau, Washington, DC. 16, 18, 20, 58

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. 1

Xin, S., Yang, H., Xian, W., Ester, M., Bu, J., Wang, Z., and Wang, C. (2018). Mobile access record resolution on large-scale identifier-linkage graphs. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 886–894. ACM. 1

Yancey, W. E. (2005). Evaluating string comparator performance for record linkage. *Statistics*, 5. 9, 10

Zanella, G. (2019). Informed proposals for local mcmc in discrete spaces. *Journal of the American Statistical Association*, 0(0):1–27. 21, 22, 23, 59, 83, 86, 91, 93, 125, 126

Appendix

Appendix A

Technical Details

A.1 MCMC Updates for Locally Balanced Moves

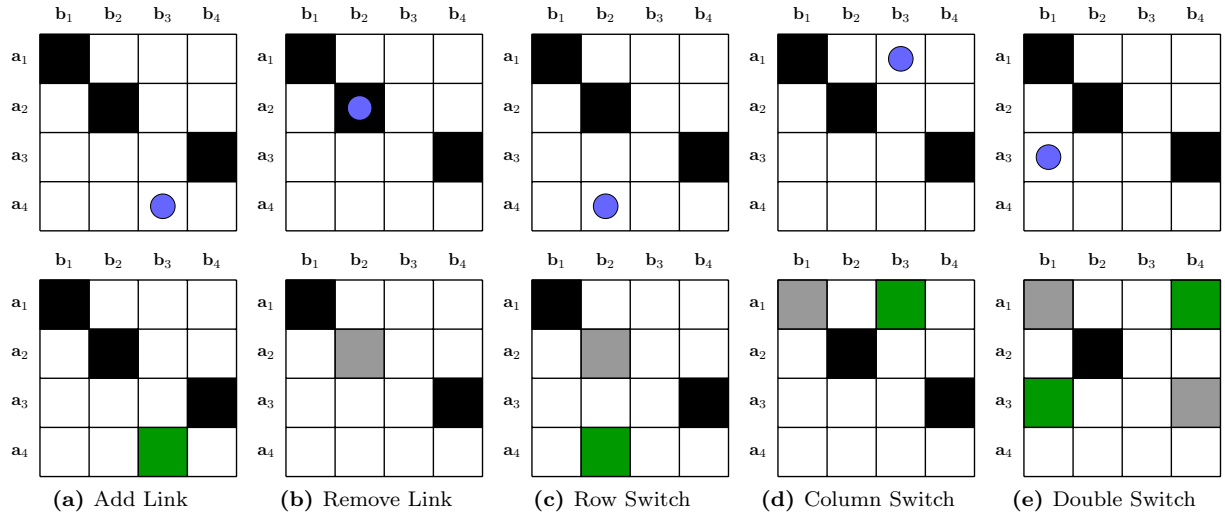
Zanella (2019) introduced "locally balanced moves" which allow informed (leveraging likelihood information) proposals to be made on discrete parameter spaces within a Markov chain Monte Carlo (MCMC) sampler. This is analogous to using gradient information when performing updates for continuous parameter spaces. The incorporation of such information can significantly improve the mixing rate, and therefore efficiency of the overall sampler. Zanella (2019) defines a transition kernel for sampling states C' based on the current state C:

$$Q(C, C') \propto g\left(\frac{\mathcal{L}(C' \mid \Gamma, m, u)\pi(C')}{\mathcal{L}(C \mid \Gamma, m, u)\pi(C)}\right) K(C, C') \tag{A.1}$$

for a balancing function g. The move from C to C', which is proposed with probability $Q(C, C')$ is the accepted with probability:

$$\min\left\{\frac{\mathcal{L}(C' \mid \Gamma, m, u)\pi(C')Q(C', C)}{\mathcal{L}(C \mid \Gamma, m, u)\pi(C)Q(C, C')}, 1\right\}. \tag{A.2}$$

The set of moves considered is controlled by the sampling kernel K, which defines non-zero sampling probabilities to a set of possible moves. Figure A.1 shows the base proposal scheme (K) introduced in Zanella (2019) to constructed the informed proposals. First a record pair is sampled, with each record pair mapping to a specific proposed update. If neither the row (record from A) nor the column (record from A) is linked then the sampled record pair is linked as shown in A.1a. Alternatively, if the sampled record pair is already linked then the link is removed as shown in A.1b. If either the row or column is already linked, but not both, then that link is moved to the sampled record pair as shown in A.1c and A.1d respectively. Finally, if both the row and column are already linked then the assignments are swapped as shown in A.1e. For the doubleswitch move, we note that there is an a second possible record pair (a_1, b_4) which, if sampled,



(a) Add Link **(b)** Remove Link **(c)** Row Switch **(d)** Column Switch **(e)** Double Switch

Figure A.1: Updates to C, the first row shows an existing link structure (links as black squares) with a sampled record pair marked with a blue dot. The second row shows the corresponding update with new links shaded green and removed links shaded grey.

would lead to an identical update. Thus, for an $n_A \times n_B$ set of records with L linked records there are only $n_A n_B - L$ possible updates.

For the balancing function g we employ the Barker balancing function:

$$\frac{t}{1+t}, \tag{A.3}$$

see Zanella (2019) for additional details on choice of balancing functions.

A.2 Derivation of Beta-bipartite with Blocking

Suppose we have K different blocks of size $n_{a_1} \times n_{b_1}, \ldots, n_{a_K} \times n_{b_K}$. As before let $q_k = \min(n_{a_k}, n_{b_k})$ and $r_k = \max(n_{a_k}, n_{b_k})$. Let L_k, the number of links in block k follow a Binomial(q_k, p) distribution. Furthermore, conditional on the value of L_k let C_k, the link structure within block k, follow a uniform

distribution. Then distribution of C, the overall link structure

$$\begin{aligned}
\pi(C \mid p) &= \prod_{i=1}^{K} \binom{q_i}{L_i} p^{L_i}(1-p)^{q_i - L_i} \left(\binom{q_i}{L_i} \binom{r_i}{L_i} L_i \right)^{-1} \\
&= \prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} p^{L_i}(1-p)^{q_i - L_i} \\
&= \left(\prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} \right) p^{\sum_i^K L_i}(1-p)^{\sum_i^K q_i - L_i} \\
&= \left(\prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} \right) p^{L}(1-p)^{Q-L}
\end{aligned}$$

where $L = \sum_{i=1}^{K} L_i$ and $Q = \sum_{i=1}^{K} q_i$. We note that since L is the sum of K independent Binomial distributions with parameter p that $L \sim$Binomial(Q, p). If we let $p \sim Beta(\alpha, \beta)$ then

$$\begin{aligned}
\pi(C) &= \int \pi(C \mid p) \frac{1}{B(\alpha, \beta)} p^{\alpha - 1}(1-p)^{\beta - 1} dp \\
&= \int \left(\prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} \right) p^{L}(1-p)^{Q-L} \frac{1}{B(\alpha, \beta)} p^{\alpha - 1}(1-p)^{\beta - 1} dp \\
&= \left(\prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} \right) \frac{1}{B(\alpha, \beta)} \int p^{L + \alpha - 1}(1-p)^{Q - L + \beta - 1} dp \\
&= \left(\prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} \right) \frac{B(L + \alpha, Q - L + \beta)}{B(\alpha, \beta)}
\end{aligned}$$

where B is the beta function. Abusing notation slightly define the *Block Beta-bipartite* distribution

$$BBB(C \mid L, Q, R, \alpha, \beta) = \left(\prod_{i=1}^{K} \frac{(r_i - L_i)!}{r_i!} \right) \frac{B(L + \alpha, Q - L + \beta)}{B(\alpha, \beta)}$$

A.3 Iterated Blocked Beta-bipartite

Here we considered the specific case of an Iterated Blocked Beta-bipartite distribution with only two stages where the second stage contains all record pairs. That is $g^{(2)}(A, B) = A \times B$. For this reduced case we derive $\mathbb{E}[L]$ and $\mathbb{V}[L]$. Under these assumptions given $L^{(1)}$ the second stage will sample from a $(n_A - L^{(1)}) \times (n_B -$

$L^{(1)}$) block. Thus, conditional on $L^{(1)}$, $L^{(2)}$ ∼Beta-Binomial$(n_A - L^{(1)}, \alpha^{(2)}, \beta^{(2)})$. Therefore

$$\begin{aligned}
\mathbb{E}\left[L^{(2)}\right] &= \mathbb{E}\left[\mathbb{E}\left[L^{(2)} \mid L^{(1)}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}}Q^{(2)}\middle| L^{(1)}\right]\right] \\
&= \mathbb{E}\left[\frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}}\left(n_A - L^{(1)}\right)\right] \\
&= \frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}}\left(n_A - \mathbb{E}\left[L^{(1)}\right]\right) \\
&= \frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}}\left(n_A - \frac{\alpha^{(1)}}{\alpha^{(1)}+\beta^{(1)}}Q^{(1)}\right)
\end{aligned} \tag{A.4}$$

Thus, it follows that

$$\mathbb{E}\left[L\right] = \mathbb{E}\left[L^{(1)} + L^{(2)}\right] = \frac{\alpha^{(1)}}{\alpha^{(1)}+\beta^{(1)}}Q^{(1)} + \frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}}\left(n_A - \frac{\alpha^{(1)}}{\alpha^{(1)}+\beta^{(1)}}Q^{(1)}\right) \tag{A.5}$$

where n_A and $Q^{(1)}$ are known once A and B have been observed.

Under the same assumption set we can work out the variance

$$\begin{aligned}
\mathbb{V}\left[L\right] &= \mathbb{E}\left[\mathbb{V}\left[L \mid L^{(1)}\right]\right] + \mathbb{V}\left[\mathbb{E}\left[L \mid L^{(1)}\right]\right] \\
&= \mathbb{E}\left[\mathbb{V}\left[L^{(1)} + L^{(2)} \mid L^{(1)}\right]\right] + \mathbb{V}\left[\mathbb{E}\left[L^{(1)} + L^{(2)} \mid L^{(1)}\right]\right] \\
&= \mathbb{E}\left[\mathbb{V}\left[L^{(2)} \mid L^{(1)}\right]\right] + \mathbb{V}\left[L^{(1)} + \mathbb{E}\left[L^{(2)} \mid L^{(1)}\right]\right]
\end{aligned} \tag{A.6}$$

We examine each of these terms in turn.

$$\begin{aligned}
\mathbb{E}\left[\mathbb{V}\left[L^{(2)} \mid L^{(1)}\right]\right] &= \mathbb{E}\left[\frac{\alpha^{(2)}\beta^{(2)}\left(\alpha^{(2)}+\beta^{(2)}+n_A-L^{(1)}\right)}{\left(\alpha^{(2)}+\beta^{(2)}\right)^2\left(\alpha^{(2)}+\beta^{(2)}+1\right)}\left(n_A - L^{(1)}\right)\right] \\
&= \frac{\alpha^{(2)}\beta^{(2)}}{\left(\alpha^{(2)}+\beta^{(2)}\right)^2\left(\alpha^{(2)}+\beta^{(2)}+1\right)}\mathbb{E}\left[\left(\alpha^{(2)}+\beta^{(2)}+n_A-L^{(1)}\right)\left(n_A - L^{(1)}\right)\right] \\
&= \frac{\alpha^{(2)}\beta^{(2)}}{\left(\alpha^{(2)}+\beta^{(2)}\right)^2\left(\alpha^{(2)}+\beta^{(2)}+1\right)} \\
&\times \left(\left(\alpha^{(2)}+\beta^{(2)}+n_A\right)n_A - \left(\alpha^{(2)}+\beta^{(2)}+2n_A\right)\mathbb{E}\left[L^{(1)}\right] + \mathbb{E}\left[\left(L^{(1)}\right)^2\right]\right) \\
&= \frac{\alpha^{(2)}\beta^{(2)}}{\left(\alpha^{(2)}+\beta^{(2)}\right)^2\left(\alpha^{(2)}+\beta^{(2)}+1\right)} \\
&\times \left(\left(\alpha^{(2)}+\beta^{(2)}+n_A\right)n_A - \left(\alpha^{(2)}+\beta^{(2)}+2n_A\right)\frac{\alpha^{(1)}}{\alpha^{(1)}+\beta^{(1)}}Q^{(1)} + \frac{Q^{(1)}\alpha^{(1)}\left(Q^{(1)}\left(1+\alpha^{(1)}\right)+\beta^{(1)}\right)}{\left(\alpha^{(1)}+\beta^{(1)}\right)\left(\alpha^{(1)}+\beta^{(1)}+1\right)}\right)
\end{aligned} \tag{A.7}$$

$$
\begin{aligned}
\mathbb{V}\left[L^{(1)} + \mathbb{E}\left[L^{(2)} \mid L^{(1)}\right]\right] &= \mathbb{V}\left[L^{(1)} + \frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}} Q^{(2)}\right] \\
&= \mathbb{V}\left[L^{(1)} + \frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}} \left(n_A - L^{(1)}\right)\right] \\
&= \left(1 - \frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}}\right)^2 \mathbb{V}\left[L^{(1)}\right] \\
&= \left(1 - \frac{\alpha^{(2)}}{\alpha^{(2)}+\beta^{(2)}}\right)^2 \frac{\alpha^{(1)}\beta^{(1)}Q^{(1)}\left(\alpha^{(1)}+\beta^{(1)}+Q^{(1)}\right)}{\left(\alpha^{(1)}+\beta^{(1)}\right)^2\left(\alpha^{(1)}+\beta^{(1)}+1\right)}
\end{aligned}
\tag{A.8}
$$

A.4 MCMC Updates for Informed Prior Prior Ratios

Here we derive the ratio in density between Beta-bipartite with Blocking densities. We first consider the ratio between the same density at two adjacent values for the number of links L and L'. Where by adjacent we mean that the number of links is identical in all blocks except for a single block, within which the number of links differs by one. We then consider the ratio in densities between two Beta-bipartite with Blocking distributions which are identical but contain a single block with either one more or one fewer record in both datasets. Throughout we abbreviate the Beta-bipartite with Blocking density as BBB and make use of the well known identity that for a complex number z if $Re(z) > 0$ then $\Gamma(z+1) = z\Gamma(z)$.

A.4.1 Add Link

Consider the case where $L'_i = L_i$ for $i \neq j$ and $L'_j = L_j + 1$

$$
\begin{aligned}
\frac{BBB(C|L',Q,R,\alpha,\beta)}{BBB(C|L,Q,R,\alpha,\beta)} &= \frac{(r_j - L'_j)!}{(r_j - L_j)!}\frac{B(L'+\alpha, Q-L'+\beta)}{B(L+\alpha, Q-L+\beta)} \\
&= \frac{(r_j - L_j - 1)!}{(r_j - L_j)!}\frac{B(L+1+\alpha, Q-L-1+\beta)}{B(L+\alpha, Q-L+\beta)} \\
&= \frac{1}{r_j - L_j}\frac{\Gamma(L+1+\alpha)\Gamma(Q-L-1+\beta)/\Gamma(Q+\alpha+\beta)}{\Gamma(L+\alpha)\Gamma(Q-L+\beta)/\Gamma(Q+\alpha+\beta)} \\
&= \frac{1}{r_j - L_j}\frac{\Gamma(L+1+\alpha)}{\Gamma(L+\alpha)}\frac{\Gamma(Q-L-1+\beta)}{\Gamma(Q-L+\beta)} \\
&= \frac{1}{r_j - L_j}\frac{L+\alpha}{Q-L-1+\beta}
\end{aligned}
$$

In log scale this reduces to $\log(L+\alpha) - \log(Q-L-1+\beta) - \log(r_j - L_j)$

A.4.2 Remove Link

Consider the case where $L'_i = L_i$ for $i \neq j$ and $L'_j = L_j - 1$

$$\begin{aligned}
\frac{BBB(C|L', Q, R, \alpha, \beta)}{BBB(C|L, Q, R, \alpha, \beta)} &= \frac{(r_j - L'_j)!}{(r_j - L_j)!} \frac{B(L' + \alpha, Q - L' + \beta)}{B(L + \alpha, Q - L + \beta)} \\
&= \frac{(r_j - L_j + 1)!}{(r_j - L_j)!} \frac{B(L - 1 + \alpha, Q - L + 1 + \beta)}{B(L + \alpha, Q - L + \beta)} \\
&= (r_j - L_j + 1) \frac{\Gamma(L - 1 + \alpha)\Gamma(Q - L + 1 + \beta)/\Gamma(Q + \alpha + \beta)}{\Gamma(L + \alpha)\Gamma(Q - L + \beta)/\Gamma(Q + \alpha + \beta)} \\
&= (r_j - L_j + 1) \frac{\Gamma(L - 1 + \alpha)}{\Gamma(L + \alpha)} \frac{\Gamma(Q - L + 1 + \beta)}{\Gamma(Q - L + \beta)} \\
&= (r_j - L_j + 1) \frac{Q - L + 1 + \beta}{L - 1 + \alpha}
\end{aligned}$$

In log scale this reduces to $\log(Q - L + 1 + \beta) - \log(L - 1 + \alpha) + \log(r_j - L_j + 1)$

A.4.3 Increase block size

Consider the case where $q'_i = q_i$, $r'_i = r_i$ for $i \neq j$ and $q'_j = q_j + 1$, $r'_j = r_j + 1$

$$\begin{aligned}
\frac{BBB(C|L, Q', R', \alpha, \beta)}{BBB(C|L, Q, R, \alpha, \beta)} &= \frac{(r'_j - L_j)!}{(r_j - L_j)!} \frac{r_j!}{r'_j!} \frac{B(L + \alpha, Q' - L + \beta)}{B(L + \alpha, Q - L + \beta)} \\
&= \frac{(r_j - L_j + 1)!}{(r_j - L_j)!} \frac{(r_j)!}{r_j + 1!} \frac{B(L + \alpha, Q + 1 - L + \beta)}{B(L + \alpha, Q - L + \beta)} \\
&= \frac{r_j - L_j + 1}{r_j + 1} \frac{\Gamma(L + \alpha)\Gamma(Q + 1 - L + \beta)/\Gamma(Q + \alpha + \beta + 1)}{\Gamma(L + \alpha)\Gamma(Q - L + \beta)/\Gamma(Q + \alpha + \beta)} \\
&= \frac{r_j - L_j + 1}{r_j + 1} \frac{\Gamma(Q + 1 - L + \beta)}{\Gamma(Q - L + \beta)} \frac{\Gamma(Q + \alpha + \beta)}{\Gamma(Q + \alpha + \beta + 1)} \\
&= \frac{r_j - L_j + 1}{r_j + 1} \frac{Q + 1 - L + \beta}{Q + \alpha + \beta}
\end{aligned}$$

In log scale $\log(r_j - L_j + 1) - \log(r_j + 1) + \log(Q + 1 - L + \beta) - \log(Q + \alpha + \beta)$

A.4.4 Decrease block size

Consider the case where $q'_i = q_i$, $r'_i = r_i$ for $i \neq j$ and $q'_j = q_j - 1$, $r'_j = r_j - 1$

$$\begin{aligned}
\frac{BBB(C|L, Q', R', \alpha, \beta)}{BBB(C|L, Q, R, \alpha, \beta)} &= \frac{(r'_j - L_j)!}{(r_j - L_j)!} \frac{r_j!}{r'_j!} \frac{B(L+\alpha, Q'-L+\beta)}{B(L+\alpha, Q-L+\beta)} \\
&= \frac{(r_j - L_j - 1)!}{(r_j - L_j)!} \frac{r_j!}{(r_j - 1)!} \frac{B(L+\alpha, Q-1-L+\beta)}{B(L+\alpha, Q-L+\beta)} \\
&= \frac{r_j}{r_j - L_j} \frac{\Gamma(L+\alpha)\Gamma(Q-1-L+\beta)/\Gamma(Q-1+\alpha+\beta)}{\Gamma(L+\alpha)\Gamma(Q-L+\beta)/\Gamma(Q+\alpha+\beta)} \\
&= \frac{r_j}{r_j - L_j} \frac{\Gamma(Q-1-L+\beta)}{\Gamma(Q-L+\beta)} \frac{\Gamma(Q+\alpha+\beta)}{\Gamma(Q-1+\alpha+\beta)} \\
&= \frac{r_j}{r_j - L_j} \frac{Q-1+\alpha+\beta}{Q-1-L+\beta}
\end{aligned}$$

In log scale $\log(r_j) - \log(r_j - L_j) + \log(Q - 1 + \alpha + \beta) - \log(Q - 1 - L + \beta)$