Bayesian Matching with Relaxed Duplication Assumptions through Dirchlet Record Linkage

Brian Kundinger

July 25, 2024

Abstract

Probabilistic record linkage is the use of statistical methods to identify unique entities within and across databases. Bayesian methods can adopt different prior distributions for various assumptions on the nature of record duplications in the linkage task, but become computationally expensive as the size of the linkage task grows. In this paper, we propose Dirichlet Record Linkage (`DRL`), a method for linking duplicate-free reference file to target file that may have internal duplications. We use a Dirichlet prior for the parameter governing the number of matches a record in the reference file has in the target file. We decompose this parameter into a sequence of conditional probabilities, and use these parameters in a computationally efficient Gibbs sampler to conduct the linkage task. We demonstrate the speed and accuracy of our method relative to other recent Bayesian methods through simulations and case studies. In particular, we show that `DRL` exhibits strong performance using default, noninformative hyperparameters.

1 INTRODUCTION

Probabilistic record linkage is the use of statistical methods to identify unique entities within and across databases, generally without the use of unique identifiers. This is an increasingly important task in "data cleaning," either for its own sake, or as a preliminary step before conducting subsequent analysis. These techniques are used in government, non-profit organizations, healthcare, industry, and their use continues to grow. (e.g., **????**). In this paper, we propose a method for conducting record linkage between a duplicate free reference file and some other file that may have internal duplications.

Many probabilistic record linkage methods are based on the foundational work of Fellegi & Sunter (1969). In these Fellegi-Sunter (FS) methods, the data analyst first creates comparison vectors for each pair of records in the data files. These vectors indicate how similar the records are on a set of variables measured in both files, known as the linkage variables. Using these comparison vectors, the analyst classifies each pair as a match or nonmatch using a likelihood ratio test. An alternative paradigm is to model the linkage variables directly (e.g., Tancredi et al. 2011, Steorts et al. 2016, Marchant et al. 2021, Betancourt et al. 2022). In this article, we build on the contributions to the comparison vector approach.

In the original FS formulation, record pairs are independently classified as matches or nonmatches, but through a Bayesian framework, practitioners can formalize different assumptions on the type of matchings between files through different prior distributions on the linkage structure. For example, Sadinle (2017) proposed prior distribution on one-to-one matchings between two files, and Aleshin-Guendel & Sadinle (2023) proposed a prior on the partitions of records corresponding to unique entities found in arbitrarily many files. These more tailored record linkage methods have the advantage of producing coherent sets of matches that respect transitivity without the need for post-processing, but have limited scalability.

Much work has been done to expand the the scalability of FS methods. One approach is to reduce the number of comparisons vectors needed for analysis through "blocking", using some feature in the data to allocate records into smaller partitions (or blocks), and running

linkage algorithms independently on each block (Christen 2019). Of note, blocking on an unreliable field can lead to missed matches, making this form of blocking often undesirable (Steorts et al. 2014). Others use "filtering", which uses some deterministic or probabilistic criteria to set the match probability for certain comparison vectors to zero after they have been created (Murray 2016, McVeigh et al. 2019). In the creation of the `fastLink` method, Enamorado et al. (2019) formalized the use of a lower dimensional representation of the set of comparison vectors that allowed for much faster parameter estimation of the standard FS model; Kundinger, Reiter & Steorts (2024) and Kundinger, Aleshin-Guendel & Steorts (2024) then adopted this approach for a faster implementation of the bipartite model of Sadinle (2017).

In this work, we continue this work of proposing new record linkage methods tailored for important practical scenarios, and the computational methods to implement them in an efficient and scalable manner. We introduce Dirichlet Record Linkage (`DRL`, pronounced "drill"), an extension of the comparison vector record linkage framework for the scenario of linking one duplicate-free reference file to another file with unknown amounts of internal duplications. To do so, we model the parameter governing the amount of matches between files as Dirichlet process. Since the resulting set of possible matches for each record is extremely high dimensional, we propose an innovative Gibbs sampler utilizing the stick-breaking represenation of the Dirichlet process for tractable posterior inference. We adopt the dimension reduction techniques of Kundinger, Reiter & Steorts (2024) for additional gains in speed and scalability.

In what follows, Section 2 reviews the work of Fellegi & Sunter (1969) and a selection of Bayesian extensions to contextualize our intervention. Section 3 introduces `DRL`, and describes our approach for efficient posterior sampling. Sections 4 and 5 demonstrate the speed and accuracy of our method in relation to several other comparison vector techniques in a series of simulations a case study of voter registration records in North Carolina. Finally, Section 6 summarizes our work and discusses areas for further research.

2 THE FELLEGI-SUNTER FRAMEWORK FOR RECORD LINKAGE

Consider two data files A and B comprising n_A and n_B records, respectively, and including F linkage variables measured in both files. For $i = 1, \ldots, n_A$, let record i be given by $A_i = (A_{i1}, \ldots, A_{iF})$, so that $A = (A_i : i = 1, \ldots, n_A)$. Similarly, for $j = 1, \ldots, n_B$, let record j be given by $B_j = (B_{j1}, \ldots, B_{jF})$, so that $B = (B_j : j = 1, \ldots, n_B)$.

In record linkage tasks, records that refer to the same entity should be similar, and records that refer to different entities should be dissimilar. To represent this, Fellegi & Sunter (1969) proposed using the comparison vector $\gamma_{ij} = (\gamma_{ij}^1, \ldots, \gamma_{ij}^F)$, where γ_{ij}^f is a comparison for field f between records A_i and B_j. Binary comparisons are commonly used due to their simplicity; partial agreement or more complicated comparisons can be used through similarity metrics or distance functions (Winkler 1990, Bilenko et al. 2003, Elmagarmid et al. 2007). We assume that each field's comparison is discretized , and we let L_f denote the number of categories for field f. We collect all of the comparison vectors as $\gamma = \{\gamma_{ij}\}_{i=1,j=1}^{n_A,n_B}$.

Consider records from A and B as disjoint set of nodes. We have an edge between two records if they are coreferent (or matching). Our parameter of interest is the collection of edges representing coreferent records. This parameter can be expressed in various ways depending on the model assumptions for a particular approach to record linkage. For example, Fellegi & Sunter (1969) used a coreferent matrix $\Delta \in \{0, 1\}^{n_A \times n_B}$, where

$$\Delta_{ij} = \begin{cases} 1, & \text{if records } A_i \text{ and } B_j \text{ refer to the same entity;} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

This sparse matrix representation can become cumbersome for large linkage tasks. In the setting where each record in B can match to at most one record in A, Sadinle (2017) proposed using the more compact vector $\boldsymbol{Z} = (Z_1, \ldots, Z_{n_B})$ for the records in B such that

$$Z_j = \begin{cases} i, & \text{if records } A_i \text{ and } B_j \text{ refer to the same entity;} \\ n_A + j, & \text{if record } B_j \text{ does not have a match in } A. \end{cases} \tag{2}$$

2.1 Models for Comparison Vector Based Record Linkage

For modeling the collection of $n_A n_B$ random variables Γ_{ij}, Fellegi & Sunter (1969) employ two independence assumptions: first, that comparison vectors are independent given the matching status of the record pair, and second, that the matching status of each record pair is independent of the matching status of other pairs. Using these independence assumptions, one specifies a mixture model for each Γ_{ij} (e.g., as in Winkler 1999, Jaro 1989, Larsen & Rubin 2001, Enamorado et al. 2019). We have

$$\Gamma_{ij} \mid \Delta_{ij} = 1 \overset{iid}{\sim} \mathcal{M}(\boldsymbol{m}), \tag{3a}$$

$$\Gamma_{ij} \mid \Delta_{ij} = 0 \overset{iid}{\sim} \mathcal{U}(\boldsymbol{u}), \tag{3b}$$

$$\Delta_{ij} \overset{iid}{\sim} \text{Bernoulli}(\lambda). \tag{3c}$$

Here, $\mathcal{M}$ and $\mathcal{U}$ are the distributions for matching and nonmatching record pairs, $\boldsymbol{m}$ and $\boldsymbol{u}$ are their respective sets of parameters, and λ is the marginal probability that a record pair is a match. When using comparison vectors with discrete agreement levels, $\mathcal{M}$ and $\mathcal{U}$ are collections of independent multinomial distributions for each linkage feature. Accordingly, $\boldsymbol{m} = (\boldsymbol{m}_1, \ldots, \boldsymbol{m}_F)$, where $\boldsymbol{m}_f = (m_{f1}, \ldots, m_{fL_f})$ and $m_{fl} = p(\Gamma_{ij}^f = l | \Delta_{ij} = 1)$ for all fields f and agreement levels l. The $\boldsymbol{u}$ parameters are defined similarly, with $u_{fl} = p(\Gamma_{ij}^f = l | \Delta_{ij} = 0)$.

One advantage of Bayesian methods is the ability to use prior distributions tailored to specific model assumptions, which tends to produce better results than post-processing after using more general methods. Under the assumption that there are no duplicates within files, Sadinle (2017) proposed the "beta distribution for bipartite matching," given by

$$\mathbb{P}(Z \mid \pi) = \frac{[n_1 - n_{12}(Z)]!}{n_1!} \pi^{n_{12}(Z)} (1 - \pi)^{n_2 - n_{12}(Z)}, \tag{4}$$

$$\pi \sim \text{Beta}(\alpha_\pi, \beta_\pi). \tag{5}$$

This prior induces a Gibbs sampler that strictly enforces one-to-one matching, and has has been shown to outperform the base FS method when model assumptions are satisfied. However, maintaining the dependencies between records has limited this method to small to moderate linkage tasks.

More recently, Aleshin-Guendel & Sadinle (2023) extended the FS framework for the setting of arbitrary amount of duplications within and across an arbitrary number of files. To do so, they presented a generative process for multifile partitions, and parameterized each step of this generative process. This leads to a structured prior for multifile partitions consisting of

1. A prior for the number of unique entities exhibited in the linkage task. By default, this is taken to be uniform over total number of records in the linkage task.

2. Given the total number of unique entities, a prior for the *overlap table*, or the number of unique entities exhibited in each file. This prior is given by a multinomial-Dirichlet distribution.

3. Given the number of unique entities exhibited in each file, a prior for the number of records in each file associated with each unique entity. By default, this is taken to be a Poisson distribution truncated between 1 and the number of records in each file.

4. Given the number of records in each file associated with each unique entity, a prior for the within-file partitions of records to entities. This is taken to be uniform over the number of such assignments.

5. Given the overlap table and within-file partitions, a prior for the matching between files. This is taken to be uniform over all possible such matchings.

To our knowledge, this is as of yet the only comparison vector method that can identify duplicates within files with out employing the full record pair independence from the original Fellegi & Sunter (1969). This structured prior induces a Gibbs sampler on the partition of records corresponding to the unique entities in the record linkage task. Similarly the beta prior for bipartite matching however, this Gibbs sampler becomes infeasible for larger linkage tasks.

2.1.1 Fast Beta Linkage

Kundinger, Reiter & Steorts (2024) proposed to model each element of Z independently, relaxing the assumption that there are no duplicates within each database. Under this

relaxation, they introduced the fast beta linkage (`fabl`) prior:

$$\mathbb{P}(Z_j = i \mid \pi) = \begin{cases} \pi/n_1, & \text{if } i \leq n_1, \\ 1 - \pi, & \text{if } i = n_1 + j, \end{cases} \tag{6}$$

$$\pi \sim \text{Beta}(\alpha_\pi, \beta_\pi), \tag{7}$$

where α_π, β_π are fixed and known. This prior says that a record B has some match in A with probability π, and that each record in A is equally likely to be that match. The `fabl` prior closely mimics the beta prior for bipartite matchings in (4), without enforcing the bipartite matching restriction within the Gibbs sampler. This prior was originally proposed for an efficient Gibbs sampler, but was recently shown to be estimable through variational inference for substantial computational gains (Kundinger, Aleshin-Guendel & Steorts 2024). If desired, a bipartite matching can be acquired through a simple post-processing step after computing the Bayes estimate.

3 DIRICHLET RECORD LINKAGE

We showed in Kundinger, Reiter & Steorts (2024) and Kundinger, Aleshin-Guendel & Steorts (2024) that the `fabl` framework tends to outperform standard FS when each record in B has at most one match in A. However, significant problems arise when a record in B has multiple matching records in A. In particular, since matching probability is normalized among all records in A, matching probability is split among multiple records such that none of the matches achieves high enough posterior probability to be identified through the Bayes estimate. This amounts to a paradox: the more matches that record B_j has in A, the less likely the algorithm is to find a match. Here, we attempt to resolve this paradox by extending the `fabl` framework to handle these internal duplications in A. We emphasize that we are not interested in deduplication within A for its own sake, but rather aim to conduct record linkage in light the problems posed by these duplications. Therefore, duplications within files that have no duplications across files will go undetected; we believe this acceptable for many practical applications.

We provide updated notation to allow us to describe matching in this setting. Let

$Z_j = (Z_{j,1}, \ldots)$ be a set containing the indices for all of the records in A that are a match with record B_j, and let $Z = \{Z_j | j = 1, \ldots, n_B\}$ denote the collection of such sets for all records in B. Let $|Z_j|$ denote the number of records in A that are linked to B_j. We use $Z_j = \emptyset$ to denote when B_j has no match in A.

Define a vector of probabilities $\boldsymbol{\pi} = (\pi_0, \ldots)$ where π_k is the probability that some record B_j has exactly k matches in A. To avoid setting a maximum number of matches for any B_j, we assign $\boldsymbol{\pi}$ a Dirichlet process prior. That is,

$$f(\boldsymbol{\pi}) = \sum_{k=0}^{\infty} \pi_k \delta_{\pi_k}(\boldsymbol{\pi}), \tag{8}$$

where δ_{π_k} is the indicator function which evaluates to zero everywhere, except for $\delta_{\pi_k}(\pi_k) = 1$. We model each π_k as a product of conditional probabilities: let η_k be the probability that some record in B has at least k matches, given that it has at least $k-1$ matches. This gives us the stick breaking representation

$$\pi_k = (1 - \eta_{k+1}) \prod_{c=1}^{k} \eta_c, \tag{9}$$

where η_k are independent random variables from a Beta$(1, \beta_\eta)$ distribution.

Conditional on B_j having k matches, we construct a prior specification on Z such that each matching Z_j of length $|Z_j|$ is equally likely. Since there are $\binom{n_A}{|q|} = \frac{n_A!}{(n_A - |q|)!|q|!}$ ways to select $|q|$ matching records out of all n_A possible records, we use

$$p(Z_j = q | \boldsymbol{\pi}) = \frac{(n_A - |q|)!|q|!}{n_A!} \pi_{|q|}. \tag{10}$$

In Appendix 7.1, we show that the full conditional for Z_j in this setting is given by

$$p\left(Z_j = q | \gamma, \boldsymbol{m}, \boldsymbol{u}, \pi\right) \propto \frac{(n_A - |q|)!|q|!}{n_A!} \pi_{|q|} \prod_{i \in q} w_{ij}, \tag{11}$$

where for all $i \in \{1, \ldots, n_A\}$ and $j \in \{1, \ldots, n_B\}$,

$$w_{ij} = \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}} \right)^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)}. \tag{12}$$

Finally, to obtain an estimate $\hat{\boldsymbol{Z}}$ of the linkage structure, we use the loss functions and Bayes estimate from Sadinle (2017) and adapted by Kundinger, Reiter & Steorts (2024).

Since (10) does not strictly enforce the requirement that there are no duplicates in B, it is possible for this Bayes estimate to link multiple records in B to the same record in A. To obtain a Bayes estimate that corresponds to our model assumptions, we minimize the expected loss subject to the constraint that $\hat{Z}_{j,k} \neq \hat{Z}_{j',k}$ for all $j \neq j'$ and all k. See Supplement 7.4 for details regarding this initial Bayes estimate and this post-processing procedure.

3.1 Sequential Sampler

Sampling the full conditional in (11) is infeasible for almost any record linkage task. In particular, there are 2^{n_A} possible matchings for each B_j, meaning that computational complexity for the sampler would be $O\left(n_B 2^{n_A}\right)$. One could reduce this computational burden by setting a maximum number of matches, K, per record in B, but this would still require $\sum_{k=1}^{K} \frac{n_A!}{(n_A-k)!k!}$ possible options for the set Z_j, and would be prohibitive for most values of n_A seen in record linkage problems. Through Gibbs sampling however, we can break this joint distribution into a sequence of more simple conditional univariate distributions. This allows for a more computationally efficient sampler, and allows us to learn K from the data, rather than setting it ahead of time.

We use the stick breaking representation in (9) to generalize the fast beta prior in (6), producing a sequence of priors that allows for multiple matchings. When B_j has been linked to $k-1$ records, we say that the probability that B_j has a k^{th} match is η_k, and that all remaining records in A are equally likely to be linked. Let $Z_{j,-k} = (Z_{j,1}, \ldots, Z_{j,k-1})$ be the set of records linked to B_j before the k^{th} matching phase. We use

$$p(Z_{j,k} = q_k | \eta_k) = \begin{cases} \frac{\eta_k}{n_A-(k-1)}, & q_k \notin A_{j,k}, \\ 1-\eta_k, & q_k = \emptyset; \end{cases} \tag{13}$$

where $A_{j,k} = [n_A] \setminus Z_{j,-k}$ is the set of records in A that are available to be matched with B_j. This sequence of priors leads to sequence of posteriors that can be used to sample

arbitrarily many links for record B_j. These posteriors are given by

$$p(Z_{j,k} = q_k | Z_{j,k-1}, \eta_k, \boldsymbol{m}, \boldsymbol{u}, \gamma) \propto \begin{cases} \frac{\eta_k}{n_A - (k-1)} w_{q_k,j}, & q_k \in A_{j,k}, \\ 1 - \eta_k, & q_k = \emptyset, \end{cases} \tag{14}$$

as derived in Appendix 7.2.

This sequential sampler produces an output $Z_j = q = (q_1, \ldots, q_k)$ when $Z_{j,c} = q_c$ for steps $c \in \{1, \ldots, k\}$, and the $k+1$ step produces $Z_{j,k+1} = \emptyset$. While this vector is neccesarily ordered, all reorderings of the same elements are equivalent for the purposes of record linkage. That is, $Z_j = (i, i')$ and $Z_j = (i', i)$ both communicate that record B_j is matched to records A_i and $A_{i'}$. Let $\sigma(q)$ denote all possible orderings of the elements of q, and note that there are $|q|!$ such orderings. Marginalizing over such orderings, we have

$$p(Z_{j,k+1} = \emptyset | \Gamma_{\cdot j}, \boldsymbol{m}, \boldsymbol{u}, \boldsymbol{\eta}) \sum_{q' \in \sigma(q)} \prod_{c=1}^{k} p(Z_{j,c} = q'_c | \Gamma_{\cdot j}, \boldsymbol{m}, \boldsymbol{u}, \boldsymbol{\eta}) \tag{15}$$

$$\propto (1 - \eta_{k+1}) \sum_{q' \in \sigma(q)} \prod_{c=1}^{k} \frac{\eta_c}{n_A - (c-1)} \prod_{c=1}^{k} w_{q'_c,j} \tag{16}$$

$$= (1 - \eta_{k+1}) k! \prod_{c=1}^{k} \frac{\eta_c}{n_A - (c-1)} \prod_{c=1}^{k} w_{q_c,j} \tag{17}$$

$$= \frac{(n_A - k)! k!}{n_A!} (1 - \eta_{k+1}) \prod_{c=1}^{k} \eta_c \prod_{c=1}^{k} w_{q_c,j} \tag{18}$$

$$= \frac{(n_A - k)! k!}{n_A!} \pi_k \prod_{c=1}^{k} w_{q_c,j} \tag{19}$$

$$= p\left(Z_j = q | \gamma, \boldsymbol{m}, \boldsymbol{u}, \pi\right). \tag{20}$$

Thus, the probability that the sequential process samples each of the components of q (in any order) is equal to the joint probability of q as expressed in the joint distribution in (11).

This sequential sampler amounts to an extension of `fabl` with an iterative matching phase. In each iteration of the Gibbs sampler, we sample an initial set of links using η_1. For each record in B that was found to have a link, we remove the linked record in A from consideration, and then sample another potential link with η_2. We continue, using η_k in the k^{th} matching step, until no new links are found, at which we point the matching phase

terminates. The $\boldsymbol{\eta}, \boldsymbol{m}$, and $\boldsymbol{u}$ parameters are estimated based on all of the links identified, regardless of the order in which they are sampled. Crucially, there is no need to specify a maximum number of links per record, as this estimated through the model.

3.2 Efficient Sequential Sampling

Following Kundinger, Reiter & Steorts (2024) we use hashing to reduce the computational complexity of the Gibbs sampler. Since each component γ_{ij}^f of the comparison vector is discrete, there are only finitely many possible realizations of the comparison vector γ_{ij}. Let P be the number of unique agreement patterns observed in Γ. This number is bounded above by $P^* = \prod_{f=1}^{F}(L_f + 1)$, where the addition of 1 to L_f for each field accounts for the possibility of missing values. This upper bound does not scale with n_1 or n_2, but rather is determined by F and L_f.

Let $N_{p_j} = \sum_{i=1}^{n_B} I(\gamma_{ij} = h_p)$ denote the number of records in A with which record j in B has agreement pattern p. Collect these counts in $\mathcal{N} = \{N_{p_j} \mid j \in [n_B], p \in [P]\}$. Let $N_p = \sum_{j=1}^{n_B} N_{p_j}$ denote the total number of record pairs with agreement pattern p. Finally, let $r_{p_j} = \{i \in [n_A] \mid \gamma_{ij} = h_p\}$ be the set of records in A with which record j in B has agreement pattern p, and collect these sets as $\mathcal{R} = \{r_{p_j} \mid p \in [P], j \in [n_2]\}$.

We adapt the split sampler from Kundinger, Reiter & Steorts (2024) for this multiple match setting. Let $r_{p_j}(k) = r_{p_j} \; Z_{j,-k}$ be the set of records of agreement pattern p available for matching at the k^{th} matching step, and let $N_{p_j}(k) = |r_{p_j}(k)|$ be the number of such records. We first sample among $P + 1$ options for the agreement pattern between B_j and its potential link. Define r as an arbitrary set of records. We have

$$p\left(Z_{j,k} \in r \mid \tilde{\gamma}, \boldsymbol{m}, \boldsymbol{u}, \eta_k\right) \propto \begin{cases} \frac{\eta_k N_{p_j}(k)}{n_A - (k-1)} w_p, & r = r_{p_j}(k); \\ 1 - \eta_k, & r = \emptyset. \end{cases} \tag{21}$$

Since all remaining records in A sharing the same agreement pattern with B_j are equally likely, we then sample among candidate records uniformly using

$$p\left(Z_{j,k} = q \mid Z_{j,k} \in r, \boldsymbol{m}, \boldsymbol{u}, \eta_k\right) = \begin{cases} \frac{1}{N_{p_j}(k)}, & r = r_{p_j}(k) \text{ and } q \in r; \\ 1, & r = \emptyset \text{ and } q = \emptyset. \end{cases} \tag{22}$$

See Appendix 7.2 for a more detailed explanation of hashing.

We take a moment to emphasize the computational advantages to this approach. Sampling from the full conditional shown in (11) would have complexity $O(2^{n_A} n_B)$, and would be nearly impossible for any reasonable record linkage task. Using the sequential sampler shown in (14) has complexity $O\left(n_A(n_B + \sum_{k=1}^{K} n_k)\right)$, which would still still grow quadratically as the size of the linkage task grows. However, sampling from (21) and (22) has complexity $O\left(P(n_B + \sum_{k=1}^{K} n_k)\right)$, which grows linearly in the size of the base dataset. Thus, we have produced a computationally efficient sampler with speed comparable to the $O(Pn_B)$ complexity of base `fabl`.

4 SIMULATIONS

We demonstrate the accuracy of the `DRL` approach at various levels of overlap between files and errors between matching records through an adaptation of the simulation study from Sadinle (2017) and Kundinger, Reiter & Steorts (2024).

For each simulation, we construct two data files A and B such that there are in which there are 25, 125, or 225 records in B that have matching records in A. The matching record in A exhibits 1, 2, or 3 errors across the five fields used for linkage. Then, every record in A that has a matching record in B is duplicated, such that each simulation has 50, 250, or 450 matching record pairs in total. We use uniform priors for the $\boldsymbol{m}$ and $\boldsymbol{u}$, with $\alpha_{fl} = \beta_{fl} = 1$ for all f and l. We use uniform priors for π for standard `fabl`, and also uniform priors for each η_k in sequential sampler for `DRL`. We run the Gibbs sampler for 1000 iterations, and discard the first 100 as burn-in.

We run `multilink` under two different parameter settings. We first use the default setting (referred to as "mutlink_1" in the figures below), where the prior distribution for the number of records in A that can match a single record in B is a Poisson distribution with mean 1. Second, we use a prior closer to the data generating process for this simulation (referred to as "mutlink_2" in the figures below), and use a Poisson distribution with mean 2, and the maximum number of matching records also set to 2.

In these simulations, standard `fabl` drastically underperforms, so much so we omit

Figure 1: Recall for simulations in Section 4.

results from the figures below. For each record in B with matching records in A, the posterior match probability is split between the two matching records. Due to the randomness of the Gibbs sampler, one of the records occasionally has a posterior probability over 0.5 and is thus identified through the Bayes estimate, but often, both records have posterior probability below 0.5. Under `vabl`, the posterior probability of the two records is cut precisely in half, and thus `vabl` did not identify any matching record pairs in any of the simulations. Again, these results are omitted.

We see that `fastLink` generally has poorer performance than the more complex models at mid to high overlap. In particular, without a post-processing step tailors the scenario where one record in B can match to multiple records in A, `fastLink` tends to produce a large number of false positives, hindering precision, as shown in Figure 2.

The performance of `multilink` is more varied. When using substantive prior information, `multilink` outperforms `DRL` in terms of overall F-measure in the medium and high overlap settings. However, when the number of matching records pairs is low, `multilink` underperforms. This may be because the model being fit is more complex, and so it is more difficult to learn the parameters from such few observations. Notably however, the performance of `multilink` is highly sensitive to prior specification. When using default

Figure 2: Precision for simulations in Section 4.

Figure 3: F-Measure for simulations in Section 4.

settings, `multilink` showed considerably lower recall and precision at all levels of error and overlap.

We see that `DRL` is a strong alternative to `multilink` and `fastLink` in this setting. `DRL` maintains strong performance even in the low overlap scenario where `multilink` fails, and outperforms `fastLink` in terms of F-measure at all levels of error and overlap. We accomplish this through default, uniform priors for the sequence of η_k parameters, without needing the informative priors required for `multilink`. Additionally, the average computation time for 1000 iterations of the Gibbs sampler for `DRL` was around 30 seconds, while it was around 1000 seconds for `multilink`.

5 CASE STUDY

We demonstrate our method on the North Carolina Voter Registration (`NCVR`) database taken two months apart (Christen 2014). The snapshots are filtered to include only those voters whose details changed over the two-month period, so there are matching records with full agreement on all fields. We use first name, middle name, last name, as string fields, and street address and age as categorical fields. Unique voter registration numbers are provided, however they are known to contain some errors. The `NCVR` dataset is not publicly available due to sensitive information. However, we have permission to utilize it for publication by its owner.

Using the voter registration numbers, we can see that each file has internal duplication rates of about 1%. In this analysis, we deduplicate file B, and left A with duplicates. In practice, such low amount of internal duplication may not warrant the use of Dirichlet Beta Linkage since `vabl` is considerably faster. However, we demonstrate here that even with such few internal duplications, `DRL` is effective at identifying this multiple matches and does not declare too many false matches.

I compare four approaches: standard `fabl`, our proposed `DRL` method, `fastLink`, and `fastLink` with the Jaro correction. This task is too large for running `multilink` in its current implementation in `R`, and is thus omitted. For `fastLink`, we use the threshold of 0.5 to declare matches, and for `fabl` and `DRL`, we use losses that are equivalent to a threshold

method	recall	precision	f_measure
`DRL`	0.9891623	0.9735309	0.9812843
`fabl`	0.9743739	0.9798799	0.9771191
`fastLink`	0.9988472	0.8494865	0.9181321
`fastLink` (with Jaro)	0.9514490	0.9664229	0.9588775

Table 1: Accuracy results for `NCVR`, showing the `DRL` has the highest f-measure.

of 0.5. Results are in Table 1.

We see that `fastLink` without the one-to-one post-processing results in the highest recall, but leads to an undesirable amount of false positives. Since the posterior match probability of each record pair is computed independently, we expect such behavior on large linkage tasks such as this. When we use the Jaro post-processing to achieve a one-to-one matching, we get considerably better results. However, we lose the possibility of identifying cases where one record in B matches to two records in A, and we still have worse recall and precision than attained under standard `fabl`.

In Figure 4, we show the show the precision, recall, and F-measure under various choices of match probability thresholds. We see that `fabl` and `DRL` produce more refined match probability estimates, allowing the accuracy metrics to vary smoothly across the different thresholds. In contrast, all record pairs of the same agreement pattern under `fastLink` have the same posterior probability, leading to much a much coarser posterior distribution, leading to more erratic curves. We see that `fabl` maintains the highest precision at all thresholds, but that `DRL` is able to identify enough more additional true matches that `DRL` maintains the highest overall F-measure.

Lastly, we note a limitation in using preexisting software for record linkage tasks. In its currently implementation, `fastLink` allows users to define normalized Levenshtein distance thresholds for coding field comparisons into three discrete levels. In this data however, it is common for a record to contain a middle initial (like "E"), rather than a full middle name (like "Elizabeth"). Using Levenshtein distance thresholds, the comparison field for middle name for such a pair of records could coded as a full disagreement, when it should be more

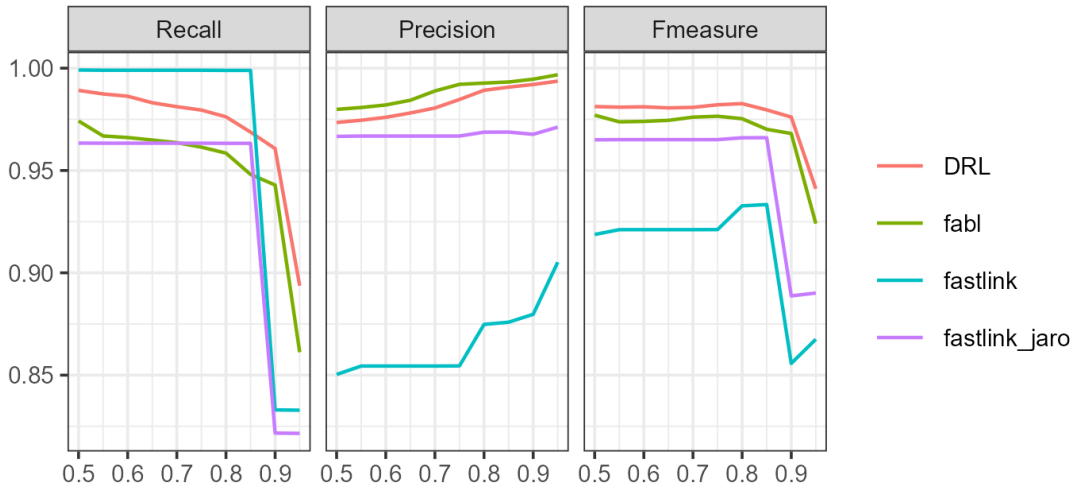

Figure 4: Accuracy metrics for NCVR data at various match probability thresholds. We see that `DRL` maintains the strongest F-measure for all thresholds considered.

reasonably given its own agreement level. Likewise, when two records have a middle initial that happens to match, this is coded as a full agreement, even though it is possible these initials represent different names. In this analysis, 27.6% of the false matches under `DRL` at the 0.5 probability threshold included at least one record containing a middle initial rather than middle name. A more careful construction of the comparison vectors is likely to improve results for all methods considered.

6 CONCLUSION

In Kundinger, Aleshin-Guendel & Steorts (2024), we showed how the `fabl` model could be fit through variational inference, resulting in variational beta linkage, or `vabl`. As variational inference is much faster than MCMC methods, we showed that `vabl` was 50-250 times faster than `fabl` in a number of simulations and case studies. Unfortunately, it is infeasible to fit the `DRL` model directly through variational inference, since the joint distribution of all possible combinations of records in A is far too high dimensional to estimate without the use of the sequential sampler. However, if we set a limit for the number of matches for each record in B, and use stringent blocking criteria to reduce the set of possible combinations of matches, variational inference may be of use in this scenario as well. (I think this is

probably too specific. Just typing it out as brainstorm for what to put here.)

REFERENCES

Aleshin-Guendel, S. & Sadinle, M. (2023), 'Multifile Partitioning for Record Linkage and Duplicate Detection', *Journal of the American Statistical Association* **118**(543), 1786–1795.

Betancourt, B., Sosa, J. & Rodríguez, A. (2022), 'A prior for record linkage based on allelic partitions', *Computational Statistics & Data Analysis* **172**, 107 – 474.

Bilenko, M., Mooney, R. J., Cohen, W. W., Ravikumar, P. & Fienberg, S. E. (2003), 'Adaptive Name Matching in Information Integration', *IEEE Intelligent Systems* **18**(5), 16–23.

Christen, P. (2014), Preparation of a real temporal voter data set for record linkage and duplicate detection research, Technical report, Australian National University.
URL: *http://users.cecs.anu.edu.au/ christen/publications/ncvoter-report-29june2014.pdf*

Christen, P. (2019), 'Data Linkage: The Big Picture', *Harvard Data Science Review* **1**(2). https://hdsr.mitpress.mit.edu/pub/8fm8lo1e.

Elmagarmid, A. K., Ipeirotis, P. G. & Verykios, V. S. (2007), 'Duplicate Record Detection: A Survey', *IEEE Transactions on Knowledge and Data Engineering* **19**(1), 1–16.

Enamorado, T., Fifield, B. & Imai, K. (2019), 'Using a probabilistic model to assist merging of large-scale administrative records', *American Political Science Review* **113**, 353–371.

Fellegi, I. P. & Sunter, A. B. (1969), 'A Theory for Record Linkage', *Journal of the American Statistical Association* **64**(328), 1183–1210.

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.

Kundinger, B., Aleshin-Guendel, S. & Steorts, R. C. (2024), 'Variational Beta Linkage', *TBD* .

Kundinger, B., Reiter, J. & Steorts, R. C. (2024), 'Efficient and Scalable Bipartite Matching with Fast Beta Linkage (fabl)', *Bayesian Analysis* .

Larsen, M. D. & Rubin, D. B. (2001), 'Iterative Automated Record Linkage Using Mixture Models', *Journal of the American Statistical Association* **96**(453), 32–41.

Marchant, N. G., Kaplan, A., Elazar, D. N., Rubinstein, B. I. P. & Steorts, R. C. (2021), 'd-blink: Distributed end-to-end bayesian entity resolution', *Journal of Computational and Graphical Statistics* **30**(2), 406–421.
URL: *https://doi.org/10.1080/10618600.2020.1825451*

McVeigh, B. S., Spahn, B. T. & Murray, J. S. (2019), 'Scaling bayesian probabilistic record linkage with post-hoc blocking: An application to the california great registers', *arXiv preprint arXiv:1905.05337* .

Murray, J. S. (2016), 'Probabilistic record linkage and deduplication after indexing, blocking, and filtering', *Journal of Privacy and Confidentiality* **7**(1), 3–24.

Sadinle, M. (2017), 'Bayesian Estimation of Bipartite Matchings for Record Linkage', *Journal of the American Statistical Association* **112**(518), 600–612.

Steorts, R. C., Hall, R. & Fienberg, S. E. (2016), 'A Bayesian Approach to Graphical Record Linkage and Deduplication', *Journal of the American Statistical Association* **111**(516), 1660–1672.

Steorts, R. C., Ventura, S. L., Sadinle, M. & Fienberg, S. E. (2014), A comparison of blocking methods for record linkage, *in* J. Domingo-Ferrer, ed., 'Privacy in Statistical Databases', Springer International Publishing, Cham, pp. 253–268.

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.

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, pp. 354–359.

Winkler, W. E. (1999), The State of Record Linkage and Current Research Problems, Technical report, Statistical Research Division, U.S. Bureau of the Census.

Wortman, J. P. H. (2019), Record linkage methods with applications to causal inference and election voting data, PhD thesis, Duke University.

7 APPENDIX

7.1 Full Conditional for Z_j

Following the observation of Wortman (2019) and elaborated by Kundinger, Reiter & Steorts (2024), when B_j does not link to any record in A (such that $|Z_j| = 0$) the contribution to the likelihood is simply a product of u parameters, which we will call c_j:

$$p(\Gamma_{\cdot j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_j = \emptyset) = \prod_{i=1}^{n_A} \prod_{f=1}^{F} \prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)} = c_j. \tag{23}$$

When $Z_j = q = (q_1, \ldots, q_k)$ for some $|q| > 0$, we have

$$p(\Gamma_{\cdot j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_j = q) = \prod_{i \in q} \prod_{f=1}^{F} \prod_{l=1}^{L_f} m_{fl}^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)} \prod_{i \notin q} \prod_{f=1}^{F} \prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)}. \tag{24}$$

We multiply and divide by the u parameters for the matching record pairs to obtain

$$p(\Gamma_{\cdot j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_j = q) = \prod_{i \in q} \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}} \right)^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)} \prod_{i=1}^{n_A} \prod_{f=1}^{F} \prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)} \tag{25}$$

$$= c_j \prod_{i \in q} w_{ij}. \tag{26}$$

Lastly, we multiply the likelihood by the prior in (**??**) to obtain the posterior distribution. For $Z_j = q$ where $|q| = k$, we have

$$p\left(Z_j = q|\gamma, \boldsymbol{m}, \boldsymbol{u}, \pi\right) = \frac{\frac{(n_A - k)!|k|!}{n_A!} \pi_k c_j \prod_{i \in q} w_{ij}}{\sum_{h \in \mathcal{Z}} \frac{(n_A - |h|)!|h|!}{n_A!} \pi_{|h|} c_j \prod_{i \in h} w_{ij}} \tag{27a}$$

$$= \frac{\frac{(n_A - k)!|k|!}{n_A!} \pi_k \prod_{i \in q} w_{ij}}{\sum_{h \in \mathcal{Z}} \frac{(n_A - |h|)!|h|!}{n_A!} \pi_{|h|} \prod_{i \in h} w_{ij}} \tag{27b}$$

$$\propto \frac{(n_A - k)!|k|!}{n_A!} \pi_k \prod_{i \in q} w_{ij}. \tag{27c}$$

7.2 Derivation of Sequential Sampler

We now provide the derivation of the sequential sampler, following the argument presented in Section **??**. Suppose B_j has been linked to k records in A. Let $Z_{j,-k} = (Z_{j,1}, \ldots, Z_{j,k-1})$ denote the vector of records already linked to B_j. When B_j has no additional link in A, the

contribution to the likelihood is a product of the u parameters for all remaining records. That is,

$$p(\Gamma_{\cdot j}|\boldsymbol{m},\boldsymbol{u},\pi,Z_{j,k}=\emptyset,Z_{j,-k}=q_{-k})=\prod_{i\notin Z_{j,-k}}\prod_{f=1}^{F}\prod_{l=1}^{L_f}u_{fl}^{I(\gamma_{ij}^f=l)I_{obs}(\gamma_{ij}^f)}=c_{Z_{j,-k}}. \tag{28}$$

When $Z_{j,k}=q_k$ for some $q_k>0$, we have

$$p(\Gamma_{\cdot j}|\boldsymbol{m},\boldsymbol{u},\pi,Z_{j,k}=q_k,Z_{j,-k}=q_{-k})=\prod_{f=1}^{F}\prod_{l=1}^{L_f}m_{fl}^{I(\gamma_{q_k,j}^f=l)I_{obs}(\gamma_{q_k,j}^f)}\prod_{i\notin(q_{-k},q_k)}\prod_{f=1}^{F}\prod_{l=1}^{L_f}u_{fl}^{I(\gamma_{ij}^f=l)I_{obs}(\gamma_{ij}^f)} \tag{29}$$

$$=\prod_{f=1}^{F}\prod_{l=1}^{L_f}\left(\frac{m_{fl}}{u_{fl}}\right)^{I(\gamma_{q_k,j}^f=l)I_{obs}(\gamma_{q_k,j}^f)}\prod_{i\notin(Z_{j,-k}}\prod_{f=1}^{F}\prod_{l=1}^{L_f}u_{fl}^{I(\gamma_{ij}^f=l)I_{obs}(\gamma_{ij}^f)} \tag{30}$$

$$=c_{Z_{j,-k}}w_{q_k,j} \tag{31}$$

To obtain the posterior, we multiply by the prior in (13). The posterior distribution this is given by

$$p(Z_{j,k}=q_k|Z_{j,k-1},\eta_k,\boldsymbol{m},\boldsymbol{u},\gamma)=\frac{\left(\frac{\eta_k}{n_A-(k-1)}c_{Z_{j,-k}}w_{q_k,j}\right)^{I(q_k\in N_{j,k})}+\left(c_{Z_{j,-k}}(1-\eta_k)\right)^{I(q_k=\emptyset)}}{\frac{\eta_k}{n_A-(k-1)}c_{Z_{j,-k}}\sum_{i\notin Z_{j,-k}}w_{ij}+c_{Z_{j,-k}}(1-\eta_k)} \tag{32}$$

$$=\frac{\left(\frac{\eta_k}{n_A-(k-1)}w_{q_k,j}\right)^{I(q_k\in N_{j,k})}+(1-\eta_k)^{I(q_k=\emptyset)}}{\frac{\eta_k}{n_A-(k-1)}\sum_{i\notin Z_{j,-k}}w_{ij}+(1-\eta_k)} \tag{33}$$

$$\propto\begin{cases}\frac{\eta_k}{n_A-(k-1)}w_{q_k,j}, & q_k\in N_{j,k},\\ 1-\eta_k, & q_k=\emptyset.\end{cases} \tag{34}$$

Importantly, the constant $c_{Z_{j,-k}}$ is not found in the final expression because the probability mass associated with every potential value for Z_j shares the same $c_{Z_{j,-k}}$. This does not occur due to proportionality.

7.3 Efficient Sequential Sampling Details

7.4 Bayes Estimate