1 DIRICHLET RECORD LINKAGE

Recall that for `fabl`, we used the following prior for the linkage structure Z and the rate of matching parameter π:

$$p(Z_j = q|\pi) = \begin{cases} \frac{1}{n_A}\pi, & q > 0; \\ 1 - \pi, & q = 0; \end{cases} \tag{1a}$$

$$\pi \sim \text{Beta}(\alpha_\pi, \beta_\pi). \tag{1b}$$

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. We now generalize these prior distributions to allow the records in B to match with multiple records in A.

1.1 Prior Specification

We provide updated notation to allow us to describe one record in B having multiple matches in A. Let Z_j 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 $|Z_j| = \sum_{k=1}^{\infty} I(Z_{j,k} > 0)$ 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.

We modify the `fabl` model to allow each record in B to match to multiple records in A. Define a vector of probabilities $\boldsymbol{\pi} = (\pi_0, \ldots, \pi_{n_A})$ where π_k is the probability that some record in B has exactly k matches in A. 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 replace (1a) with

$$p(Z_j = q|\boldsymbol{\pi}) = \frac{(n_A - |q|)!|q|!}{n_A!}\pi_{|q|}. \tag{2}$$

Since π is multivariate, we replace (1b) with

$$\boldsymbol{\pi} \sim \text{Dirichlet}(\alpha_\pi). \tag{3}$$

1.2 Full Conditional for Z_j

Following the observation of Wortman (2019) and elaborated by Kundinger et al. (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:

$$p(\Gamma_{.j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_j = \emptyset) = \prod_{i=1}^{n_A}\prod_{f=1}^{F}\prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l)I_{obs}(\gamma_{ij}^f)} = c_j. \tag{4}$$

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

$$p(\Gamma_{.j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_j = q) = \prod_{i \in q}\prod_{f=1}^{F}\prod_{l=1}^{L_f} m_{fl}^{I(\gamma_{ij}^f = l)I_{obs}(\gamma_{ij}^f)} \prod_{i \notin q}\prod_{f=1}^{F}\prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l)I_{obs}(\gamma_{ij}^f)}. \tag{5}$$

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

$$\begin{aligned} p(\Gamma_{.j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_j = q) &= \prod_{i \in q}\prod_{f=1}^{F}\prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}}\right)^{I(\gamma_{ij}^f = l)I_{obs}(\gamma_{ij}^f)} \prod_{i=1}^{n_A}\prod_{f=1}^{F}\prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l)I_{obs}(\gamma_{ij}^f)} && (6) \\ &= c_j \prod_{i \in q} w_{ij}. && (7) \end{aligned}$$

Lastly, we multiply the likelihood by the prior in (2) to obtain the posterior distribution. For $Z_j = q$ where $|q| = k$, we have

$$\begin{aligned} p\left(Z_j = q|\gamma, \boldsymbol{m}, \boldsymbol{u}, \pi\right) &= \frac{\frac{(n_A - k)!|k|!}{n_A!}\pi_k c_j \prod_{i \in q} w_{ij}}{\sum_{h \in \mathcal{Z}} \frac{(n_A - |h|)!|h|!}{n_A!}\pi_{|h|} c_j \prod_{i \in h} w_{ij}} && \text{(8a)} \\ &= \frac{\frac{(n_A - k)!|k|!}{n_A!}\pi_k \prod_{i \in q} w_{ij}}{\sum_{h \in \mathcal{Z}} \frac{(n_A - |h|)!|h|!}{n_A!}\pi_{|h|} \prod_{i \in h} w_{ij}} && \text{(8b)} \\ &\propto \frac{(n_A - k)!|k|!}{n_A!}\pi_k \prod_{i \in q} w_{ij}. && \text{(8c)} \end{aligned}$$

1.3 Sequential Sampler

Sampling the full conditional in (8c) 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 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 $\text{Beta}(\alpha_\eta, \beta_\eta)$ distribution.

We generalize the fast beta prior in (1a) to 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

$$p(Z_{j,k} = q_k | \eta_k) = \begin{cases} \frac{\eta_k}{n_A - (k-1)}, & q_k \notin A_{j,k}, \\ 1 - \eta_k, & q_k = \emptyset; \end{cases} \tag{10}$$

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

$$p(Z_{j,k} = q_k | Z_{j,k-1}, \eta_k, \boldsymbol{m}, \boldsymbol{u}, \gamma) \propto \begin{cases} \frac{\eta_k}{n_A - (k-1)} w_{q_k, j}, & q_k \in A_{j,k}, \\ 1 - \eta_k, & q_k = \emptyset, \end{cases} \tag{11}$$

as derived in Appendix 5.1.

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

$$\begin{aligned}
p(Z_{j,k+1} = \emptyset|\Gamma_{.j}, \boldsymbol{m}, \boldsymbol{u}, \boldsymbol{\eta}) &\sum_{q' \in \sigma(q)} \prod_{c=1}^{k} p(Z_{j,c} = q'_c|\Gamma_{.j}, \boldsymbol{m}, \boldsymbol{u}, \boldsymbol{\eta}) && (12)\\
&\propto (1-\eta_{k+1}) \sum_{q' \in \sigma(q)} \prod_{c=1}^{k} \frac{\eta_c}{n_A - (c-1)} \prod_{c=1}^{k} w_{q'_c,j} && (13)\\
&= (1-\eta_{k+1}) k! \prod_{c=1}^{k} \frac{\eta_c}{n_A - (c-1)} \prod_{c=1}^{k} w_{q_c,j} && (14)\\
&= \frac{(n_A-k)!k!}{n_A!} (1-\eta_{k+1}) \prod_{c=1}^{k} \eta_c \prod_{c=1}^{k} w_{q_c,j} && (15)\\
&= \frac{(n_A-k)!k!}{n_A!} \pi_k \prod_{c=1}^{k} w_{q_c,j} && (16)\\
&= p\left(Z_j = q|\gamma, \boldsymbol{m}, \boldsymbol{u}, \pi\right). && (17)
\end{aligned}$$

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 (8c).

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.

1.4 Efficient Sequential Sampling

Following Kundinger et al. (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 et al. (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

$$p\left(Z_{j,k} \in r \mid \tilde{\gamma}, \boldsymbol{m}, \boldsymbol{u}, \eta_k\right) \propto \begin{cases} \frac{\eta_k N_{p_j}(k)}{n_A - (k-1)} w_p, & r = r_{p_j}(k); \\ 1 - \eta_k, & r = \emptyset. \end{cases} \tag{18}$$

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

$$p\left(Z_{j,k} = q \mid Z_{j,k} \in r, \boldsymbol{m}, \boldsymbol{u}, \eta_k\right) = \begin{cases} \frac{1}{N_{p_j}(k)}, & r = r_{p_j}(k) \text{ and } q \in r; \\ 1, & r = \emptyset \text{ and } q = \emptyset. \end{cases} \tag{19}$$

See Appendix 5.1 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 (8c) 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 (11) 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 (18) and (19) 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 $O(P n_B)$ complexity of base `fabl`.

2 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 et al. (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 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

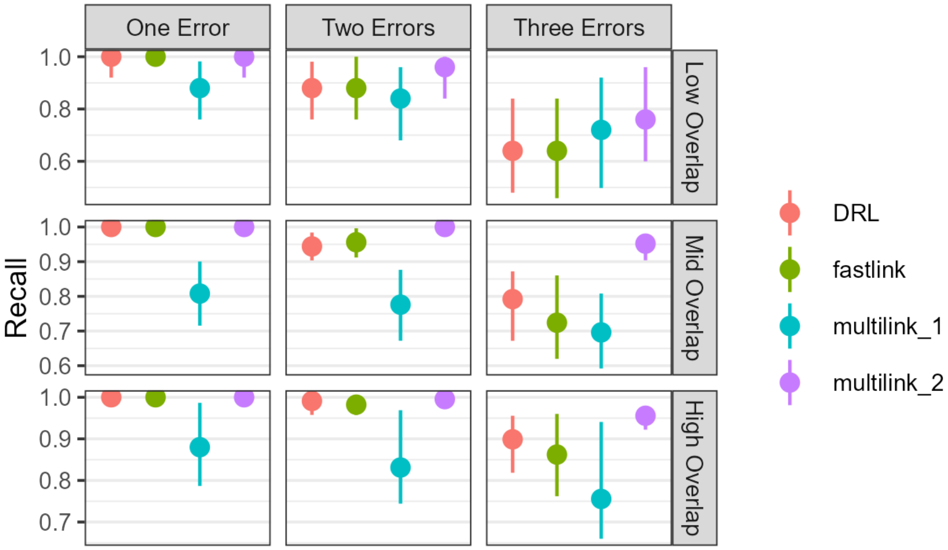

Figure 1: Recall for simulations in Section 2.

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 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`.

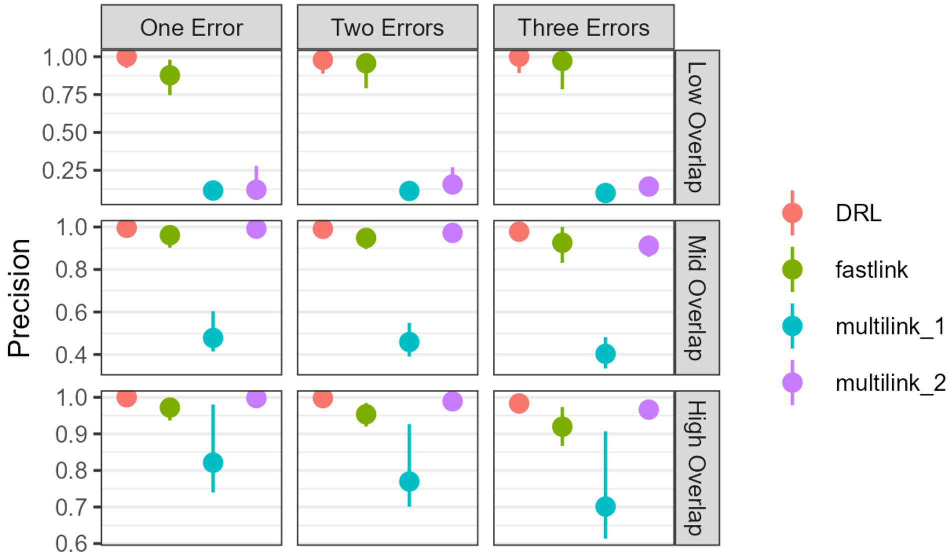

Figure 2: Precision for simulations in Section 2.

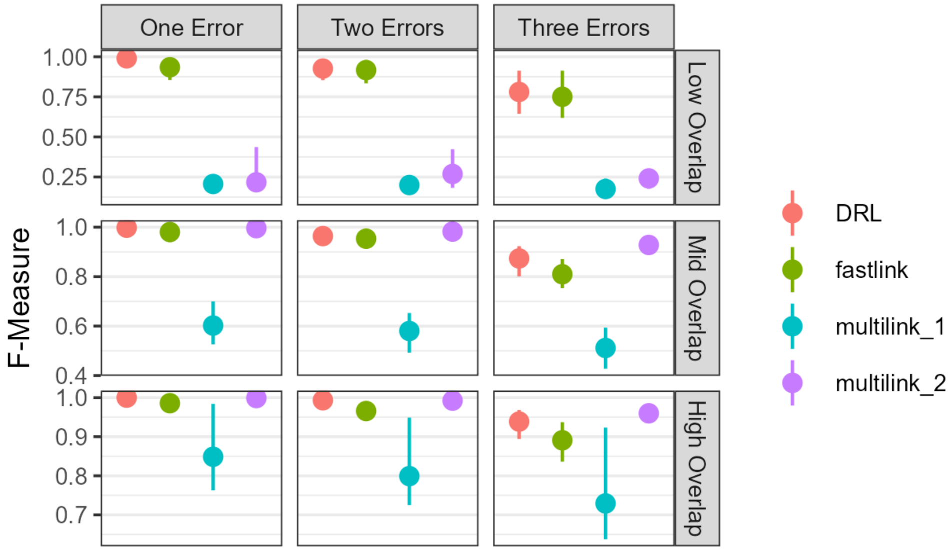

Figure 3: F-Measure for simulations in Section 2.

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.

3 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, DBL 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 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

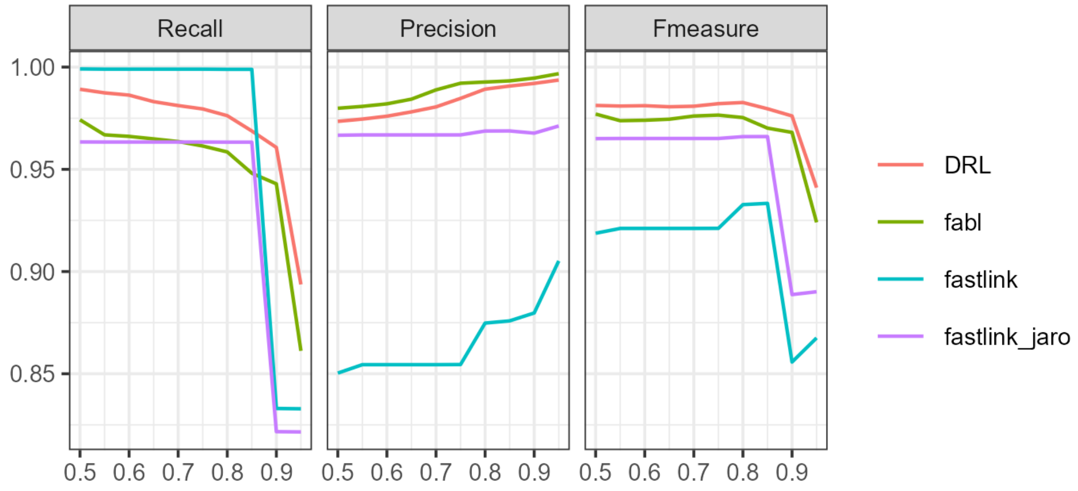

Figure 4: Accuracy metrics for NCVR data at various match probability thresholds. We see that DRL maintains the strongest F-measure for all thresholds considered.

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 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.

4 CONCLUSION

REFERENCES

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*

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

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

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

5 APPENDIX

5.1 Derivation of Sequential Sampler

We now provide the derivation of the sequential sampler, following the argument presented in Section 1.2. 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,

$$p(\Gamma_{.j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_{j,k} = \emptyset, Z_{j,-k} = q_{-k}) = \prod_{i \notin Z_{j,-k}} \prod_{f=1}^{F} \prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)} = c_{Z_{j,-k}}. \tag{20}$$

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

$$p(\Gamma_{.j}|\boldsymbol{m}, \boldsymbol{u}, \pi, Z_{j,k} = q_k, Z_{j,-k} = q_{-k}) = \prod_{f=1}^{F} \prod_{l=1}^{L_f} m_{fl}^{I(\gamma_{q_k,j}^f = l) I_{obs}(\gamma_{q_k,j}^f)} \prod_{i \notin (q_{-k}, q_k)} \prod_{f=1}^{F} \prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)} \tag{21}$$

$$= \prod_{f=1}^{F} \prod_{l=1}^{L_f} \left(\frac{m_{fl}}{u_{fl}} \right)^{I(\gamma_{q_k,j}^f = l) I_{obs}(\gamma_{q_k,j}^f)} \prod_{i \notin (Z_{j,-k}} \prod_{f=1}^{F} \prod_{l=1}^{L_f} u_{fl}^{I(\gamma_{ij}^f = l) I_{obs}(\gamma_{ij}^f)} \tag{22}$$

$$= c_{Z_{j,-k}} w_{q_k,j} \tag{23}$$

To obtain the posterior, we multiply by the prior in (10). The posterior distribution this is given by

$$p(Z_{j,k} = q_k | Z_{j,k-1}, \eta_k, \boldsymbol{m}, \boldsymbol{u}, \gamma) = \frac{\left(\frac{\eta_k}{n_A - (k-1)} c_{Z_{j,-k}} w_{q_k,j} \right)^{I(q_k \in N_{j,k})} + \left(c_{Z_{j,-k}} (1 - \eta_k) \right)^{I(q_k = \emptyset)}}{\frac{\eta_k}{n_A - (k-1)} c_{Z_{j,-k}} \sum_{i \notin Z_{j,-k}} w_{ij} + c_{Z_{j,-k}} (1 - \eta_k)} \tag{24}$$

$$= \frac{\left(\frac{\eta_k}{n_A - (k-1)} w_{q_k,j} \right)^{I(q_k \in N_{j,k})} + (1 - \eta_k)^{I(q_k = \emptyset)}}{\frac{\eta_k}{n_A - (k-1)} \sum_{i \notin Z_{j,-k}} w_{ij} + (1 - \eta_k)} \tag{25}$$

$$\propto \begin{cases} \frac{\eta_k}{n_A - (k-1)} w_{q_k,j}, & q_k \in N_{j,k}, \\ 1 - \eta_k, & q_k = \emptyset. \end{cases} \tag{26}$$

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.

5.2 Efficient Sequential Sampling Details