Journal of Applied Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/cjas20

A Bayesian approach for de-duplication in the presence of relational data

Juan Sosa & Abel Rodríguez

To cite this article: Juan Sosa & Abel Rodríguez (2024) A Bayesian approach for de-duplication in the presence of relational data, Journal of Applied Statistics, 51:2, 197-215, DOI: 10.1080/02664763.2022.2118678

To link to this article: https://doi.org/10.1080/02664763.2022.2118678

JOURNAL OF APPLIED STATISTICS
2024, VOL. 51, NO. 2, 197–215
https://doi.org/10.1080/02664763.2022.2118678

Taylor & Francis
Taylor & Francis Group

A Bayesian approach for de-duplication in the presence of relational data

Juan Sosa [a] and Abel Rodríguez [b]

[a]Departamento de Estadística, Universidad Nacional de Colombia, Bogotá, Colombia; [b]Department of Statistics, University of Washington, Seattle, WA, USA

ABSTRACT

In this paper, we study the impact of combining profile and network data in solving record de-duplication problems. We also assess the influence of a range of prior distributions on the linkage structure, and explore the use of stochastic gradient Hamiltonian Monte Carlo methods as a faster alternative to obtain samples from the posterior distribution for network parameters. Our methodology is evaluated using the RLdata500 data, which is a popular dataset in the record linkage literature.

ARTICLE HISTORY
Received 15 November 2021
Accepted 18 August 2022

KEYWORDS
Allelic partitions; De-duplicaction; entity resolution; microclustering; network data; latent space models; record linkage

1. Introduction

In database management, record de-duplication aims to identify multiple records that correspond to the same individual in the absence of a unique identifier. This process, which is a special case of the more general entity resolution problem, can be treated as a clustering problem in which each latent entity is associated with one or more noisy database records.

Various approaches for de-duplication have been considered in the literature. Domingos [13] treats the problem of de-duplication within one file through an uni-partite graph, allowing information to propagate from one candidate match to another via the attributes they have in common. Sadinle and Fienberg [33] and Sadinle [32] look for duplicate records by partitioning the data file into groups of co-referent records. They present an approach that targets this partition as the parameter of interest, thereby ensuring transitive decisions. The works by Steorts [39] and Steorts et al. [40] also permit de-duplication while handling multiple files simultaneously. Other recent contributions in entity resolution include Enamorado and Steorts [14], Tancredi et al. [42], Marchant et al. [23], and Aleshin-Guendel and Sadinle [1]. At this point, it is very important to highlight that the existing literature mainly relies on files composed of categorical or string-valued fields. To the best of our knowledge, from a model-based perspective, there are not available Bayesian approaches that simultaneously consider both attribute and relational data.

From a model-based perspective, popular choices for clustering include finite mixture models and Dirichlet/Pitman-Yor process mixture models [8,26,27]. Although these

CONTACT Juan Sosa jcsosam@unal.edu.co; jcsosam1983@gmail.com Departamento de Estadística, Universidad Nacional de Colombia, Bogotá 110931, Colombia

alternatives have proven to be successful in all sorts of applications, they are not realistic for de-duplication problems where small clusters are the rule rather than the exception. Indeed, unlike models exhibiting infinitely exchangeable clustering features, models specifically conceived for de-duplication need to generate small clusters with a minor number of records, no matter how large the database is. Specifically, we require clusters whose sizes grow sublinearly with the total number of records in order to accurately identify the latent entity underlying each observed record [3–5,25].

Findings in Sosa and Rodríguez [38] show that network data can substantially improve merging online social networks (OSNs). Hence, it makes sense that network data can be also useful in other record linkage tasks such as record de-duplication. This might be useful, for example, in identifying covert users in an OSN, which might have multiple profiles linking to the same group of individuals. Thus, our goal in this manuscript is three-fold. First, we extend the model in Sosa and Rodríguez [38] in order to address de-duplication problems. Second, we examine a range of priors on the linkage structure (cluster assignments), and then assess their influence on the posterior linkage. And finally, we also explore stochastic gradient Hamiltonian Monte Carlo methods [9] as a faster way to obtain samples from the posterior distribution. As a result, we provide a novel approach to perform ER tasks that jointly handles both attribute and relational data. Specifically, our model assumes that the file of interest has associated with it an undirected binary network in which nodes correspond to observed records in the dataset. Then, we model the network through a latent social space model [18], and use the resulting latent positions in social space to inform about potential matches. Similar to Narayanan and Shmatikov [28], the key underlying assumption is that entities that, based on their network connectivity, appear to occupy very similar positions, are more likely to be co-referent.

The remainder of this article is organized as follows: Section 2 introduces a model for de-duplication handling both attribute and relational data; there, we discuss in detail every aspect of the model including prior specification and computation. Section 3 examines the concept of microclustering and presents a number of prior distributions on the linkage structure. Section 4 compares the performance of the resulting procedures using the RLdata500 data, a popular dataset in the record linkage literature. Section 5 explores the robustness of the results to the prior specification as well as the structural features of the network information. Section 6 shows a faster way to draw samples from the posterior distribution for network parameters based on stochastic gradient methods. Lastly, we discuss our findings and future work in Section 7.

2. A de-duplication model incorporating relational data

We rely on the formulation provided in Steorts et al. [40] and later adopted in Sosa and Rodríguez [38] for one file. Specifically, we consider a single file with I records, each containing L fields in common, for which both profile (attribute) data $\mathbf{P} = [p_{i,\ell}]$ and network (relational) data $\mathbf{Y} = [y_{i,i'}]$ are available. On the one hand, we assume that attributes consists of L common categorical fields (with field ℓ having M_ℓ levels). For instance, a profile might consist of data on gender, state of residency, and race. Under this setting, $\mathbf{P}$ exhibits $L = 3$ features whose entries have $M_1 = 3$ (male, female, non-binary), $M_2 = 50$ (as of 2022, there are 50 states in the United States), and $M_3 = 6$ (White, Black or African-American, American Indian or Alaska Native, Asian, Native Hawaiian or Other Pacific

Islander, and some other race) levels, respectively. On the other hand, we also assume that relations are binary and undirected, which means that $\mathbf{Y}$ is composed of dyads indicating the absence ($y_{i,i'} = 0$) or presence ($y_{i,i'} = 1$) of a connection of interest such that $y_{i,i'} \equiv y_{i',i}$ (i.e. $\mathbf{Y}$ is symmetric). For example, the network of friendship relations as well as the network of sexual contacts can be classified as undirected networks, since there is no directionality implicit in such relations.

Our goal is straightforward: Matching (mapping) the I observed records to a set of N latent entities, where N can be as small as 1 if every record corresponds to the same entity, or as large as I if each record corresponds to a distinct entity. Such a task can be thought of as a clustering task since we aim to uncover sets of records pointing to the same latent (unobserved) entity. The mapping among observed records and latent entities can be fully characterized through the linkage linkage structure $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_I)$, where ξ_i is an integer from 1 to N indicating the latent entity the i-th record points to. Under this setting, $\boldsymbol{\xi}$ defines a partition $\mathcal{C}_{\boldsymbol{\xi}} = \{\mathrm{C}_1, \ldots, \mathrm{C}_N\}$ of size N on $\{1, \ldots, I\}$. Here, C_n is the set of those records pointing to latent entity n, i.e. $\mathrm{C}_n = \{i \in \{1, \ldots, I\} : \xi_i = n\}$. Thus, $N = \max\{\xi_i\}$, since we label the cluster assignments in $\boldsymbol{\xi}$ with consecutive integers from 1 to N. Next, we model both sources of information independently given the linkage structure $\boldsymbol{\xi}$.

2.1. Model formulation

First, we let $\pi_{n,\ell}$ be the 'true' value of field ℓ for the n-th latent entity, and also, we define binary variables $w_{i,\ell}$ such that $w_{i,\ell} = 1$ if $p_{i,\ell} \neq \pi_{\xi_i,\ell}$, and $w_{i,\ell} = 0$ if $p_{i,\ell} = \pi_{\xi_i,\ell}$, i.e. $w_{i,\ell}$ are defined as indicator variables pointing out distorted profile values $p_{i,\ell}$ with respect to latent profile values $\pi_{\xi_i,\ell}$. Thus, we model the attribute data according to the status of the field-specific distortion indicators $w_{i,\ell}$ [40], through

$$p_{i,\ell} \mid w_{i,\ell}, \xi_i, \pi_{\xi_i,\ell}, \boldsymbol{\vartheta}_\ell \overset{\text{ind}}{\sim} \begin{cases} \delta_{\pi_{\xi_i,\ell}}, & w_{i,\ell} = 0; \\ \text{Categorical}\,(\boldsymbol{\vartheta}_\ell), & w_{i,\ell} = 1, \end{cases} \tag{1}$$

where δ_a is the distribution of a point mass at a (degenerate distribution putting probability one on a), and $\boldsymbol{\vartheta}_\ell$ is an M_ℓ-dimensional vector of multinomial probabilities. Furthermore, distortion indicators $w_{i,\ell}$ are assumed to be independent a priori following a Bernoulli distribution, $w_{i,\ell} \mid \psi_\ell \overset{\text{ind}}{\sim} \text{Bernoulli}(\psi_\ell)$.

Second, we let $\boldsymbol{u}_n = (u_{n,1}, \ldots, u_{n,K})$ be the 'true' social position associated with the n-th latent entity. We assume that such a position is embedded in a Euclidean social space of dimension K. Thus, we model relational data following a standard latent distance model ([18]; see Sosa and Buitrago [37] for a review) of the form

$$y_{i,i'} \mid \beta, \xi_i, \xi_{i'}, \boldsymbol{u}_{\xi_i}, \boldsymbol{u}_{\xi_{i'}} \overset{\text{ind}}{\sim} \text{Bernoulli}\left(\text{expit}\left(\beta - \|\boldsymbol{u}_{\xi_i} - \boldsymbol{u}_{\xi_{i'}}\|\right)\right), \tag{2}$$

where $\text{expit}(x) = 1/(1 + e^{-x})$ is the inverse of the logit function, and β is the global propensity of observing a link between two records. Under this formulation, the probability of observing a link between two records is a function of their corresponding latent entities' distance in social space: If $\boldsymbol{u}_{\xi_i}$ and $\boldsymbol{u}_{\xi_{i'}}$ **move away** from each other in the social space, then $\|\boldsymbol{u}_{\xi_i} - \boldsymbol{u}_{\xi_{i'}}\|$ increases, and as a consequence, the probability of observing connection between entities n and m decreases. Moreover, we assume that latent positions $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$ are conditionally independent across entities, following a zero-mean Normal

distribution with spherical covariance matrix $\sigma^2\,\mathbf{I}_K$, where $\mathbf{I}_K$ is the $K \times K$ identity matrix, $\boldsymbol{u}_n \mid \sigma^2 \overset{\text{iid}}{\sim} \text{Normal}(\mathbf{0}, \sigma^2\,\mathbf{I}_K)$.

We complete the model by specifying independent priors:

$$\boldsymbol{\vartheta}_\ell \overset{\text{ind}}{\sim} \text{Dirichlet}(\boldsymbol{\alpha}_\ell), \quad \psi_\ell \overset{\text{ind}}{\sim} \text{Beta}(a_\ell, b_\ell),$$
$$\beta \sim \text{Normal}(0, \omega^2), \quad \sigma^2 \sim \text{Inverse-Gamma}(a_\sigma, b_\sigma).$$

Hence, our model constitutes a full hierarchical Bayesian framework for ER tasks, where $\boldsymbol{\alpha}_\ell$, a_ℓ, b_ℓ, ω^2, a_σ, b_σ, are known hyperparameters. Lastly, we devote Section 3 to discuss several prior formulations for $\boldsymbol{\xi}$.

As a final remark, we note that latent positions $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$ play a similar role in the model to that played by the true profiles $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N$, because these sets of parameters serve as 'bridges' to establish a linkage structure. In addition, we also highlight that the relational likelihood given in (2) is invariant to rotations and translations of the latent space $\mathbb{R}^K$. Since latent positions $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$ are typically nuisance parameters in the context of record linkage applications, this invariance property does not constitute a major issue.

2.2. Hyperparameter elicitation

Following Krivitsky and Handcock [21], we let network hyperparameters take the values $\omega = 100$, $a_\sigma = 2 + 0.5^{-2}$, and $b_\sigma = (a_\sigma - 1)\,\frac{\sqrt{I}}{\sqrt{I}-2}\,\frac{\pi^{K/2}}{\Gamma(K/2+1)}\,I^{2/K}$. On the other hand, following Steorts [39], we let profile hyperparameters take the values $\boldsymbol{\alpha}_\ell = \mathbf{1}_{M_\ell}$, $a_\ell = a = 1$, and $b_\ell = b = 99$.

2.3. Computation

Computation for this model can still be achieved via Markov chain Monte Carlo (MCMC) algorithms [16]. Hence, the full set of parameters in this case is

$$\boldsymbol{\Upsilon} = \left(\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N, \boldsymbol{\vartheta}_1, \ldots, \boldsymbol{\vartheta}_L, w_{1,1}, \ldots, w_{I,L}, \psi_1, \ldots, \psi_L, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N, \beta, \sigma^2, \boldsymbol{\xi}, \boldsymbol{\phi}\right),$$

where $\boldsymbol{\phi}$ includes those parameters in the prior distribution of $\boldsymbol{\xi}$. Full conditional distributions are available in closed form for all the profile parameters. As far as the network parameters is concerned, random walk Metropolis-Hastings steps can be used. Appendix 1 provides details to sample all the model parameters. Recall that the main inference goal is to make inferences about $\boldsymbol{\xi}$ by drawing samples $\boldsymbol{\xi}^{(1)}, \ldots, \boldsymbol{\xi}^{(S)}$ from the posterior distribution $p(\boldsymbol{\Upsilon} \mid \mathbf{P}, \mathbf{Y})$, and then, getting a point estimate $\hat{\boldsymbol{\xi}}$ of the overall linkage structure.

Summaries of the posterior distribution of any parameter of interest can be easily approximated using the posterior samples generated by the MCMC algorithm. For example, the posterior mean of the population size $N = \max\{\xi_i\}$ can be calculated as

$$\mathsf{E}\left[N \mid \mathbf{P}, \mathbf{Y}\right] = \frac{1}{S}\sum_{s=1}^{S} \max\left\{\xi_i^{(s)}\right\}.$$

Moreover, it is very easy to provide probabilistic statements about any pair of records. Recall that two records i_1 and i_2 match if and only if they point to the same latent individual,

i.e. $\xi_{i_1} = \xi_{i_2}$. Hence, the posterior probability of a match can be computed as

$$\Pr\left[\xi_{i_1} = \xi_{i_2} \mid \mathbf{P}, \mathbf{Y}\right] = \frac{1}{S}\sum_{s=1}^{S} \mathsf{I}\left[\xi_{i_1}^{(s)} = \xi_{i_2}^{(s)}\right].$$

2.4. Performance assessment

We assess the performance of the estimated linkage structure using conventional precision and recall metrics. Given the ground truth about the linkage structure, there are four possible ways of how predictions about pairs of records can be classified: Correct links (true positives, TP), correct non-links (true negatives, TN), incorrect links (false positives, FP), and incorrect non-links (false negatives, FN). The usual definitions of recall and precision are

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{and} \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

Since most records are unique in practical ER settings, the vast majority of record pairs are typically classified as TN. Hence, we aim to achieve the highest possible recall (true positive rate) while keeping precision (positive predictive value) close to the maximum. In addition, the two measures can be combined into a single metric, the F_1 score, given by

$$\text{F}_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

which is the harmonic mean of precision and recall. As is the case for recall and precision, the F_1 score reaches its best value at 1 and worst at 0.

3. Microclustering

Finite mixture models and Dirichlet/Pitman-Yor process mixture models are widely used in many clustering applications [26]. These models generate cluster sizes that grow linearly with the number of records I, i.e. for all n, $\frac{1}{I}\sum_{i=1}^{I} \mathsf{I}\left[\xi_i = n\right] \xrightarrow{\text{a.s.}} \Pr\left[\xi_i = n\right]$ when $I \to \infty$, where $\mathsf{I}\left[\cdot\right]$ denotes the indicator function. Such a property is not appealing to address de-duplication problems because we need to generate a large number of clusters with a negligible number of records (mostly singletons and pairs).

In order to formulate more realistic models for de-duplication, Miller et al. [25] introduce the concept of microclustering, in which the model is required to produce clusters whose sizes grow sublinearly with I. Formally, a model exhibits the microclustering property if $\frac{M}{I} \xrightarrow{\text{p}} 0$ as $I \to \infty$, where $M = \max\left\{|\text{C}_n| : \text{C}_n \in \mathcal{C}_{\boldsymbol{\xi}}\right\}$ is the size of the largest cluster in $\mathcal{C}_{\boldsymbol{\xi}}$. No mixture model can exhibit the microclustering property, unless its parameters are allowed to vary with I [4].

Miller et al. [25] show that in order to obtain nontrivial models exhibiting the microclustering property, we must sacrifice either finite exchangeability or projectivity. We follow Betancourt et al. [4] in that regard and enforce the former since sacrificing projectivity is less restrictive in the context of ER. As a consequence, inference on $\boldsymbol{\xi}$ will not depend on the order of the data, but the implied joint distribution over a subset of records will not be the same as the joint distribution obtained by modeling the subset directly. Previous work of Wallach et al. [43] sacrifices exchangeability instead.

3.1. Kolchin partition priors

The Kolchin partition priors (KPPs) are originally introduced in Betancourt et al. [4] as a way to enforce the microclustering property. This approach consists in placing a prior on the number of clusters, $N \sim \kappa$, and then, given N, the cluster sizes $S_1, \ldots, S_N$ are modeled directly $S_n \mid N \overset{\text{iid}}{\sim} \mu$, where κ and μ are probability distributions over $\mathbb{N} = \{1, 2, \ldots\}$. In this way, given $I = \sum_{n=1}^{N} S_n$, it is straightforward to generate a set of cluster assignments $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_I)$, which in turn induces a random partition $\mathcal{C}_{\boldsymbol{\xi}} = \{\mathrm{C}_1, \ldots, \mathrm{C}_N\}$, by drawing a vector uniformly at random from the set of permutations of $1, \ldots, 1$ (S_1 times), …, $N, \ldots, N$ (S_N times). Hence, conditioning on I (the total number of records is observed), it can be shown that the probability of any given partition is

$$\mathsf{Pr}\left[\mathcal{C}_{\boldsymbol{\xi}} \mid I\right] \propto |\mathcal{C}_{\boldsymbol{\xi}}|\, \kappa\left(|\mathcal{C}_{\boldsymbol{\xi}}|\right) \prod_{n=1}^{N} |\mathrm{C}_n|!\, \mu\left(|\mathrm{C}_n|\right),$$

where $|\cdot|$ denotes the cardinality of a set. We discuss below two particular choices of κ and μ that have proven to exhibit the microclustering property. We use these alternatives as a baseline in Section 4. We remit the reader to Miller et al. [25], Betancourt et al. [4], Betancourt et al. [5], and Betancourt et al. [3] for details about computation and prior elicitation.

The **Negative Binomial–Negative Binomial Prior** (NBNBP) assumes that both κ and μ are negative binomial distributions (truncated to $\mathbb{N}$) with parameters a and q and η and θ, respectively. Here, $a > 0$ and $q \in (0, 1)$ are fixed hyperparameters, while $\eta > 0$ and $\theta \in (0, 1)$ are distributed as $\eta \sim \mathsf{Gamma}(a_\eta, b_\eta)$ and $\theta \sim \mathsf{Beta}(a_\theta, b_\theta)$, for fixed hyperparameters a_η, b_η, a_θ, and b_θ. When evaluating the performance of this prior, we follow the authors and set a and q in a way that $\mathsf{E}[N] = \sqrt{\mathsf{Var}[N]} = \frac{I}{2}$, $a_\eta = b_\eta = 1$, and $a_\theta = b_\theta = 2$.

The **Negative Binomial–Dirichlet Prior** (NBDP) still assumes that κ is a Negative Binomial distribution (truncated to $\mathbb{N}$) with parameters a and q, but this time $\mu \sim \mathsf{DP}(\alpha, \mu^0)$, where DP denotes the Dirichlet Process (see [27] for a formal treatment of the DP). Here, a and q are once again fixed hyperparameters, α is a concentration parameter and μ^0 is a base measure with $\sum_{k=1}^{\infty} \mu^0(k) = 1$ and $\mu^0(k) \geq 0$, for all k. The parameters a and q are set as before, while $\alpha = 1$ and μ^0 is set to be a Geometric distribution over $\mathbb{N}$ with parameter 0.5.

3.2. Allelic partition priors

Here, we consider a class of prior distributions on the cluster assignments $\boldsymbol{\xi}$ based on allelic partitions [12]. Very recently, such a class of priors has been fully developed and characterized in Betancourt et al. [3] in the context of just attribute data. Let $\mathcal{C}_{\boldsymbol{\xi}} = \{\mathrm{C}_1, \ldots, \mathrm{C}_N\}$ be the partition implicitly represented by the linkage structure $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_I)$, and also, let $\boldsymbol{r} = (r_1, \ldots, r_I)$ be the allelic partition induced by $\mathcal{C}_{\boldsymbol{\xi}}$, where r_i denotes the number of clusters of size i in $\mathcal{C}_{\boldsymbol{\xi}}$.

Assuming that partitions corresponding to the same allelic partition occur with the same probability, we can generate a random partition by first drawing an allelic partition uniformly at random, and then, selecting uniformly at random again among partitions for

which that specific allelic partition holds. This simple reasoning allow us to write

$$p(\boldsymbol{\xi}) = \frac{1}{I!} \prod_{i=1}^{I} i!^{r_i} \, r_i! \times p(\boldsymbol{r}),$$

which fully determines an exchangeable partition probability function. Thus, we just need to place a distribution on $\boldsymbol{r}$ in order to complete the prior specification. To do so, a natural way to proceed consists in factorizing the distribution on $\boldsymbol{r}$ as

$$p(\boldsymbol{r} \mid M) = p(r_M)\, p(r_{M-1} \mid r_M)\, p(r_{M-2} \mid r_{M-1}, r_M) \, \ldots \, p(r_1 \mid r_2, \ldots, r_M),$$

and let

$$r_M \sim \mathsf{Binomial}(Q_M, \theta_M),$$
$$r_k \mid r_{k+1}, \ldots, r_M \sim \mathsf{Binomial}(Q_k, \theta_k), \quad r_1 \mid r_2, \ldots, r_M \sim \delta_{Q_1},$$

where $Q_M = \lfloor I/M \rfloor$, $Q_k = \lfloor (I - \sum_{i=k+1}^{M} i\, r_i)/k \rfloor$ for $k = 2, \ldots, M-1$, and $Q_1 = I - \sum_{i=2}^{M} i\, r_i$, and $\lfloor \cdot \rfloor$ denotes the floor of scalar.

We consider the **Allelic Binomial Prior** (ABP), setting the maximum cluster size to $M = 2$, which leads to the allelic partition $\boldsymbol{r} = (I - 2r_2, r_2, 0, 0, \ldots, 0)$, where r_2 is the number of clusters of size 2, and then, letting $r_2 \sim \mathsf{Beta\text{-}Binomial}(a_2, b_2)$. This approach guarantees that the microclustering property holds, because the value of M is being handled directly. We let $a_2 = \frac{\rho - \gamma^2}{(1+\rho)\gamma^2}$ and $b_2 = a_2\, \rho$, with $\rho = (1-\pi)/\pi$, where $\pi = 0.8$ is the prior probability of expecting a singleton, and $\gamma = 0.5$ is the corresponding coefficient of variation.

3.3. Ewens-Pitman priors

Finally, another popular alternative that does not satisfy the microclustering property but is convenient for practical reasons, is the Ewens-Pitman Prior (EPP, McCullagh and Yang [24]). The probability mass function for the EPP is given by $p(\boldsymbol{\xi} \mid \theta) = \frac{\Gamma(\theta)}{\Gamma(I+\theta)}\, \theta^N \prod_{n=1}^{N} \Gamma(S_n)$, with $\theta \sim \mathsf{Gamma}(a_\theta, b_\theta)$. The parameters a_θ and b_θ need to be carefully chosen in order to match the prior beliefs given in the ABP.

4. Evaluation

We investigate the impact of including relational data in the de-duplication process as well as the performance of the ABP compared to other existing priors. To this end, we consider the RLdata500 dataset from the RecordLinkage package [6] in R, which has been considered by many authors to test their methodologies, including Christen and Pudjijono [10], Christen and Vatsalan [11], Steorts et al. [41], Steorts [39], and Tancredi et al. [42]. This is a syntectic dataset with $I = 500$ records, 50 of which are duplicates. Each record has associated with it seven fields, namely, name's first component, name's second component, last name's first component, last name's second component, year of birth, month of birth, and day of birth. We only consider the last three fields (categorical fields) for illustrative purposes. The ground truth (true cluster assignments) is also available.

Table 1. Structural features of the network data.

Scenario	β	σ^2	K	Transitivity	Assortativity	Density
Scenario 1	10	178	2	0.576	0.680	0.126
Scenario 2	10	278	2	0.562	0.754	0.082

We augment this dataset by generating social ties between records following the latent distance model (2) along with the true linkage structure in the dataset. We consider two scenarios (see Table 1), which allows us to study how structural network features influence the de-duplication process.

We fit our de-duplication model using just profile data as well as using both profile and network data with $K = 2$. We also implement each prior specification given in Section 3, along with an uniform prior (UP) as in Steorts et al. [40]. In particular, we calibrate the ABP, in such a way that 80% and 50% of clusters are a priori singleton clusters with a 0.5 coefficient of variation for $M = 2$ (ABP1 and ABP2, respectively). The EPP is calibrated in the same way. We also calibrate the ABP around 80% of singleton clusters with a 0.5 coefficient of variation for $M = 3$ (ABP3). Histograms of the number of singleton clusters for some of these prior distributions are shown in Figure 1. Lastly, we run the Gibbs sampler described in Appendix 1 based on 100,000 samples obtained after a burn-in period of 500,000 iterations. We decided to use such a long burn-in period in order

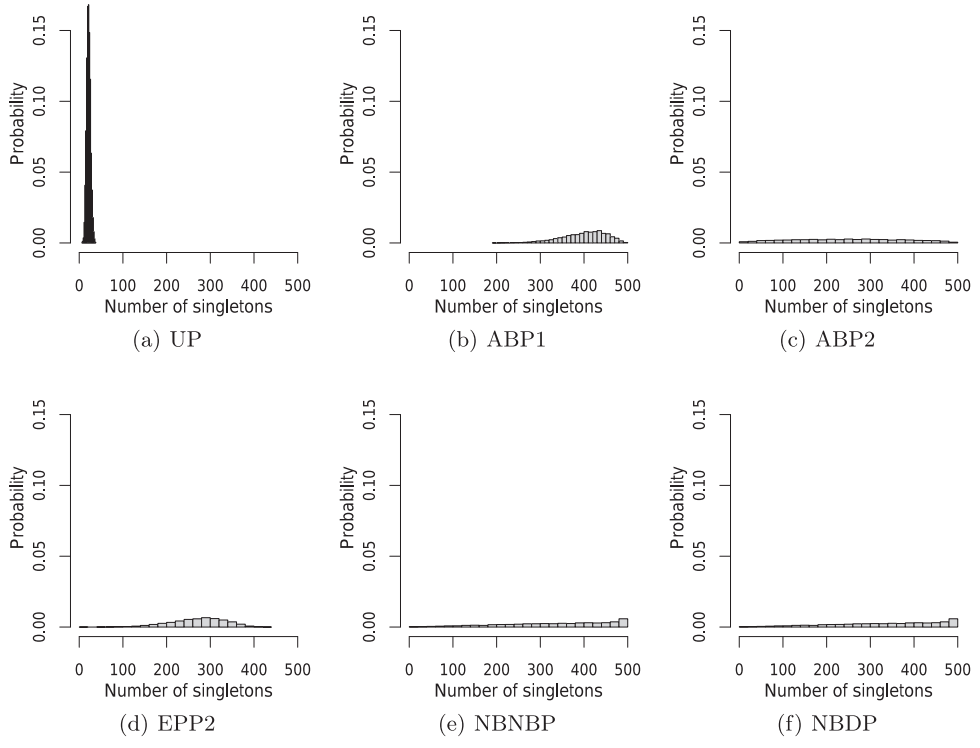

Figure 1. Prior distribution of the number of singleton clusters. (a) UP. (b) ABP1. (c) ABP2. (d) EPP2. (e) NBNBP. (f) NBDP.

Table 2. Performance assessment and summary statistics for each prior distribution using just profile data and using both profile and network data.

Prior	Recall	Precision	F_1	$E[N \mid \text{data}]$	$SD[N \mid \text{data}]$
Profile data					
UP	0.62	0.45	0.52	264.78	2.63
ABP1	**0.58**	**0.88**	**0.70**	**467.86**	**2.83**
ABP2	**0.58**	**0.88**	**0.70**	**467.97**	**2.85**
ABP3	**0.58**	**0.88**	**0.70**	**468.43**	**3.85**
EPP1	0.02	0.50	0.04	497.97	0.74
EPP2	0.06	1.00	0.11	492.06	1.87
NBDP	0.58	0.88	0.70	469.19	2.66
NBNBP	0.58	0.88	0.70	467.89	2.62
Profile and network data (Scenario 1)					
UP	1.00	0.32	0.49	344.12	1.56
ABP1	**0.94**	**0.94**	**0.94**	**450.20**	**2.28**
ABP2	**0.94**	**0.92**	**0.93**	**450.35**	**1.12**
ABP3	**0.94**	**0.94**	**0.94**	**447.81**	**0.77**
EPP1	0.84	0.81	0.82	449.38	2.19
EPP2	0.80	0.85	0.82	450.74	3.19
NBDP	0.94	0.71	0.81	445.66	2.46
NBNBP	0.92	0.82	0.87	441.45	1.47
Profile and network data (Scenario 2)					
UP	0.94	0.31	0.47	346.34	1.91
ABP1	**0.90**	**0.92**	**0.91**	**450.32**	**1.23**
ABP2	**0.90**	**0.94**	**0.92**	**451.85**	**1.44**
ABP3	**0.94**	**0.92**	**0.93**	**448.28**	**0.45**
EPP1	0.76	0.70	0.73	447.20	4.53
EPP2	0.84	0.78	0.81	448.85	2.67
NBDP	0.92	0.82	0.87	441.34	2.29
NBNBP	0.90	0.75	0.82	443.50	1.69

to avoid any kind of convergence issue, since we are dealing with several prior distributions over a large set of discrete parameters. In our experience, this period can be reduced greatly in order to save computation time. Finally, we use the clustering methodology proposed by Lau and Green [22] to obtain a point estimate $\hat{\boldsymbol{\xi}}$ of the linkage structure, which employs a decision-theoretic approach based on the minimization of posterior expected losses through pairwise clustering probabilities. Lastly, computational code to reproduce all our findings is available at https://github.com/jstats1702/NetDedup.

We report the results of our experiments in Table 2. When the model is fitted using only profile data, the recall of the procedure is relatively intermediate. There seems to be no difference between the ABPs and the KPPs in this setting. On the other hand, notice that the EPP's behavior is particularly poor. This fact suggests that satisfying the microclustering property is crucial, specially when only profile information is available and the number of fields is small. Even though the UP's recall seems higher, its precision is substantially low. In general, the population size is being overestimated; this is not the case for the UP because it has such a strong pull towards a small number of singletons as shown in Figure 1.

As expected, including network data substantially improves the accuracy of the posterior linkage as well as the estimate of the population size, especially in cases like these, where profile data is not abundant. In general, every prior seems to favor a fair estimate of the population size, except the UP. On the other hand, looking at the F_1 score, the models based on ABPs clearly outperform the rest. Interestingly, there is not much difference in performance between ABP1 and ABP2. Not surprisingly, those priors that do not satisfy

the microclustering property perform worse than those that do. Notice also that the ABP produces similar results for both $M = 2$ and $M = 3$. Lastly, it seems to be the case that accuracy values tend to decrease a little when the network data is less dense. This feature is more evident for the EPP.

5. Sensitivity analysis

We fitted our de-duplication model making specific choices for several quantities. Specifically, we chose $\psi_\ell \overset{\text{iid}}{\sim} \text{Beta}(a_\ell, b_\ell)$, with $a_\ell = a = 1$ and $b_\ell = b = 99$, which corresponds to a prior mean of 0.01 for the distortion probabilities. Typically, this is a sensible choice because distortion probabilities cannot be close to 1 (only a small number of corrupted fields is expected), and based on our choice of the Beta distribution, it follows that $b >> 1$. Here, we consider the effect of varying the values of a and b on the posterior linkage and the estimate of the population size N. To this end, we fit our model again using both profile and network data along with the ABP2 as a prior distribution for the linkage structure.

In the same spirit of spirit of Steorts [39], we explore several cases to assess the robustness of our model to the choice of a and b. First, we fix the prior mean of each distortion probability at $a/(a + b) = 0.002$ (instead of 0.01) and vary a and b proportionally, which decreases the variance of the prior distribution. Then, we consider the effect of varying the prior mean $a/(a + b)$ while holding $a + b$ fixed at either $a + b = 100$ or $a + b = 10$. Results are shown in Table 3.

We see that these results are fairly consistent to those presented in the second panel of Table 2, although there is a non-negligible improvement when $a = 0.1$ and $b = 49.9$; such a setting makes both recall and precision almost perfect as well as the estimate of the population size. On the other hand, precision tends to decrease when the prior variance of

Table 3. Performance assessment and summary statistics for the ABP2 using both profile and network data (Scenario 1).

a	b	Recall	Precision	F_1	$E[N \mid \text{data}]$	$SD[N \mid \text{data}]$
$a/(a+b) = 0.002$						
0.004	1.996	0.94	0.90	0.92	447.48	0.57
0.010	4.990	0.94	0.87	0.90	445.79	0.61
0.020	9.980	0.94	0.84	0.89	442.42	1.02
0.040	19.960	0.96	0.89	0.92	445.39	1.62
0.100	**49.900**	**0.96**	**0.96**	**0.96**	**449.26**	**0.57**
0.200	99.800	0.98	0.91	0.94	446.46	0.50
$a+b = 100$						
0.030	99.970	0.96	0.92	0.94	448.02	1.07
0.100	99.900	0.92	0.84	0.88	444.57	1.75
0.300	99.700	0.96	0.91	0.93	446.56	1.83
1.000	99.000	0.94	0.92	0.93	450.35	1.12
3.000	97.000	0.96	0.89	0.92	445.51	0.86
10.000	90.000	0.94	0.82	0.88	441.76	0.85
$a+b = 10$						
0.003	9.997	0.96	0.91	0.93	446.56	0.69
0.010	9.990	0.96	0.89	0.92	445.20	0.62
0.030	9.970	0.92	0.92	0.92	449.39	1.08
0.100	9.900	0.94	0.87	0.90	445.37	0.73
0.300	9.700	0.92	0.92	0.92	449.17	0.37
1.000	9.000	0.96	0.81	0.88	440.54	2.55

the distortion probabilities increases, e.g. $a = 10$ and $b = 90$, and also $a = 1$ and $b = 9$; prior specifications of this kind also lead to an underestimate of the population size. These findings suggest that our approach is quite robust to the prior specification of the distortion probabilities.

6. An alternative way to draw samples for the network parameters

Suppose we want to generate samples from the posterior distribution of $\boldsymbol{\theta}$ given a set of independent observations $\boldsymbol{x} \in \mathrm{D}$, $p(\boldsymbol{\theta} \mid \mathrm{D}) \propto \exp(-U(\boldsymbol{\theta}))$, where the potential energy function U is given by $U(\boldsymbol{\theta}) = -\sum_{\boldsymbol{x}\in\mathrm{D}} \log p(\boldsymbol{x} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$. A Hamiltonian Monte Carlo (HMC) algorithm introduces a set of auxiliary variables $\boldsymbol{r}$ and draws samples from the joint distribution $p(\boldsymbol{\theta}, \boldsymbol{r}) \propto \exp\left(-U(\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{r}^{\mathsf{T}}\mathbf{M}\boldsymbol{r}\right)$, by simulating from a Hamiltonian system, where $\mathbf{M}$ is a mass matrix usually specified as the identity matrix. If we simply discard the resulting $\boldsymbol{r}$ samples, the $\boldsymbol{\theta}$ samples have the desired marginal distribution $p(\boldsymbol{\theta} \mid \mathrm{D})$. See Neal [30] for details.

Now, along the lines of Chen et al. [9], instead of computing the gradient $\nabla U(\boldsymbol{\theta})$ using the entire dataset D, the stochastic gradient HMC (SGHMC) considers a noisy estimate based on a minibatch $\tilde{\mathrm{D}}$ sampled uniformly at random from D:

$$\nabla \tilde{U}(\theta) = -\frac{|\mathrm{D}|}{|\tilde{\mathrm{D}}|} \sum_{\boldsymbol{x}\in\tilde{\mathrm{D}}} \log p(\boldsymbol{x} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta})\,.$$

Clearly, we want minibatches to be small in order to obtain a significant reduction in the computational cost of $\nabla U(\boldsymbol{\theta})$. Details about the SGHMC are provided in Appendix 1.

We want to compare a random-walk (RW) and a SGHMC in terms of accuracy and computational cost using the data provided in Section 4. Once again we fit our de-duplication model using both profile and network data, and the ABP2 as a prior distribution for the linkage structure. To do so, we follow the algorithm outlined in Appendix 1 using both a RW and a SGHMC to sample from the conditional distribution of β and each $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$. The RW adaptively finds the value of the tunning parameter in order to automatically find a good proposal distribution. Regarding the SGHMC, we set the mass matrix $\mathbf{M}$ to the identity matrix; after some experimentation, we decided to make the scaling factor $\epsilon = 0.001$ and the number of leapfrogs steps $L = 5$. Such values provide reasonable acceptance rates in this case. Lastly, minibatches are chosen by sampling uniformly at random 20% of the corresponding data points. We run both algorithms based on 100,000 samples obtained after a burn-in period of 500,000 iterations.

Table 4 shows the corresponding results. We see that the SGHMC provides sensible levels of accuracy in comparison with the RW. In particular, both approaches yield to

Table 4. Performance assessment and summary statistics for the ABP2 using both profile and network data (Scenario 1).

Algorithm	Recall	Precision	F_1	$\mathrm{E}[N \mid \text{data}]$	$\mathrm{SD}[N \mid \text{data}]$	Time sec/100
RW	0.94	0.92	0.93	450.35	1.12	9.05
SGHMC	0.96	0.72	0.82	425.59	4.96	5.16

Notes: Time is given in seconds per 100 iterations using a standard laptop with 16 GB of RAM and a 2.60 GHz Intel Core i7 processor.

extremely good recall values. Even though we loose some precision with the SGHMC, we reduce the computation time around 43%. These results are comparable with those in Table 2, where fitting the model using other prior distributions such as the EPP and the KPPs produces similar levels of accuracy.

7. Discussion

We have proposed a novel approach for de-duplication that easily reconciles both profile and network data. We have implemented a new prior specification on the cluster assignments, the ABP, which is easy to implement, naturally satisfies the microclustering property, and also makes it straightforward to incorporate prior believes about the linkage structure. We have also considered stochastic gradient Hamiltonian Monte Carlo methods in order to speed up the de-duplication process maintaining reasonable levels of accuracy. Our experiments show that our formulation is quite robust to prior specification and outperforms its competitors by substantially improving the accuracy of the posterior linkage, and as a consequence, the estimate of the population size as well.

As part of the revision process, referees were concerned about the slow convergence rate of our original sampling scheme. Certainly, this can be the case depending on a number of factors, including how disturbed the dataset is, the number of latent variables in the model, and the discrete nature of the linkage structure. In part, this issue motivated our SGHMC algorithm, which is much faster than our original MCMC algorithm, while maintaining reasonable levels of accuracy. However, there is certainly room for improvement. On the one hand, chain mixing can be substantially improved by considering alternative sampling approaches for the cluster assignments, in the flavor of either split-merge [19] or chaperons [25] algorithms. On the other hand, the latent field values can be integrated out [3] in order to avoid some computational burden as well. These alternatives will be pursued elsewhere.

Finally, there are several opportunities for future research. Our approach relies on a relatively simple model for the network data. Natural extensions range from using a mixture prior for the latent positions to allow for community structure in the network (e.g. Handcock et al. [17]) to considering a non-Euclidean social spaces (e.g. see Smith et al. [36]). On the other hand, we could weaken the independence assumption by adding an extra layer of structure to account for dependencies between profile and network data. In addition, we could also either add an extra hierarchy to model the size of the larger cluster in a way that microclustering is preserved or consider a different distribution for the allelic partition. Lastly, it also may be worth considering other fast approximation techniques in the flavor of variational approximations [2,7,20,34]. This would allow us to consider bigger datasets with even millions of records.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Juan Sosa http://orcid.org/0000-0001-7432-4014
Abel Rodríguez http://orcid.org/0000-0001-5503-7394

References

[1] S. Aleshin-Guendel and M. Sadinle, *Multifile partitioning for record linkage and duplicate detection*, J. Am. Stat. Assoc. (2022), pp. 1–10. Available at https://doi.org/10.1080/01621459.2021.2013242.

[2] M.J. Beal, *Variational Algorithms for Approximate Bayesian Inference*, University of London, University College London (United Kingdom), 2003.

[3] B. Betancourt, J. Sosa, and A. Rodríguez, *A prior for record linkage based on allelic partitions*, Comput. Stat. Data. Anal. 172 (2022), pp. 107474.

[4] B. Betancourt, G. Zanella, J.W. Miller, H. Wallach, A. Zaidi, and R.C. Steorts, *Flexible models for microclustering with application to entity resolution*, Adv. Neural. Inf. Process. Syst. 29 (2016), pp. 1417–1425.

[5] B. Betancourt, G. Zanella, and R.C. Steorts, *Random partition models for microclustering tasks*, J. Am. Stat. Assoc. (2020), pp. 1–13. Available at https://doi.org/10.1080/01621459.2020.1841647.

[6] A. Borg and M Sariyar, *RecordLinkage: Record Linkage in R*. 2016, R package version 0.4-10.

[7] T. Broderick and R.C Steorts, *Variational Bayes for merging noisy databases*, preprint (2014). Available at arXiv:1410.4792.

[8] G. Casella, E. Moreno, and F.J. Girón, *Cluster analysis, model selection, and prior distributions on models*, Bayesian Analysis 9 (2014), pp. 613–658.

[9] T. Chen, E. Fox, and C Guestrin, *Stochastic gradient Hamiltonian Monte Carlo*, International Conference on Machine Learning, PMLR, 2014, pp. 1683–1691.

[10] P. Christen and A Pudjijono, *Accurate synthetic generation of realistic personal information*, Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2009, pp. 507–514.

[11] P. Christen and D Vatsalan, *Flexible and extensible generation and corruption of personal data*, Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, pp. 1165–1168.

[12] H. Crane, *The ubiquitous Ewens sampling formula*, Stat. Sci. 31 (2016), pp. 1–19.

[13] P Domingos, *Multi-relational record linkage*, Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining, Citeseer, 2004.

[14] T. Enamorado and R.C Steorts, *Probabilistic blocking and distributed Bayesian entity resolution*, International Conference on Privacy in Statistical Databases, Springer, 2020, pp. 224–239.

[15] T. Ferguson, *A bayesian analysis of some nonparametric problems*, Ann. Stat. 1 (1973), pp. 209–230.

[16] D. Gamerman and H.F. Lopes, *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*, CRC Press, Boca Raton, FL, 2006.

[17] M.S. Handcock, A.E. Raftery, and J.M. Tantrum, *Model-based clustering for social networks*, J. R. Stat. Soc.: Ser. A (Stat. Soc.) 170 (2007), pp. 301–354.

[18] P.D. Hoff, A.E. Raftery, and M.S. Handcock, *Latent space approaches to social network analysis*, J. Am. Stat. Assoc. 97 (2002), pp. 1090–1098.

[19] S. Jain and R.M. Neal, *A split-merge Markov chain Monte Carlo procedure for the dirichlet process mixture model*, J. Comput. Graph. Stat. 13 (2004), pp. 158–182.

[20] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul, *An introduction to variational methods for graphical models*, Mach. Learn. 37 (1999), pp. 183–233.

[21] P.N. Krivitsky and M.S. Handcock, *Fitting latent cluster models for networks with latentnet*, J. Stat. Softw. 24 (2008), pp. 1–23.

[22] J.W. Lau and P.J. Green, *Bayesian model-based clustering procedures*, J. Comput. Graph. Stat. 16 (2007), pp. 526–558.

[23] N.G. Marchant, A. Kaplan, D.N. Elazar, B.I. Rubinstein, and R.C. Steorts, *d-blink: Distributed end-to-end Bayesian entity resolution*, J. Comput. Graph. Stat. 30 (2021), pp. 406–421.

[24] P. McCullagh and J Yang, *Stochastic classification models*, International Congress of Mathematicians, Vol. 3, Citeseer, 2006, pp. 72–145.

[25] J. Miller, B. Betancourt, A. Zaidi, H. Wallach, and R.C Steorts, *Microclustering: When the cluster sizes grow sublinearly with the size of the data set*, preprint (2015). Available at arXiv:1512.00792.

[26] J.W. Miller and M.T. Harrison, *Mixture models with a prior on the number of components*, J. Am. Stat. Assoc. 113 (2018), pp. 340–356.
[27] P. Müller and A. Rodríguez, *Nonparametric Bayesian Inference*, Institute of Mathematical Statistics, 2013. Available at https://imstat.org/overview/.
[28] A. Narayanan and V Shmatikov, *De-anonymizing social networks*, 2009 30th IEEE Symposium on Security and Privacy, IEEE, 2009, pp. 173–187.
[29] R.M. Neal, *Markov chain sampling methods for dirichlet process mixture models*, J. Comput. Graph. Stat. 9 (2000), pp. 249–265.
[30] R.M. Neal, *MCMC using Hamiltonian dynamics*, Handbook of Markov chain Monte Carlo, Vol. 2, 2011, pp. 114–162,
[31] J.S Rosenthal, *Optimal proposal distributions and adaptive MCMC*, Handbook of Markov Chain Monte Carlo, 4(10.1201). 2011.
[32] M. Sadinle, *Detecting duplicates in a homicide registry using a Bayesian partitioning approach*, Ann. Appl. Stat. 8 (2014), pp. 2404–2434.
[33] M. Sadinle and S.E. Fienberg, *A generalized Fellegi–Sunter framework for multiple record linkage with application to homicide record systems*, J. Am. Stat. Assoc. 108 (2013), pp. 385–397.
[34] L.K. Saul, T. Jaakkola, and M.I. Jordan, *Mean field theory for sigmoid belief networks*, J Artif Intell Res 4 (1996), pp. 61–76.
[35] J. Sethuraman, *A constructive definition of dirichlet priors*, Stat. Sin. 4 (1994), pp. 639–650.
[36] A.L. Smith, D.M. Asta, and C.A. Calder, *The geometry of continuous latent space models for network data*, Stat. Sci. 34 (2019), pp. 428–453.
[37] J. Sosa and L. Buitrago, *A review of latent space models for social networks*, Revista Colombiana De Estadística 44 (2021), pp. 171–200.
[38] J. Sosa and A Rodríguez, *A record linkage model incorporating relational data*, preprint (2018). Available at arXiv:1808.04511.
[39] R.C. Steorts, *Entity resolution with empirically motivated priors*, Bayesian Analysis 10 (2015), pp. 849–875.
[40] R.C. Steorts, R. Hall, and S.E. Fienberg, *A Bayesian approach to graphical record linkage and deduplication*, J. Am. Stat. Assoc. 111 (2016), pp. 1660–1672.
[41] R.C. Steorts, S.L. Ventura, M. Sadinle, and S.E Fienberg, *A comparison of blocking methods for record linkage*, International Conference on Privacy in Statistical Databases, Springer, 2014, pp. 253–268.
[42] A. Tancredi, R. Steorts, and B. Liseo, *A unified framework for de-duplication and population size estimation (with discussion)*, Bayesian Analysis 15 (2020), pp. 633–682.
[43] H. Wallach, S. Jensen, L. Dicker, and K Heller, *An alternative prior process for nonparametric Bayesian clustering*, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2010, pp. 892–899.

Appendices

Appendix 1. Computation for ER model

A.1 Markov chain Monte Carlo

Taking this into account that $\pi_{n,\ell}$, $w_{i,\ell}$ and ξ_i are all interconnected, since if $w_{i,\ell} = 0$, then it must be the case that $\pi_{\xi_i,\ell} = p_{i,\ell}$, the joint posterior reduces to

$$
\begin{aligned}
p(\boldsymbol{\Upsilon} \mid \mathbf{Y}, \mathbf{P}) \propto & \prod_{i<i'} \theta_{i,i'}^{y_{i,i'}} (1-\theta_{i,i'})^{1-y_{i,i'}} \times \exp\left(-\tfrac{1}{\omega^2}\,\beta^2\right) \\
& \times \prod_{n} (\sigma^2)^{-K/2} \exp\left(-\tfrac{1}{\sigma^2}\,\|\boldsymbol{u}_n\|^2\right) \times (\sigma^2)^{-(a_\sigma+1)} \exp\left(-\frac{b_\sigma}{\sigma^2}\right) \\
& \times \prod_{i,\ell,m} \left((1-w_{i,\ell})\,\delta_{\pi_{\xi_i,\ell}}(p_{i,\ell}) + w_{i,\ell}\,\vartheta_{\ell,m}^{\mathbb{I}[p_{i,\ell}=m^{(i)}]} \right)
\end{aligned}
$$

$$\times \prod_{i,\ell} \psi_\ell^{w_{i,\ell}} (1-\psi_\ell)^{1-w_{i,\ell}} \times \prod_{n,\ell,m} \vartheta_{\ell,m}^{\mathsf{I}[\pi_{n,\ell}=m^{(i)}]}$$
$$\times \prod_{\ell,m} \vartheta_{\ell,m}^{\alpha_{\ell,m}-1} \times \prod_{\ell} \psi_\ell^{a_\ell-1}(1-\psi_\ell)^{b_\ell-1} \times p(\boldsymbol{\xi}),$$

where $\theta_{i,i'} = \text{expit}(\beta - \|\boldsymbol{u}_{\xi_i} - \boldsymbol{u}_{\xi_{i'}}\|)$, and $m^{(i)}$ is the (category) index in which $p_{i,\ell}$ is having a value.

Our MCMC algorithm iterates over the model parameters $\boldsymbol{\Upsilon}$. Where possible we sample from the full conditional posterior distributions as in a Gibbs sampling algorithm. Otherwise, we use Metropolis-Hastings steps. The MCMC algorithm proceeds by generating a new state $\boldsymbol{\Upsilon}^{(s+1)}$ from a current state $\boldsymbol{\Upsilon}^{(s)}$ as follows:

(1) Sample $\xi_i^{(s+1)}$ following Algorithm 5 provided in [29]. Repeat the following update of ξ_i R times:
   (i) Draw a proposal, ξ_i^*, uniformly at random from the set of values or following a (fixed) distribution independent of the current state of the Markov chain. However, crucially, it may depend on the observed data. The proposal ξ_i^* must leave the members of $\mathcal{C}_{\boldsymbol{\xi}}$ only having one (singletons) or two (pairs) records from different files.
   (ii) If ξ_i^* stars a new cluster, sample a value for $\pi_{\xi_i^*,\ell}$ from $\mathsf{Categorical}(\boldsymbol{\vartheta}_\ell)$ for each ℓ, and equivalently, sample a value for $u_{\xi_i^*,k}$ from $\mathsf{Normal}(0,\sigma^2)$ for each k.
   (iii) Compute the acceptance probability

$$a = \min\left[1, \frac{p(\mathbf{Y}, \boldsymbol{u}_{\xi_i^*} \mid \text{rest})}{p(\mathbf{Y}, \boldsymbol{u}_{\xi_i^{(s)}} \mid \text{rest})}\right].$$

   (iv) Let

$$\xi_i^{(s+1)} = \begin{cases} \xi_i^*, & \text{with probability } a; \\ \xi_i^{(s)}, & \text{with probability } 1-a. \end{cases}$$

   (v) If $\xi_i^{(s+1)} = \xi_i^*$, then $w_{i,\ell}$ has to be sampled accordingly; the same way that $\pi_{\xi_i^*,\ell}$ and $u_{\xi_i^*,k}$ need to be updated with the values drawn from the prior in ii.

   This step depends on the form of $p(\boldsymbol{\xi})$ and may involve additional parameters that also need to be sampled. See details below.

(2) Sample $w_{i,\ell}^{(s+1)}$ from $p(w_{i,\ell} \mid \text{rest}) = \mathsf{Bernoulli}(w_{i,\ell} \mid \tilde{\theta}_{i,\ell})$, where

$$\tilde{\theta}_{i,\ell} = \begin{cases} 1, & \text{if } p_{i,\ell} \neq \pi_{\xi_i,\ell}; \\ \dfrac{\psi_\ell \prod_m \vartheta_{\ell,m}^{\mathsf{I}[p_{i,\ell}=m^{(i)}]}}{\psi_\ell \prod_m \vartheta_{\ell,m}^{\mathsf{I}[p_{i,\ell}=m^{(i)}]} + (1-\psi_\ell)}, & \text{if } p_{i,\ell} = \pi_{\xi_i,\ell}. \end{cases}$$

(3) Sample $\pi_{n,\ell}^{(s+1)}$ from $p(\pi_{n,\ell} \mid \text{rest}) = \delta_{p_{i,\ell}}(\pi_{n,\ell} \mid p_{i,\ell})$ if there exists a file j and a record $i \in \mathrm{C}_n$ such that $w_{i,\ell} = 0$, where $\mathrm{C}_n = \{i \in \{1,\ldots,I\} : \xi_i = n\}$, i.e. C_n is the set of all records i linked to the latent individual n. Otherwise, sample $\pi_{n,\ell}^{(s+1)}$ from $p(\pi_{n,\ell} \mid \text{rest}) = \mathsf{Categorical}(\pi_{n,\ell} \mid \boldsymbol{\vartheta}_\ell)$.

(4) Sample $\boldsymbol{\vartheta}_\ell^{(s+1)}$ from $p(\boldsymbol{\vartheta}_\ell \mid \text{rest}) = \mathsf{Dirichlet}(\boldsymbol{\vartheta}_\ell \mid \tilde{\boldsymbol{\alpha}})$, where

$$\tilde{\alpha}_m = \alpha_{\ell,m} + \sum_n \mathsf{I}\left[\pi_{n,\ell} = m\right] + \sum_i w_{i,\ell}\, \mathsf{I}\left[p_{i,\ell} = m\right].$$

(5) Sample $\psi_\ell^{(s+1)}$ from $p(\psi_\ell \mid \text{rest}) = \mathsf{Beta}(\psi_\ell \mid a_\ell + \sum_i w_{i,\ell}, b_\ell + I - \sum_i w_{i,\ell})$.
   (a) Sample $\boldsymbol{u}_n^{(s+1)}$:
      (i) Draw a proposal, $\boldsymbol{u}_n^*$, from $\mathsf{Normal}(\boldsymbol{u}_n^{(s)}, \delta^2 \mathbf{I}_K)$, where δ^2 is a tuning parameter.

(ii) Compute the acceptance probability

$$a = \min\left[1, \frac{p(\mathbf{Y}, \boldsymbol{u}_n^* \mid \text{rest})}{p(\mathbf{Y}, \boldsymbol{u}_n^{(s)} \mid \text{rest})}\right].$$

(iii) Let

$$\boldsymbol{u}_n^{(s)} = \begin{cases} \boldsymbol{u}_n^*, & \text{with probability } a; \\ \boldsymbol{u}_n^{(s)}, & \text{with probability } 1-a. \end{cases}$$

(b) Sample $\beta^{(s+1)}$:

(i) Draw a proposal, β^*, from $\mathsf{Normal}(\beta^{(s)}, \delta^2)$, where δ^2 is a tuning parameter.

(ii) Compute the acceptance probability

$$a = \min\left[1, \frac{p(\mathbf{Y}, \beta^* \mid \text{rest})}{p(\mathbf{Y}, \beta^{(s)} \mid \text{rest})}\right].$$

(iii) Let

$$\beta^{(s+1)} = \begin{cases} \beta^*, & \text{with probability } a; \\ \beta^{(s)}, & \text{with probability } 1-a. \end{cases}$$

(c) Sample $(\sigma^2)^{(s)}$ from $p(\sigma^2 \mid \text{rest}) = \mathsf{Inverse\text{-}Gamma}\left(\sigma^2 \mid a_\sigma + \frac{1}{2}NK, b_\sigma + \frac{1}{2}\sum_n \|\boldsymbol{u}_n\|^2\right)$.

The value of the tunning parameter δ^2 in the proposal distribution is chosen to make the algorithm run efficiently. We adaptively change the value of δ^2 at the beginning of the chain in order to automatically find a good proposal distribution. See Rosenthal [31] for a review about optimal proposal scalings for Metropolis-Hastings MCMC algorithms and adaptive MCMC algorithms.

No further steps are required when the UP is considered as the prior distribution for $\boldsymbol{\xi}$. On the one hand, if the ABP2 is used, note that in step 1

$$p(\boldsymbol{\xi} \mid \text{rest}) \propto (I-2r_2)!\, 2^{r_2}\, r_2! \binom{Q_2}{r_2} \theta_2^{r_2} (1-\theta_2)^{Q_2-r_2},$$

where $Q_2 = \lfloor I/2 \rfloor$, and to complete the sampler, we need to add the following step to the algorithm:

(6) Sample $\theta_2^{(s+1)}$ from $p(\theta_2 \mid \text{rest}) = \mathsf{Beta}(\theta_2 \mid a_2 + r_2, b_2 + Q_2 - r_2)$.

On the other hand, if the EPP is used, note that $p(\boldsymbol{\xi} \mid \text{rest}) \propto \theta^N \prod_{n=1}^N \Gamma(S_n)$ in step 1, and to complete the sampler, we need to introduce an auxiliary variable η such that

$$p(\theta, \eta \mid \text{rest}) \propto p(\theta)\, \theta^{N-1}(\theta + I) \times \eta^\theta (1-\eta)^{I-1}.$$

By doing so, we need to add the following step to the algorithm:

(7) Sample $\theta^{(s+1)}$ from the two-component gamma mixture:

$$\begin{aligned} p(\theta \mid \text{rest}) = {} & \epsilon\, \mathsf{Gamma}(\theta \mid a_\theta + N, b_\theta - \log\eta) \\ & + (1-\epsilon)\mathsf{Gamma}(\theta \mid a_\theta + N - 1, b_\theta - \log\eta) \end{aligned}$$

where $\epsilon = \frac{a_\theta + N - 1}{I(b_\theta - \log\eta) + a_\theta + N - 1}$.

(8) Sample $\eta^{(s+1)}$ from $p(\eta \mid \text{rest}) = \mathsf{Beta}(\eta \mid \theta + 1, I)$.

A.2 Stochastic gradient Hamiltonian Monte Carlo

The following are the steps required to draw samples from $p(\boldsymbol{\theta} \mid \text{D})$ using a SGHMC algorithm:

(1) Draw $\tilde{\text{D}}$ uniformly at random from D.

(2) Re-sample the momentum $\boldsymbol{r}^{(s)}$ from $\mathsf{Normal}(\mathbf{0}, \mathbf{M})$.
(3) Set $(\boldsymbol{\theta}_0, \boldsymbol{r}_0) = (\boldsymbol{\theta}^{(s)}, \boldsymbol{r}^{(s)})$.
(4) Simulate Hamiltonian dynamics:
  (i) $\boldsymbol{r}_0 \leftarrow \boldsymbol{r}_0 - \frac{\epsilon}{2}\nabla\tilde{U}(\boldsymbol{\theta}_0)$.
  (ii) For $i = 1, \ldots, L$ do: $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_{i-1} + \epsilon\mathbf{M}^{-1}\boldsymbol{r}_{i-1}$ and $\boldsymbol{r}_i \leftarrow \boldsymbol{r}_{i-1} - \epsilon\nabla\tilde{U}(\boldsymbol{\theta}_i)$.
  (iii) $\boldsymbol{r}_L \leftarrow \boldsymbol{r}_L - \frac{\epsilon}{2}\nabla\tilde{U}(\boldsymbol{\theta}_L)$.
(5) Set $(\boldsymbol{\theta}^*, \boldsymbol{r}^*) = (\boldsymbol{\theta}_L, \boldsymbol{r}_L)$.
(6) Compute the acceptance probability

$$a = \exp\Big(H(\boldsymbol{\theta}^*, \boldsymbol{r}^*) - H(\boldsymbol{\theta}^{(s)}, \boldsymbol{r}^{(s)})\Big),$$

where $H(\boldsymbol{\theta}, \boldsymbol{r}) = \tilde{U}(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{r}^{\mathsf{T}}\mathbf{M}\boldsymbol{r}$ is the Hamiltonian function.
(7) Let

$$\boldsymbol{\theta}^{(s+1)} = \begin{cases} \boldsymbol{\theta}^*, & \text{with probability } a; \\ \boldsymbol{\theta}^{(s)}, & \text{with probability } 1 - a. \end{cases}$$

Now, we take this algorithm to sample β and each $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_N$, where $N = \max\{\xi_i\}$ is the total number of latent individuals, as follows:

(a) If $\boldsymbol{\theta} = \beta$, then we have that

$$U(\beta) = -\sum_{i<i'}\big(y_{i,i'}\log\theta_{i,i'} + (1 - y_{i,i'})\log(1 - \theta_{i,i'})\big) - \frac{1}{\sqrt{2\pi\omega^2}}\exp\big(-\tfrac{1}{2}\beta^2\big)$$

and

$$\nabla U(\beta) = \sum_{i<i'}\mathrm{expit}(-(2y_{i,i'} - 1)\eta_{i,i'}) + \frac{\beta}{\omega^2},$$

where $\eta_{i,i'} = \beta - \|\boldsymbol{u}_{\xi_i} - \boldsymbol{u}_{\xi_i'}\|$ and $\theta_{i,i'} = \mathrm{expit}(\eta_{i,i'})$.
(b) If $\boldsymbol{\theta} = \boldsymbol{u}_n$, then we have that

$$U(\boldsymbol{u}_n) = -\sum_{i'\in R_i}\big(y_{i,i'}\log\theta_{i,i'} + (1 - y_{i,i'})\log(1 - \theta_{i,i'})\big) - (2\pi\sigma^2)^{-K/2}\exp\left(-\frac{1}{2\sigma^2}\,\boldsymbol{u}_n^T\boldsymbol{u}_n\right)$$

and

$$\nabla U(\boldsymbol{u}_n) = \left[\sum_{\mathrm{R}_i}\mathrm{expit}(-(2y_{i,i'} - 1)\eta_{i,i'})\,\frac{u_{n,k} - u_{\xi_{i'},k}}{\|\boldsymbol{u}_n - \boldsymbol{u}_{\xi_{i'},k}\|} + \frac{u_{n,k}}{\sigma^2}\right],$$

where $\mathrm{R}_i = \{i \in \{1, \ldots, I\} : \xi_i = n\}$.

Appendix 2. Distributions

Now, we present the form of some standard probability distributions:

- Multivariate Normal:
  A $d \times 1$ random vector $\boldsymbol{X} = (X_1 \ldots, X_d)$ has a multivariate Normal distribution with parameters $\boldsymbol{\mu}$ and Σ, denoted by $\boldsymbol{X} \mid \boldsymbol{\mu}, \Sigma \sim \mathsf{Normal}(\boldsymbol{\mu}, \Sigma)$, if its density function is

$$p(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma) = (2\pi)^{-d/2}\,|\Sigma|^{-1/2}\exp\Big\{-\tfrac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\Big\}.$$

- Gamma:
  A random variable X has a Gamma distribution with parameters $\alpha, \beta > 0$, denoted by $X \mid \alpha, \beta \sim \mathsf{Gamma}(\alpha, \beta)$, if its density function is

$$p(x \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)}\,x^{\alpha-1}\exp\{-\beta x\}, \quad x > 0.$$

- Inverse Gamma:
  A random variable X has an Inverse Gamma distribution with parameters $\alpha, \beta > 0$, denoted by $X \mid \alpha, \beta \sim \mathsf{Inverse\text{-}Gamma}(\alpha, \beta)$, if its density function is

$$p(x \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{-(\alpha+1)} \exp\{-\beta/x\}, \quad x > 0\,.$$

- Beta:
  A random variable X has a Beta distribution with parameters $\alpha, \beta > 0$, denoted by $X \mid \alpha, \beta \sim \mathsf{Beta}(\alpha, \beta)$, if its density function is

$$p(x \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\,\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}, \quad 0 < x < 1\,.$$

- Dirichlet:
  A $K \times 1$ random vector $\boldsymbol{X} = (X_1, \ldots, X_K)$ has a dirichlet distribution with parameter vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$, where each $\alpha_k > 0$, denoted by $\boldsymbol{X} \mid \boldsymbol{\alpha} \sim \mathsf{Dirichlet}(\boldsymbol{\alpha})$, if its density function is

$$p(x \mid \boldsymbol{\alpha}) = \begin{cases} \dfrac{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} x_k^{\alpha_k - 1}, & \text{if } \sum_{k=1}^{K} x_k = 1; \\ 0, & \text{otherwise.} \end{cases}$$

- Categorical:
  A $K \times 1$ random vector $\boldsymbol{X} = (X_1, \ldots, X_K)$ has a categorical distribution with parameter vector $\boldsymbol{p} = (p_1, \ldots, p_K)$, where $\sum_{k=1}^{K} p_k = 1$, denoted by $\boldsymbol{X} \mid \boldsymbol{p} \sim \mathsf{Categorical}(\boldsymbol{p})$, if its probability mass function is

$$p(x \mid \boldsymbol{p}) = \begin{cases} \prod_{k=1}^{K} p_k^{\mathbb{I}[x=k]}, & \text{if } \sum_{k=1}^{K} x_k = 1; \\ 0, & \text{otherwise.} \end{cases}$$

Appendix 3. Dirichlet process

A random distribution function F is generated from a Dirichlet Process (DP) with parameters $\alpha > 0$ and G a distribution function on $\mathbb{R}$, denoted by $F \sim \mathsf{DP}(\alpha, G)$, if for any finite measurable partition $B_1, \ldots, B_k$ of $\mathbb{R}$,

$$(F(B_1), \ldots, F(B_k)) \sim \mathsf{Dirichlet}(\alpha G(B_1), \ldots, \alpha G(B_k))\,.$$

G plays the role of the center of the DP (also referred to as base probability measure, or base distribution), where as α can be viewed as a precision parameter (the larger α is, the closer we expect a realization F from the process to be to G). See Ferguson [15] for the role of G on more technical properties of the DP.

Alternatively, the constructive definition of the DP [35] states that $F \sim \mathsf{DP}(\alpha, G)$ if F is (almost surely) of the form

$$F(\cdot) = \sum_{k=1}^{\infty} \omega_k \delta_{\vartheta_k}(\cdot)$$

where δ_{ϑ} denotes a point mass at ϑ (degenerate distribution putting probability one on ϑ), $\vartheta_k \overset{\text{iid}}{\sim} G$, $\omega_k = z_k \prod_{\ell=1}^{k-1}(1 - z_\ell)$, $z_k \overset{\text{iid}}{\sim} \mathsf{Beta}(1, \alpha)$, for $k = 1, 2, \ldots$. Hence, the DP generates distributions that can be represented as countable mixtures of point masses (the locations ϑ_k arise i.i.d. from the base distribution G), whose weights ω_k arise through a stick-breaking construction (it can be shown that $\sum_{k=1}^{\infty} \omega_k = 1$ almost surely). Based on its constructive definition, it is evident that the DP generates (almost surely) discrete distributions on $\mathbb{R}$.

Appendix 4. Notation

The cardinality of a set A is denoted by $|A|$. If P is a logical proposition, then $\mathsf{I}[\mathrm{P}] = 1$ if P is true, and $\mathsf{I}[\mathrm{P}] = 0$ if P is false. The Gamma function is given by $\Gamma(x) = \int_0^\infty u^{x-1}\, e^{-u}\, \mathrm{d}u$. Matrices and vectors with entries consisting of subscripted variables are denoted by a boldfaced version of the letter for that variable. For example, $\boldsymbol{x} = (x_1, \ldots, x_n)$ denotes an $n \times 1$ column vector with entries $x_1, \ldots, x_n$. We use $\mathbf{0}$ and $\mathbf{1}$ to denote the column vector with all entries equal to 0 and 1, respectively, and $\mathbf{I}$ denote the identity matrix. A subindex in this context refers to the corresponding dimension; for instance, $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. The transpose of a vector $\boldsymbol{x}$ is denoted by $\boldsymbol{x}^\mathsf{T}$; analogously for matrices. Moreover, if $\mathbf{X}$ is a square matrix, we use $\mathrm{tr}(\mathbf{X})$ to denote its trace and $\mathbf{X}^{-1}$ to denote its inverse. The norm of $\boldsymbol{x}$, given by $\sqrt{\boldsymbol{x}^\mathsf{T}\boldsymbol{x}}$, is denoted by $\|\boldsymbol{x}\|$.