A Bayesian Approach for De-duplication in the Presence of Relational Data

Juan Sosa

Universidad Nacional de Colombia

Abel Rodríguez

University of Washington

Abstract. In this paper, we study the impact of combining profile and network data in a de-duplication setting. We also assess the influence of a range of prior distributions on the linkage structure. Furthermore, we explore 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.

Keywords. Allelic Partitions; 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. This process can be treated as a clustering problem, in which each latent entity is associated with one or more noisy database records. From a model-based perspective, popular choices for clustering include finite mixture models and Dirichlet/Pitman-Yor process mixture models (Müller and Rodriguez, 2013, Casella et al., 2014, Miller and Harrison, 2016). Although these alternatives have proven to be successful in all sorts of applications, they are not realistic for de-duplication problems.

Some approaches for de-duplication have been considered during the past few years. Domingos (2004) treat 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 (2013) and Sadinle (2014) look for duplicate records partitioning the data file into groups of coreferent records. They present an approach that targets this partition of the file as the parameter of interest, thereby ensuring transitive decisions. The work of Steorts (2015) and Steorts et al. (2016) also permit de-duplication while handling multiple files simultaneously. Other recent contributions in this area that have proven to be very useful are Enamorado and Steorts (2020), Tancredi et al. (2020), Marchant et al. (2021), and Aleshin-Guendel and Sadinle (2021).

Unlike models exhibiting infinitely exchangeable clustering features, models specifically conceived for entity resolution (ER) 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 (Miller et al., 2015, Betancourt et al., 2016, Betancourt et al., 2020b, Betancourt et al., 2020a).

On the other hand, findings in Sosa and Rodriguez (2018) show that network data can substantially improve merging online social networks (OSNs). Hence, it make sense that network

E-mail: jcsosam@unal.edu.co
E-mail: abelrod@uw.edu

data can be also useful in other ER tasks such as de-duplication. This might be useful, for example, in identifying covert users in a social network, which might have multiple profiles linking to the same groups of individuals. Thus, our goal in this manuscript is three-fold. First, we extend the model in Sosa and Rodriguez (2018) from OSNs matching to handle de-duplication tasks. 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 (Chen et al., 2014) as a faster way to obtain samples from the posterior distribution for network parameters.

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 in detail the concept of microclustering. Section 4 presents a number of prior distributions on the linkage structure. Section 5 compares the performance of the resulting procedures using the RLdata500 data, a popular dataset in the record linkage literature. Section 6 explores the robustness of the results to the prior specification and the structural features of the network information. Section 7 presents a faster way to draw samples for the network parameters based on stochastic gradient methods. Lastly, we discuss our findings and directions for future work in Section 8.

2. A de-duplication model incorporating relational data

We rely on the formulation provided in Steorts et al. (2016) and Sosa and Rodriguez (2018) for $J = 1$ file. Specifically, we have a single file with I records, each containing L fields in common, for which both profile data $\mathbf{P} = [p_{i,\ell}]$ and network data $\mathbf{Y} = [y_{i,i'}]$ are available in order to uncover multiple records corresponding to the same latent identity.

2.1. Model formulation

We model both sources of information independently given the linkage structure $\boldsymbol{\xi} = (\xi_1, \dots, \xi_I)$, which defines a partition $\mathcal{C}_{\boldsymbol{\xi}}$ on $\{1, \dots, I\}$. Entries in $\boldsymbol{\xi}$ are labeled consecutively from 1 to N. See Section 3 in Sosa and Rodriguez (2018) for more details about notation and the nature of the problem.

Accordingly, we model the relational data through a latent distance model (Hoff et al., 2002) of the form

$$y_{i,i'} \mid \beta, \boldsymbol{u}_{\xi_i}, \boldsymbol{u}_{\xi_{i'}} \overset{\text{ind}}{\sim} \mathsf{Ber}\left(\operatorname{expit}\left(\beta - \|\boldsymbol{u}_{\xi_i} - \boldsymbol{u}_{\xi_{i'}}\|\right)\right), \tag{1}$$

where each $\boldsymbol{u}_n$ is embedded in a K-dimensional social space. Then, the attribute data are modeled according to the status of field-specific distortion indicators $w_{i,\ell}$ through

$$p_{i,\ell} \mid \pi_{\xi_i,\ell}, w_{i,\ell}, \boldsymbol{\vartheta}_\ell \overset{\text{ind}}{\sim} \begin{cases} \delta_{\pi_{\xi_i,\ell}}, & w_{i,\ell} = 0; \\ \mathsf{Cat}(\boldsymbol{\vartheta}_\ell), & w_{i,\ell} = 1, \end{cases} \tag{2}$$

where $\boldsymbol{\vartheta}_\ell$ is an M_ℓ-dimensional vector of multinomial probabilities. The rest of the model, but the prior specification on $\boldsymbol{\xi}$, is given exactly as in Sosa and Rodriguez (2018, Sec. 3 and 4). Lastly, we devote Section 4 in this document to discuss several prior formulations for $\boldsymbol{\xi}$.

2.2. Hyperparameter elicitation

Following the same line of thought given in Krivitsky and Handcock (2008), we let the 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, for the profile hyperparameters we follow very closely Steorts (2015) and set $\alpha_{\ell,m} = 1$, $a_\ell = a = 1$, and $b_\ell = b = 99$.

2.3. Computation

Computation for this model can still be achieved via Markov chain Monte Carlo (MCMC) algorithms (Gamerman and Lopes, 2006). Hence, the full set of parameters in this case is

$$\boldsymbol{\Upsilon} = (\boldsymbol{\xi}, \boldsymbol{\phi}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_N, \beta, \sigma, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_N, \boldsymbol{\vartheta}_1, \ldots, \boldsymbol{\vartheta}_L, \boldsymbol{w}, \boldsymbol{\psi})$$

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. The Appendix 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 $p(\boldsymbol{\Upsilon} \mid \text{data})$ and then getting a point estimate of the overall linkage structure.

3. Microclustering

Finite mixture models and Dirichlet/Pitman-Yor process mixture models are widely used in many clustering applications (Miller and Harrison, 2016). These models generate cluster sizes that grow linearly with the number of records I, i.e., for all n, $\frac{1}{I}\sum_i \mathbb{I}\{\xi_i = n\} \xrightarrow{\text{a.s.}} \Pr[\xi_i = n]$ when $I \to \infty$. Such a property is unappealing to address de-duplication problems because we need to generate a large number clusters with a negligible number of records (mostly singletons and pairs).

In order to formulate more realistic models for de-duplication, Miller et al. (2015) 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 $M = \max\{|C_n| : C_n \in \mathcal{C}_{\boldsymbol{\xi}}\}$ 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 (Betancourt et al., 2016).

Miller et al. (2015) 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. (2016) 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. (2010) sacrifices exchangeability instead.

4. Prior specification on the linkage structure

4.1. Kolchin partition priors

The Kolchin partition priors (KPPs) are originally introduced in Betancourt et al. (2016) 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 $S_1, \ldots, S_N \mid N \overset{\text{iid}}{\sim} \mu$. Here 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 $\mathcal{C}_{\boldsymbol{\xi}} = \{C_1, \ldots, C_N\}$, 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 usually observed), it can be shown that the probability of any given partition is

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

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, and adopt them as a baseline in Section 5. We remit the reader to Miller et al. (2015), Betancourt et al. (2016), and Betancourt et al. (2020b) 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 $\eta \sim \mathsf{Gam}(a_\eta, b_\eta)$ and $\theta \sim \mathsf{Beta}(a_\theta, b_\theta)$ for fixed hyper-parameters $a_\eta, b_\eta, a_\theta, b_\theta$. When evaluating the performance of this prior, we follow the authors and set a and q in a way that $\mathbb{E}[N] = \sqrt{\mathbb{V}\mathrm{ar}[N]} = \frac{I}{2}$, $a_\eta = b_\eta = 1$, 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 $\mu \sim \mathsf{DP}(\alpha, \mu^0)$. Here, a and q are once again fixed hyperparameters, α is a fixed concentration parameter and μ^0 is a fixed base measure with $\sum_{m=1}^{\infty} \mu^0(m) = 1$ and $\mu^0(m) \geq 0$, for all m. 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.

4.2. Allelic partition priors

Here we consider a class of prior distributions on the cluster assignments $\boldsymbol{\xi}$ based on allelic partitions (Crane et al., 2016). Let $\mathcal{C}_{\boldsymbol{\xi}} = \{C_1, \ldots, C_N\}$ be the partition implicitly represented by $\boldsymbol{\xi}$, and 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 and then selecting uniformly among partitions for which that specific allelic partition holds. This simple reasoning allow us to write

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

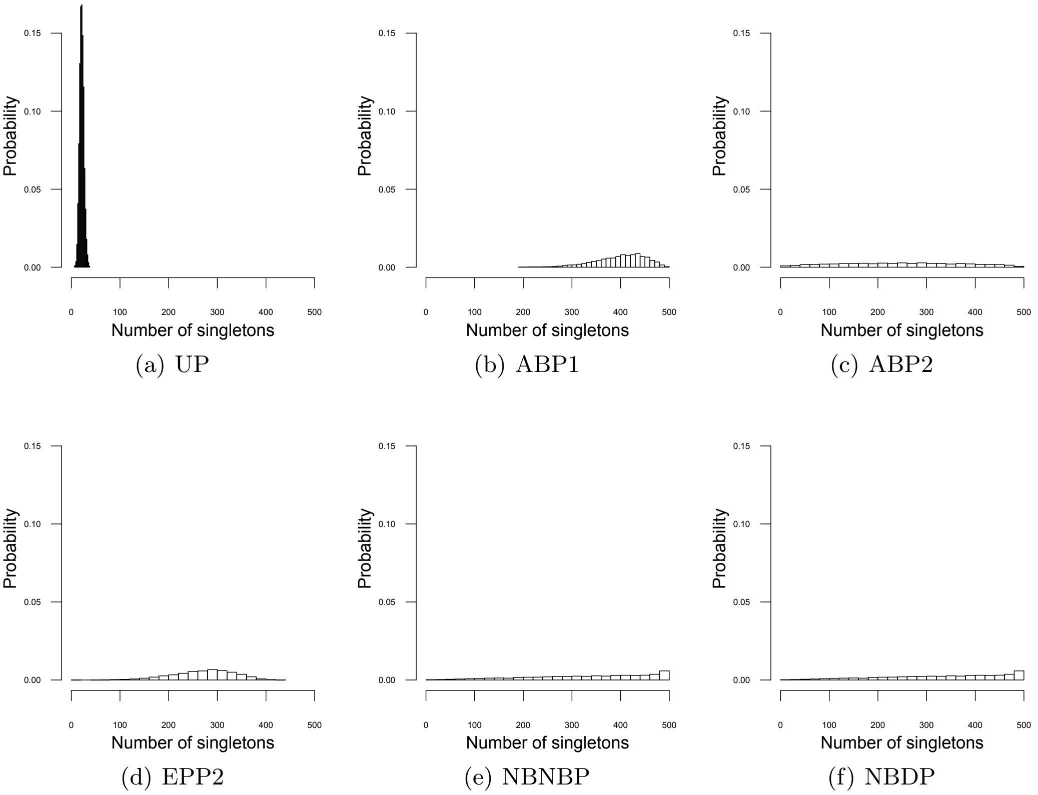
which fully determines an exchangeable partition probability function. Thus, we just need to place a distribution on $\boldsymbol{r}$ in order to complete the prior specification. We discuss below two instances that can be framed in the context of allelic partitions. We remit the reader to (Betancourt et al., 2020a) for details about computation and prior elicitation.

We consider the allelic binomial prior (ABP, Betancourt et al., 2020a), setting the maximum cluster size to $M = 2$, which leads to the allelic partition $\boldsymbol{r} = (I - 2r_2, r_2, 0, 0, \ldots, 0)$, and then letting $r_2 \sim \mathsf{Beta\text{-}Bin}(a_2, b_2)$. 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.

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, 2006). 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 $\theta \sim \mathsf{Gam}(a_\theta, b_\theta)$. The parameters a_θ and b_θ are carefully chosen in order to match the prior beliefs given in the ABP.

5. 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



(a) UP (b) ABP1 (c) ABP2

(d) EPP2 (e) NBNBP (f) NBDP

Figure 1. Prior distribution of the number of singleton clusters.

RLdata500 dataset from the RecordLinkage package (Borg and Sariyar, 2016) in R, which has been considered by many authors to test their methodologies, including Christen and Pudjijono (2009), Christen and Vatsalan (2013), Steorts et al. (2014), Steorts (2015), and Tancredi et al. (2020). 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.

We augment this dataset by generating social ties between records following the latent distance model (1), where $u_{n,k} \mid \sigma^2 \overset{\text{iid}}{\sim} \mathsf{N}(0, \sigma^2)$ and $\boldsymbol{\xi}$ corresponds to the true linkage structure in the dataset. We consider two scenarios (see Table 1), which allows us to study how structural features influence the de-duplication process.

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

Table 1: Features of the network data.

We fit our de-duplication model using just profile data as well as using both profile and

Prior	Recall	Precision	F_1	$\mathbb{E}[N \mid \text{data}]$	$\text{SD}[N \mid \text{data}]$
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

Table 2: Performance assessment and summary statistics for each prior distribution using just profile data and also using both profile and network data. Only categorical fields are considered.

network data with $K = 2$. We also implement each prior specification given in Section 4, along with an uniform prior (UP) as in Steorts et al. (2016). 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 aslo calibrated in a similar fashion. 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 A based on 100,000 samples obtained after a burn-in period of 500,000 iterations. In addition, the clustering methodology proposed by Lau and Green (2007) was used to obtain a point estimate of the posterior linkage structure.

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. There seems to be no difference between our prior 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; specially 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 our prior 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.

6. Sensitivity analysis

We fitted our de-duplication model making specific choices for several quantities. Specifically, we chose $\psi_\ell \overset{\text{iid}}{\sim} \mathsf{Beta}(a_\ell, b_\ell)$, with $a_\ell = a = 1$ and $b_\ell = b = 99$. Here we consider the effect of varying the values of a and b on the posterior linkage and the estimate of the population size. 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.

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 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.

7. 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 $p(\boldsymbol{\theta}, \boldsymbol{r}) \propto \exp\left\{-U(\boldsymbol{\theta}) - \frac{1}{2}\boldsymbol{r}^T\mathbf{M}\boldsymbol{r}\right\}$ by simulating from a Hamiltonian system, where $\mathbf{M}$ is a mass matrix usually defined as the identity matrix. If we simply discard the resulting $\boldsymbol{r}$ samples, the $\boldsymbol{\theta}$ samples have marginal distribution $p(\boldsymbol{\theta} \mid \mathrm{D})$. See Neal et al. (2011) for details.

Now, along the lines of Chen et al. (2014), 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:

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

where $|\cdot|$ denotes the cardinality of a set. 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

a	b	Recall	Precision	F_1	$\mathbb{E}\,[N \mid \text{data}]$	$\text{SD}\,[N \mid \text{data}]$
$a/(a+b) = 0.002$						
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

Table 3: Performance assessment and summary statistics for the ABP2 using both profile and network data (Scenario 1). Several values of a and b have been considered.

are provided in Appendix A.

We want to compare a random-walk (RW) and a SGHMC in terms of accuracy and computational cost using the data provided in Section 5. 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 the Appendix using both a RW and a SGHMC to sample from the conditional distribution of β and each $\boldsymbol{u}_n$. 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 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.

8. Discussion

We have proposed a novel approach for de-duplication that easily reconciles both profile and network data. We have also developed a new prior specification on the cluster assignments, the

Algorithm	Recall	Precision	F_1	$\mathbb{E}[N \mid \text{data}]$	$\text{SD}[N \mid \text{data}]$	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

Table 4: Performance assessment and summary statistics for the ABP2 using both profile and network data (Scenario 1). Time is given in seconds per 100 iterations using a standard laptop with 16GB of RAM and a 2.60GHz Intel Core i7 processor.

ABP, which is easy to implement, naturally satisfies the microclustering property, and also makes it straightforward to incorporate prior believes about the linkage structure. 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. 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 work opens several doors for future research. We could either add an extra hierarchy to model the size of the larger cluster M in a way that microclustering is preserved or consider a different joint distribution for the corresponding allelic partition $\boldsymbol{r}$. Lastly, it also may be worth considering other fast approximation techniques in the flavor of variational approximations (Saul et al., 1996, Jordan et al., 1998, Beal, 2003, Broderick and Steorts, 2014). This would allow us to consider bigger datasets with even millions of records.

References

Aleshin-Guendel, S. and Sadinle, M. (2021). Multifile partitioning for record linkage and duplicate detection.

Beal, M. J. (2003). Variational algorithms for approximate Bayesian inference. University of London.

Betancourt, B., Sosa, J., and Rodríguez, A. (2020a). A prior for record linkage based on allelic partitions. arXiv preprint arXiv:2008.10118.

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

Betancourt, B., Zanella, G., and Steorts, R. C. (2020b). Random partition models for microclustering tasks. Journal of the American Statistical Association, pages 1–13.

Borg, A. and Sariyar, M. (2016). RecordLinkage: Record Linkage in R. R package version 0.4-10.

Broderick, T. and Steorts, R. C. (2014). Variational bayes for merging noisy databases. arXiv preprint arXiv:1410.4792.

Casella, G., Moreno, E., Girón, F. J., et al. (2014). Cluster analysis, model selection, and prior distributions on models. Bayesian Analysis, 9(3):613–658.

Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient hamiltonian monte carlo. In International Conference on Machine Learning, pages 1683–1691.

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

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

Crane, H. et al. (2016). The ubiquitous ewens sampling formula. Statistical Science, 31(1):1–19.

Domingos, P. (2004). Multi-relational record linkage. In In Proceedings of the KDD-2004 Workshop on Multi-Relational Data Mining. Citeseer.

Enamorado, T. and Steorts, R. C. (2020). Probabilistic blocking and distributed bayesian entity resolution. In International Conference on Privacy in Statistical Databases, pages 224–239. Springer.

Gamerman, D. and Lopes, H. F. (2006). Markov chain Monte Carlo: stochastic simulation for Bayesian inference. CRC Press.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. Journal of the american Statistical association, 97(460):1090–1098.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1998). An introduction to variational methods for graphical models. Springer.

Krivitsky, P. N. and Handcock, M. S. (2008). Fitting position latent cluster models for social networks with latentnet. Journal of statistical software, 24.

Lau, J. W. and Green, P. J. (2007). Bayesian model-based clustering procedures. Journal of Computational and Graphical Statistics, 16(3):526–558.

Marchant, N. G., Kaplan, A., Elazar, D. N., Rubinstein, B. I., and Steorts, R. C. (2021). d-blink: Distributed end-to-end bayesian entity resolution. Journal of Computational and Graphical Statistics, pages 1–16.

McCullagh, P. and Yang, J. (2006). Stochastic classification models. In Proceedings of the International Congress of Mathematicians Madrid, August 22–30, 2006, pages 669–686.

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

Miller, J. W. and Harrison, M. T. (2016). Mixture models with a prior on the number of components. Journal of the American Statistical Association.

Müller, P. and Rodriguez, A. (2013). Nonparametric bayesian inference. Institute of Mathematical Statistics.

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, 2(11).

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

Sadinle, M. and Fienberg, S. E. (2013). A generalized fellegi–sunter framework for multiple record linkage with application to homicide record systems. Journal of the American Statistical Association, 108(502):385–397.

Saul, L. K., Jaakkola, T., and Jordan, M. I. (1996). Mean field theory for sigmoid belief networks. Journal of artificial intelligence research, 4(1):61–76.

Sosa, J. and Rodriguez, A. (2018). A record linkage model incorporating relational data. arXiv preprint arXiv:1808.04511.

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

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

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

Tancredi, A., Steorts, R., and Liseo, B. (2020). A unified framework for de-duplication and population size estimation (with discussion). Bayesian Analysis, 15(2):633–682.

Wallach, H., Jensen, S., Dicker, L., and Heller, K. (2010). An alternative prior process for nonparametric bayesian clustering. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 892–899.

A. Computation for ER Model

A.1. Markov chain Monte Carlo

Consider the MCMC algorithm presented in Sosa and Rodriguez (2018). No further steps are required when the UP is considered as the prior distribution for $\boldsymbol{\xi}$. However, if the ABP2 is used, note that

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

in step 1, and to complete the sampler, we need to add the following step to the algorithm:

9. 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:

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

$$p(\theta \mid \text{rest}) = \epsilon\, \mathsf{Gam}(\theta \mid a_\theta + N, b_\theta - \log\eta) + (1-\epsilon)\mathsf{Gam}(\theta \mid a_\theta + N - 1, b_\theta - \log\eta)$$

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

10. 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 $p(\boldsymbol{\theta} \mid \mathrm{D})$ using a SGHMC algorithm:

1. Draw $\tilde{\mathrm{D}}$ uniformly at random from D.
2. Re-sample the momentum $\boldsymbol{r}^{(s)}$ from $\mathsf{N}(\mathbf{0}, \mathbf{M})$.
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 $i = 1, \dots, L$ 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 $H(\boldsymbol{\theta}, \boldsymbol{r}) = \tilde{U}(\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{r}^T\mathbf{M}\boldsymbol{r}$ 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 $\boldsymbol{u}_n$, $n = 1, \dots, N$, being $N = \max\{\boldsymbol{\xi}\}$ the total number of latent individuals, as follows:

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

$$\begin{aligned} U(\beta) &= -\sum_{i<i'} [y_{i,i'} \log\theta_{i,i'} + (1 - y_{i,i'})\log(1 - \theta_{i,i'})] - \tfrac{1}{\sqrt{2\pi\omega^2}} \exp\{-\tfrac{1}{2}\,\beta^2\}, \\ \nabla U(\beta) &= \sum_{i<i'} [\text{expit}\{-(2y_{i,i'} - 1)\eta_{i,i'}\}] + \tfrac{\beta}{\omega^2}, \end{aligned}$$

where $\eta_{i,i'} = \beta - \|\boldsymbol{u}_{\xi_i} - \boldsymbol{u}_{\xi_i'}\|$ and $\theta_{i,i'} = \text{expit}\{\eta_{i,i'}\}$.

(b) If $\boldsymbol{\theta} = \boldsymbol{u}_n$, then we have that:

$$\begin{aligned} U(\boldsymbol{u}_n) &= -\sum_{i' \in R_i} [y_{i,i'} \log\theta_{i,i'} + (1 - y_{i,i'})\log(1 - \theta_{i,i'})] \\ &\qquad - (2\pi\sigma^2)^{-K/2} \exp\{-\tfrac{1}{2\sigma^2}\,\boldsymbol{u}_n^T\boldsymbol{u}_n\}, \\ \nabla U(\boldsymbol{u}_n) &= \left[\sum_{R_i} \text{expit}\,\{-(2y_{i,i'} - 1)\eta_{i,i'}\}\, \frac{u_{n,k} - u_{\xi_{i'},k}}{\|\boldsymbol{u}_n - \boldsymbol{u}_{\xi_{i'},k}\|} + \tfrac{u_{n,k}}{\sigma^2}\right], \end{aligned}$$

where $R_i = \{i \in [I] : \xi_i = n\}$