

Tancredi et al. (2020): documentation and experiments

Nianqiao Ju, Niloy Biswas, Pierre E. Jacob,
Gonzalo Mena, John O’Leary, Emilia Pompe

Spring 2020

Abstract

This is a draft of the documentation for the code available at https://github.com/EmiliaPompe/discussion_unified_framework. Here we briefly describe the model of Tancredi et al. [2018], followed by our own implementation choices and some numerical results.

Important note: for the moment, our implementation focuses on the update of η and N , keeping the other components $(\beta', \beta_0, \theta)$ fixed. See the end of this document for more details.

1 Context, model and target distribution

We consider a de-duplication task on a dataset v , as in Tancredi et al. [2018]. While v could potentially contain several distinct files, for simplicity we assume that we have exactly one file with records $j \in \{1, \dots, n\}$ and fields $\ell \in \{1, \dots, p\}$, taking values $v_{j,\ell} \in \{1, \dots, M_\ell\}$. We think of v as a corrupted version of an unobserved dataset with N distinct records, where N is unknown. The model relating v to this unobserved dataset involves a few components:

- A mechanism to link the n observed records to the N unobserved records, whereby more than one observed record might relate to the same unobserved record. Thus we are talking about a model of partitions of $\{1, \dots, n\}$.
- A mechanism for the corruption process linking the observed and underlying data.
- A model component related to N , the latent population size.

We write $p(\eta, N, \beta', \beta_0, \theta)$ for the target distribution, leaving its dependence on the observations v implicit. Here η describes the partition associating observed records with unique underlying entities; in particular, η is a n -vector with $\eta_j \in \{1, \dots, n\}$ for $j \in \{1, \dots, n\}$. For the other variables $N \in \mathbb{N}$ denotes the true population size, β', β_0 are associated with the corruption mechanism, and θ

describes the frequency of each field value among the unobserved records. Thus for each $\ell \in \{1, \dots, p\}$, θ_ℓ is a vector of probabilities on the set $\{1, \dots, M_\ell\}$.

Although we can describe any partition of $\{1, \dots, n\}$ with an η as described above, it will be convenient to have an alternative description of the partition for the computations below. In particular, we note that there is a bijection relating each possible η with a pair (Z, U') , where Z contains $k(Z)$ blocks of indices and U' is a list of $k(Z)$ labels. Observations i and j belong to the same block in Z exactly when $\eta_i = \eta_j$, and this common η value corresponds to the label of this block in U' . For example, if $\eta = (3, 1, 2, 5, 3)$ then $Z = (15|2|3|4)$, $U' = (3, 1, 2, 5)$, and $k(Z) = 4$. We write $U = (u'_z)$, so that block $z \in Z$ has label u'_z .

We take the target distribution to be

$$p(\eta, N, \beta', \beta_0, \theta) = \left(\prod_{z \in Z} p(v_z | Z, U', N, \beta', \theta) \right) p(U' | Z, N) p(Z | N) p(N) p(\beta' | \beta_0) p(\beta_0) p(\theta). \quad (1)$$

We describe each part of this equation in turn, below. Here the right hand side involves (Z, U') instead of η , but this is unproblematic due to the bijection described above. The term v_z refers to the set of observations (v_i) for $i \in z$.

First, the prior $p(\theta)$ follows the Dirichlet distribution. The prior $p(\beta_0)$ specifies that the components $\beta_{0,\ell}$ are $\mathcal{N}(m_0, s_0^2)$ (independently for all ℓ), with $m_0 = \text{logit}(0.01)$ and $s_0^2 = 0.1$. The conditional prior $p(\beta' | \beta_0)$ is that $\beta'_{j,\ell} \sim \mathcal{N}(\beta_{0,\ell}, s^2)$ for $j \in \{1, \dots, n\}$. We set $s^2 = 0.5$, as in [Tancredi et al. \[2018\]](#). The prior on population size is $p(N) \propto N^{-g}$ for some g , following the authors we set $g = 1.02$.

Given N , we assume (Z, U') has prior

$$p(Z, U' | N) = p(U' | Z, N) \times p(Z | N) = \frac{1}{n_k} \frac{N_k}{N^n}. \quad (2)$$

Here we write $a_b := a!/(a-b)!$ for any pair of integers $a \geq b$, and $k = k(Z)$, the number of blocks in Z . We note that the conditional distribution of Z given N is supported on partitions with at most N blocks.

We now consider the term $p(v_z | Z, U', N, \beta', \theta)$, which can be interpreted as the likelihood associated with block z of partition Z . We first assume that this factorizes into a product of p terms corresponding to each data field, so that

$$p(v_z | Z, U', N, \beta', \theta) = \prod_{\ell=1}^p p(v_{z,\ell} | Z, U', N, \beta', \theta). \quad (3)$$

In the following we will identify the label u'_z with one of the N latent records in \tilde{v} , where each $\tilde{v}_{u'_z}$ is a p -dimensional vector of data items. Note that this involves a minor abuse of notation, since u'_z refers to one of the N underlying records while the value of its label might not fall in $\{1, \dots, N\}$ if $N < n$. Nevertheless we can treat $\tilde{v}_{u'_z}$ as the component of \tilde{v} associated with block z , as long as there are fewer than N unique values among $(\eta_j)_{j \in 1:n}$.

We consider a particular ℓ , noting that the same calculations will hold for any $\ell \in \{1, \dots, p\}$. For any observation $j \in \{1, \dots, n\}$ and data item $\ell \in \{1, \dots, p\}$, we also define $\alpha'_{j,\ell} := \text{expit}(\beta'_{j,\ell}) = \exp(\beta'_{j,\ell}) / (1 + \exp(\beta'_{j,\ell}))$, recalling that β' is a vector parameterizing the data corruption mechanism.

By introducing $\tilde{v}_{u'_z,\ell}$ and simplifying the notation by removing unnecessary objects in the conditioning, we can write

$$p(v_{z,\ell}|Z, U', N, \beta', \theta) = \sum_{\tilde{v}_{u'_z,\ell}} p(v_{z,\ell}|\tilde{v}_{u'_z,\ell}, \beta'_{u'_z,\ell}, \theta) p(\tilde{v}_{u'_z,\ell}|\theta), \quad (4)$$

the sum running over the M_ℓ possible values of $\tilde{v}_{u'_z,\ell}$. We have $p(\tilde{v}_{u'_z,\ell}|\theta) = \theta_{\ell,\tilde{v}_{u'_z,\ell}}$ under the Categorical distribution. The term $p(v_{z,\ell}|\tilde{v}_{u'_z,\ell}, \beta'_{u'_z,\ell}, \theta)$ describes the corruption mechanism associated with the underlying record labelled u'_z and ℓ -th entry. If z contains just one index, say $j \in \{1, \dots, n\}$, we have

$$p(v_{j,\ell}|\tilde{v}_{u'_z,\ell}, \beta'_{u'_z,\ell}, \theta) = (1 - \alpha'_{u'_z,\ell}) \mathbb{1}(\tilde{v}_{u'_z,\ell} = v_{j,\ell}) + \alpha'_{u'_z,\ell} \theta_{\ell v_{j,\ell}},$$

thus multiplying by $\theta_{\ell,\tilde{v}_{u'_z,\ell}}$ and summing over $\tilde{v}_{u'_z,\ell}$ in $\{1, \dots, M_\ell\}$, and recalling that the sum of the entries of θ_ℓ is one, we obtain

$$(1 - \alpha'_{u'_z,\ell}) \times \theta_{\ell,v_{j,\ell}} + \alpha'_{u'_z,\ell} \theta_{\ell,v_{j,\ell}} = \theta_{\ell,v_{j,\ell}}.$$

Thus if $z = \{j\}$ we have

$$p(v_{z,\ell}|Z, U', N, \beta', \theta) = \theta_{\ell,v_{j,\ell}}. \quad (5)$$

Next, consider the case where block z contains two indices, $z = \{j_1, j_2\}$. We can differentiate between two cases: $v_{j_1,\ell} = v_{j_2,\ell}$ and $v_{j_1,\ell} \neq v_{j_2,\ell}$, and introduce the latent variable $\tilde{v}_{u'_z,\ell}$ as previously. In the former case, for any $\tilde{v}_{u'_z,\ell}$ among M_ℓ possible values,

$$\begin{aligned} & p(v_{z,\ell}|\tilde{v}_{u'_z,\ell}, Z, U', N, \beta', \theta) \\ &= p(v_{j_1,\ell}|\tilde{v}_{u'_z,\ell}, \beta'_{u'_z,\ell}, \theta) \times p(v_{j_2,\ell}|\tilde{v}_{u'_z,\ell}, \beta'_{u'_z,\ell}, \theta) \\ &= (1 - \alpha'^2_{u'_z,\ell}) \mathbb{1}(\tilde{v}_{u'_z,\ell} = v_{j_1,\ell}) \\ &+ 2(1 - \alpha'_{u'_z,\ell}) \alpha'_{u'_z,\ell} \theta_{\ell,v_{j_1,\ell}} \mathbb{1}(\tilde{v}_{u'_z,\ell} = v_{j_1,\ell}) \\ &+ \alpha'^2_{u'_z,\ell} \theta_{\ell,v_{j_1,\ell}}^2, \end{aligned}$$

thus multiplying by $\theta_{\ell,\tilde{v}_{u'_z,\ell}}$ and summing over $\tilde{v}_{u'_z,\ell}$ in $\{1, \dots, M_\ell\}$, we obtain

$$\begin{aligned} p(v_{z,\ell}|Z, U', N, \beta', \theta) &= (1 - \alpha'_{u'_z,\ell})^2 \theta_{\ell,v_{j_1,\ell}} \\ &+ (2\alpha'_{u'_z,\ell} - \alpha'^2_{u'_z,\ell}) \theta_{\ell,v_{j_1,\ell}}^2, \end{aligned}$$

while if $v_{j_1,\ell} \neq v_{j_2,\ell}$ we obtain

$$p(v_{z,\ell}|Z, U', N, \beta', \theta) = (2\alpha'_{u'_z,\ell} - \alpha'^2_{u'_z,\ell}) \theta_{\ell,v_{j_1,\ell}} \theta_{\ell,v_{j_2,\ell}}.$$

At the end of the day we obtain the same formula as in Section 4 of the article (bottom of page 11), with β, U replaced by β', U' . The recursion formula is as follows, for a block z that contains an index j^* and $s - 1$ others denoted by j_1, \dots, j_{s-1} ,

$$\begin{aligned} p(v_{z,\ell} | Z, U', N, \beta', \theta) &= \alpha'_{u'_{z,\ell}} \theta_{\ell, v_{j^*,\ell}} p(v_{z \setminus \{j^*\}, \ell} | Z, U', N, \beta', \theta) \\ &+ (1 - \alpha'_{u'_{z,\ell}}) \theta_{\ell, v_{j^*,\ell}} \prod_{h=1}^{s-1} \left[(1 - \alpha'_{u'_{z,\ell}}) \mathbb{1}(v_{j_h, \ell} = v_{j^*, \ell}) + \alpha'_{u'_{z,\ell}} \theta_{\ell, v_{j_h, \ell}} \right]. \end{aligned} \quad (6)$$

This is a key formula for the implementation of the Gibbs sampler.

Note that the derivatives of the likelihood $p(v_{z,\ell} | Z, U', N, \beta', \theta)$ could be also computed, for fixed Z, U', N , with respect to θ and with respect to β' , using the same recursions. This could be used to perform gradient-based MCMC updates for these components. Note also that for θ , which has a Dirichlet prior, the above recursions suggest that the full conditional in the posterior distribution is a mixture of Dirichlet distributions, with a number of components that grows rapidly in the number of clusters that contain more than one index.

2 Conditional distributions and Gibbs sampling

The algorithm operates on the variables $(\eta, \beta', N, \beta_0, \theta)$. The target distribution is provided in equation (1). We can draw samples from this distribution by updating the elements η, β', β_0, N and θ with a Gibbs sampler, iteratively targeting the conditional distributions of each of these variables given the others.

2.1 Draw η given the rest

To begin, we describe the update of the labels $\eta = (Z, U')$ given the other variables. We do so by updating each η_j in turn, given the other labels η_{-j} and the other state variables. For each possible label value $q \in \{1, \dots, n\}$ we compute the probability of $\{\eta_j = q\}$ under the full conditional distribution as

$$\mathbb{P}(\eta_j = q \mid \eta_{-j}, N, \beta', \beta_0, \theta, v) \propto \frac{p(v_{z_q} \mid \eta, \beta'_q, \theta)}{p(v_{z_q \setminus \{j\}} \mid \eta, \beta'_q, \theta)} \mathbb{P}(\eta_j = q \mid \eta_{-j}). \quad (7)$$

Here z_q refers to the partition block with label q , i.e. $u'_{z_q} = q$, so that the block z_q contains all indices j such that $\{\eta_j = q\}$. The leading factor in (7) can be interpreted as a likelihood ratio, with the right-most term coming from the prior. This is equation (5.5) in [Tancredi et al. \[2018\]](#). The likelihood ratio can be computed using (3) and (5) if q identifies a new block and otherwise following (6) given $p(v_{z_q \setminus \{j\}} \mid \eta, \beta'_q, \theta)$.

To compute the $\mathbb{P}(\eta_j = q \mid \eta_{-j})$ term in (7), we can use (2). Denote by k^-

the number of clusters in η_{-j} . Then,

$$\begin{aligned}\mathbb{P}(\eta_j = q \mid \eta_{-j}) &\propto \mathbb{P}(\eta_j = q, \eta_{-j}) \\ &= \begin{cases} \frac{1}{n_{k-}} \frac{N_{k-}}{N^n} & \text{if } q \text{ in an existing cluster,} \\ \frac{1}{n_{k-}+1} \frac{N_{k-}+1}{N^n} & \text{if } q \text{ in a new cluster,} \end{cases} \\ &\propto \begin{cases} 1 & \text{if } q \text{ in an existing cluster,} \\ \frac{N-k-}{n-k-} & \text{if } q \text{ in a new cluster.} \end{cases}\end{aligned}$$

Thus the implementation loops through $j = 1, \dots, n$. For each j , with probability ω (set to 10% as in the authors' code), an update of η_j is performed. The update starts by removing η_j from the existing partition. Then it computes the probability of $\eta_j = q$ for all q , corresponding to either non-empty clusters or empty clusters. It combines these probabilities with the prior as above, and samples η_j from the resulting Categorical distribution.

2.2 Draw N given the rest

The full conditional distribution of N is given by

$$p(N \mid \text{rest}) \propto \frac{N_k}{N^{n+g}} \cdot 1_{N \geq k}. \quad (8)$$

Recall that g is the hyper-parameter for the prior (set to 1.02 in the code) and k is the number of clusters in the partition induced by η . The term N_k stands for $N!/(N-k)!$ and we can sample from this conditional distribution by computing the probabilities for $N = 1, \dots, N_{\max}$ for some large truncation N_{\max} ; this is the approach implemented, as in the authors' code with $N_{\max} = 10,000$. Note that by pre-computing and storing log-factorials for all $N = 1, \dots, N_{\max}$, we can avoid these calculations at each iteration.

2.3 Draw θ given the rest

The full conditional distribution of the probability vector θ_ℓ is

$$p(\theta_\ell \mid -) \propto \left(\prod_{z \in Z} p(\nu_{z,\ell} \mid \eta, \beta'_{u'_z}, \theta) \right) p(\theta_\ell), \quad (9)$$

for each field $\ell = 1, \dots, p$. The distribution can be targeted using MH with a Dirichlet proposal distribution. We can sample proposals using a Dirichlet distribution with parameter $1 + \gamma_{\theta,\ell} \cdot \theta_\ell$, as in the authors' code. We used $\gamma_{\theta,\ell} = 10^4$ as the authors. This leads to a proposal centered nearly at the current value and with some small standard deviation.

Note that we have experimented with another type of update of θ that appears to improve the performance significantly. In this update, with some probability (say 50%) we perform an MH move as described above. Otherwise, we perform an MH move with an independent proposal. We can for instance

propose from a Dirichlet distribution with parameter given by the field frequencies, multiplied by n , multiplied by a scaling factor in $(0, 1)$, plus one. Note that this is similar to the conjugate posterior distribution under a uniform prior; the scaling factor makes the proposal distribution "flatter". We have found that 0.8 yielded good results in our experiments.

2.4 Draw $\beta = (\beta', \beta_0)$ given the rest

Sampling each $\beta_{0,\ell}$ for a given field $\ell \in \{1, \dots, p\}$ can be implemented using a direct Gibbs update following

$$p(\beta_{0,\ell} | \text{rest}) \sim \mathcal{N}(\mu_{0,\ell}, \sigma_{0,\ell}^2),$$

where

$$\sigma_{0,\ell}^2 := \left(\frac{1}{s_0^2} + \frac{n}{s^2} \right)^{-1} \quad \text{and} \quad \mu_{0,\ell} := \sigma_{0,\ell}^2 \left(\frac{m_0}{s_0^2} + \frac{\sum_{j=1}^n \beta'_{j,\ell}}{s^2} \right). \quad (10)$$

Recall that as in [Tancredi et al. \[2018\]](#), we can take $m_0 = \text{logit}(0.01)$, $s_0^2 = 0.1$ and $s^2 = 0.5$. We could also replace formulas given by (10) with analogous ones where the sum would go only over non-empty clusters; this would be a "partially collapsed Gibbs sampler".

Sampling each $\beta'_{u'_z,\ell}$ for a given cluster z and a given field $\ell \in \{1, \dots, p\}$ is implemented using a Metropolis-within-Gibbs update. We use a non-central parameterization for each $\beta'_{u'_z,\ell}$ update, such that we perform a Metropolis-within-Gibbs update on $\beta'_{u'_z,\ell} - \beta_{0,\ell}$ rather than directly on $\beta'_{u'_z,\ell}$. The full conditionals for each $\beta'_{u'_z,\ell} - \beta_{0,\ell}$ follow

$$p(\beta'_{u'_z,\ell} - \beta_{0,\ell} | \text{rest}) \propto \begin{cases} \mathcal{N}((\beta'_{u'_z,\ell} - \beta_{0,\ell}); 0, s^2) & \text{if } |z| \leq 1, \\ p(v_{z,\ell} | Z, U', N, \beta', \theta) \mathcal{N}((\beta'_{u'_z,\ell} - \beta_{0,\ell}); 0, s^2) & \text{otherwise,} \end{cases}$$

where $|z|$ is the size of the block z . Hence for empty clusters and clusters of size 1 we draw the new value from the prior. For clusters of size larger than 1 we use a random walk Metropolis–Hastings update, proposing the new point from a Normal distribution centred at the current value and standard deviation given by $\sqrt{0.5}/(\text{size of cluster } q)$. This is the proposal used by the authors in their code.

3 Coupling of each update

In order to enable convergence diagnostics as in [Biswas et al. \[2019\]](#) and unbiased estimators as in [Jacob et al. \[2020\]](#), one needs to generate pairs of chains such that they eventually meet exactly. This can be done by coupling each of the conditional updates of the Gibbs sampler described above. In this section we provide some details. The overarching goal is to obtain relatively small meeting times, defined as the first iteration at which all components of the two chains are identically equal.

3.1 η given the rest

To couple a pair of chains at the η sampling step, we must produce draws from a pair of distributions of the form in (7) that have a chance of coinciding exactly. Both are discrete distributions on $\{1, \dots, n\}$, so it seems natural to use a maximal coupling between two distributions on finite state spaces. For details, see [Johnson \[1998\]](#) and Section 5.4 of [Jacob et al. \[2020\]](#).

However, a naive implementation of this coupling of η draws did not result in practical meeting times in our experiments. Under the above, two chains may arrive at the same partition without meeting due to the use of different labels. For example $\eta = (3, 1, 2, 5, 3)$ and $\tilde{\eta} = (2, 5, 4, 1, 2)$ correspond to the same partition $Z = (15|2|3|4)$ but $\eta \neq \tilde{\eta}$. We solve this problem with a relabelling step, i.e. a transformation of the cluster labels to ensure that $\eta = \tilde{\eta}$ when the two vectors correspond to the same partition.

At each step, we relabel η as follows. Denote the clusters by z_1, \dots, z_n , where some of these might be empty. For each non-empty cluster z , we gather the indices $i_1, \dots, i_{k(z)}$ of the members of block z , and label z by the index of its smallest element, $\min_{m=1, \dots, k(z)} i_m$. Removing these assigned labels from the set $\{1, \dots, n\}$, we assign the remaining available labels to the empty clusters among z_1, \dots, z_n by increasing order.

3.2 N given the rest

Under truncation, the full conditional distribution of N given the rest (8) is a discrete distribution on $\{1, \dots, N_{\max}\}$. Thus we can again use the maximal coupling of two discrete distributions to draw coupled updates of N .

3.3 θ given the rest

For the θ sampling step, Metropolis-Hastings with Dirichlet proposals is used to target the full conditional in (9). We can couple a pair of chains at this step by:

1. jointly sampling proposals for both chains, using a maximal coupling of two Dirichlet distributions;
2. using the same uniform to accept or reject the proposals in the two chains.

When the proposals for the two chains are equal and are both accepted, then the θ vector in both chains are equal. When θ is high-dimensional, then maximal coupling of proposals may not be a good idea, as the overlap between the proposals might be low. We could instead think about using common random numbers and only attempting maximal coupling when chains are close.

When employing an independent proposal as suggested in Section 2, we find that the coupling of the θ -component occurs very rapidly.

3.4 β given the rest

To couple the update of β_0 , we can employ various couplings of Normal distributions. For reflection-maximal couplings, see Section 4.1 of [Jacob et al. \[2020\]](#) and [Bou-Rabee et al. \[2018\]](#).

For β' , recall that we work with differences $\beta'_{q,\ell} - \beta_{0,\ell}$ rather than with $\beta'_{q,\ell}$ itself. We consider three scenarios for a fixed index q .

1. Index q corresponds to clusters of size 0 (empty) or 1 for both chains.
2. Index q corresponds to clusters of size strictly greater than 1 for both chains.
3. Index q corresponds to clusters of size 0 or 1 for one of the chains but for the other chain q is associated with a cluster of size greater than 1.

In scenario 1 we can employ common random numbers to sample from the prior distribution $N(0, s^2)$, hence the coupling of this $\beta'_{q,\ell} - \beta_{0,\ell}$ follows immediately. In scenario 2 we can follow the procedure described in detail in [Jacob et al. \[2020\]](#), using maximal coupling of the proposal distributions (Algorithm 2 in [Jacob et al. \[2020\]](#)), and then using a common uniform variable at the acceptance/ rejection step.

In scenario 3 one chain is updated using a direct Gibbs update (from the prior) while the other evolves according to a Metropolis-within-Gibbs kernel. This makes this situation “interesting” from the point of view of coupling. An obvious strategy is to do “nothing special”, i.e. sample the two states independently. We may also propose a new state from a maximal coupling of the two distributions used for updating each of the chains described in Section 2: since the direct Gibbs update is a Normal, and the proposal in the MH step is also Normal, various couplings are possible, including maximal couplings. The proposed state is automatically accepted for the direct Gibbs update, while the other chain requires an acceptance/ rejection procedure; still the two components can possibly become identical.

4 Numerical experiments

In these numerical experiments we focus on the case where $(\beta', \beta_0, \theta)$ is kept fixed. Precisely, θ is set to the observed frequencies of the categories in the data, while β' is set to $\text{logit}(0.01)$ so that $\alpha = 0.01$ for all fields, and all clusters; the value of β_0 is then irrelevant.

The reason why we kept these fixed is that we did not manage to get exactly the same results as the authors when updating these components. There seems to be some discrepancy in the way we compute the MH acceptance ratio for θ , and there might be other discrepancies as well. On the other hand we hope that the coupling of the η and N updates is successful enough to make the point that such couplings are applicable and useful for such MCMC algorithms.

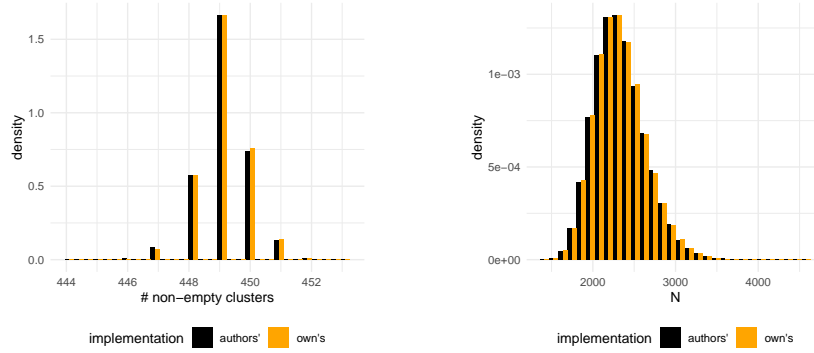


Figure 1: Results obtained with authors' code and our own implementation for $k(Z)$, the number of non-empty clusters (left), and for N (right).

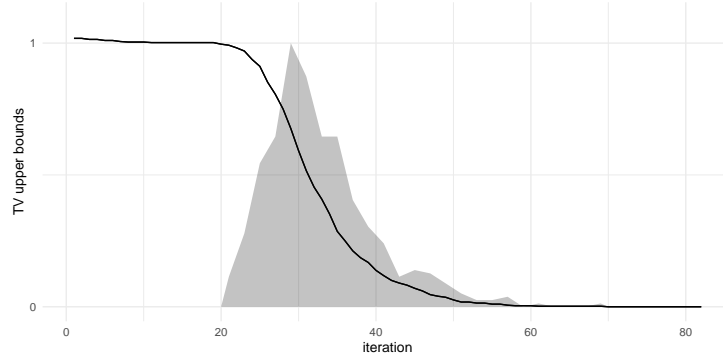


Figure 2: Upper bounds on $\|\pi_t - \pi\|_{\text{TV}}$ against iteration t , and histogram of meeting times minus lag.

When updating only N and η , we obtain very close agreement between the authors' code and our own implementation. See Figure 1, obtained with 5 independent runs for each implementation, and 20,000 iterations in total, discarding the first 100 as burn-in.

When running the two chains with a lag of 50 iterations, we obtain the estimated upper bounds on the total variation distance between the chain at step t and its limiting distribution shown in Figure 2. The grey area shows the histogram of $\tau - L$ where τ is the meeting time and L the lag equal to 50. The results suggest that the Gibbs sampler reaches stationarity in less than 60 iterations; again this is when $(\beta', \beta_0, \theta)$ kept fixed.

5 Discussion

The Gibbs sampler of Tancredi et al. [2018] can be coupled. We did so successfully for the update in η and N , and it required an additional “relabelling” step. The updates of the other components $(\beta', \beta_0, \theta)$ and their couplings are left for future work. With the modification of the θ update suggested above, the preliminary results are promising. However, at the moment we have observed some discrepancy between our implementation of these updates and the authors’ code.

References

- Andrea Tancredi, Rebecca Steorts, and Brunero Liseo. A Unified Framework for De-Duplication and Population Size Estimation. *Bayesian Analysis*, 2018. 1, 2, 4, 6, 10
- Niloy Biswas, Pierre E Jacob, and Paul Vanetti. Estimating convergence of Markov chains with L-lag couplings. In *Advances in Neural Information Processing Systems*, pages 7389–7399, 2019. 6
- Pierre E Jacob, John O’Leary, and Yves F Atchadé. Unbiased Markov chain Monte Carlo with couplings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) (with discussion) (to appear)*, 2020. 6, 7, 8
- Valen E Johnson. A coupling-regeneration scheme for diagnosing convergence in Markov chain Monte Carlo algorithms. *Journal of the American Statistical Association*, 93(441):238–248, 1998. 7
- Nawaf Bou-Rabee, Andreas Eberle, and Raphael Zimmer. Coupling and convergence for Hamiltonian Monte Carlo. *arXiv preprint arXiv:1805.00452*, 2018. 8