Random Partition Models for Microclustering Tasks

Brenda Betancourt *
University of Florida, Department of Statistics,
Giacomo Zanella
Bocconi University, Department of Decision Sciences, BIDSA and IGIER,
and
Rebecca C. Steorts
Duke University, Department of Statistical Science and Computer Science,

October 9, 2020

Abstract

In this supplementary material, we provide material that is not in the main body of our paper. Section 1 provides detailed proofs from our paper. More specifically, Section 1.1 provides the proof of Proposition 1. Section 1.2 provides the proof of Proposition 2 and Corollary 1. Section 1.3 provides the proof of Theorems 1, 2 and 3. Section 2 provides details regarding our proposed samplers that are used for posterior inference. In Section 3 we derive an importance sampler to simulate from ESC models and prove its validity. In Section 4, we provide the derivation of the likelihood that is used in our entity resolution (ER) task. Section 5 contains details about the MCMC algorithm and convergence checks for the applications. Finally, Section 6 includes additional results for the simulation study.

*BB and RCS gratefully acknowledge funding from NSF Big Data Privacy and NSF Career. GZ gratefully acknowledges support from the ERC through StG N-BNP" 306406 and from MIUR, through PRIN Project 2015SNS29B.

1 Proofs

1.1 Proof of Proposition 1

Proof of Proposition 1. We seek to compute the probability mass function (pmf) of the random partition $\Pi_n = \{C_1, \ldots, C_K\}$ obtained from Model $ESC_{[n]}(P_{\boldsymbol{\mu}})$. We denote this pmf by $\Pr(\Pi_n|E_n)$ to make explicit the conditioning on E_n in Step 1 of Model $ESC_{[n]}(P_{\boldsymbol{\mu}})$. Thus,

$$\Pr(\Pi_n|E_n) = \int \Pr(\Pi_n|\boldsymbol{\mu}, E_n)\Pr(d\boldsymbol{\mu}|E_n)\,.$$

By Bayes' theorem, we find that

$$\Pr(d\boldsymbol{\mu}|E_n) = \frac{P_{\boldsymbol{\mu}}(d\boldsymbol{\mu})\Pr(E_n|\boldsymbol{\mu})}{\Pr(E_n)}\,,$$

where given the construction in Step 1 of Model $ESC_{[n]}(P_{\boldsymbol{\mu}})$, we observe that

$$\Pr(E_n|\boldsymbol{\mu}) = \sum_{k=1}^{n} \sum_{(s_1,\ldots,s_k)\in\{1,n\}^k} \mathbb{I}\left(\sum_{j=1}^{k} s_j = n\right) \prod_{j=1}^{k} \mu_{s_j}$$

and $\Pr(E_n) = \int \Pr(E_n|\boldsymbol{\mu})P_{\boldsymbol{\mu}}(d\boldsymbol{\mu})$. Now, consider $\Pr(\Pi_n|\boldsymbol{\mu}, E_n)$. Summing over all possible cluster assignments $\mathbf{z} = (z_1, \ldots, z_n)$, we find that

$$\Pr(\Pi_n|\boldsymbol{\mu}, E_n) = \sum_{z_1,\ldots,z_n=1}^{K} \Pr(\Pi_n|\mathbf{z}, \boldsymbol{\mu}, E_n)\Pr(\mathbf{z}|\boldsymbol{\mu}, E_n).$$

The term $\Pr(\Pi_n|\mathbf{z}, \boldsymbol{\mu}, E_n)$ equals 1 for all $K!$ cluster assignments $\mathbf{z}$, leading to the partition Π_n and 0 otherwise. The term $\Pr(\mathbf{z}|\boldsymbol{\mu}, E_n)$ equals

$$\begin{aligned}\Pr(\mathbf{z}|\boldsymbol{\mu}, E_n) =& \Pr(\mathbf{z}|S_1, \ldots, S_K)\Pr(S_1, \ldots, S_K|\boldsymbol{\mu}, E_n) \\ =& \frac{\prod_{j=1}^{K} S_j!}{n!} \frac{\prod_{j=1}^{K} \mu_{S_j}}{\Pr(E_n|\boldsymbol{\mu})}\,,\end{aligned}$$

where $S_j = \sum_{i=1}^n \mathbb{I}(z_i = j)$ denote the size of the j-th cluster. It follows that

$$\Pr(\Pi_n|\boldsymbol{\mu}, E_n) = \frac{K! \prod_{j=1}^K S_j! \mu_{S_j}}{n! \Pr(E_n|\boldsymbol{\mu})} \tag{1.1}$$

and

$$\Pr(\Pi_n|E_n) = \int \frac{\Pr(\Pi_n|\boldsymbol{\mu}, E_n)\Pr(E_n|\boldsymbol{\mu})}{\Pr(E_n)} P_{\boldsymbol{\mu}}(d\boldsymbol{\mu}) = \frac{1}{n!\Pr(E_n)} \int K! \prod_{j=1}^K |S_j|! \mu_{S_j} P_{\boldsymbol{\mu}}(d\boldsymbol{\mu})\,. \tag{1.2}$$

The thesis follows from the definition of EPPF. $\square$

1.2 Proof of Proposition 2 and Corollary 1

Proof of Proposition 2 and Corollary 1. The expression for the conditional EPPF $p^{(n)}(\cdot; \mu)$ follows directly from Equation (1.1). The expression for the prediction rule follows from Bayes theorem and

$$\frac{\Pr(z_i, \mathbf{z}_{-i}|\boldsymbol{\mu}, E_n)}{\Pr(\mathbf{z}_{-i}|\boldsymbol{\mu}, E_n)} \propto k! \prod_{j=1}^k s_j! \mu_{s_j}\,.$$

$\square$

1.3 Proof of Theorems 1, 2 and 3

In this section, we prove Theorems 1, 2 and 3. The first essential ingredient for our proofs is the Renewal Theorem from the literature on Renewal processes.

Theorem 1.1 (Renewal Theorem)**.** *Assume $\mu_1 > 0$ and $\sum_{s=1} s\mu_s \leq \infty$. Then*

$$\Pr(E_n) \to \frac{1}{\sum_{s=1} s\mu_s} \text{ as } n \to \infty.$$

We refer to Barbu & Limnios (2009, Thm.2.6) for a proof of the Renewal Theorem. The second ingredient is the following technical Lemma that we prove below.

Lemma 1.1. *Let $X_1, X_2, \dots$ be a sequences of random variables and $E_1, E_2, \dots$ be a sequence of events, with E_n defined on the same probability space of X_n. If $X_n \xrightarrow{p} c$ as $n \to \infty$ for some $c \in \mathbb{R}$ and $\liminf_{n\to\infty} Pr(E_n) > 0$, then $X_n | E_n \xrightarrow{p} c$.*

Proof. Fix $\varepsilon > 0$ and define the event $A_n = \{|X_n - c| > \varepsilon\}$. Since $X_n \xrightarrow{p} c$ it follows that $\lim_{n\to\infty} Pr(A_n) = 0$. Thus

$$\limsup_{n\to\infty} Pr(A_n|E_n) = \limsup_{n\to\infty} \frac{Pr(A_n \cap E_n)}{Pr(E_n)} \leq \frac{\limsup_{n\to\infty} Pr(A_n)}{\liminf_{n\to\infty} Pr(E_n)} = 0\,,$$

where the last equality follows from $\lim_{n\to\infty} Pr(A_n) = 0$ and $\liminf_{n\to\infty} Pr(E_n) > 0$. It follows that, for any $\varepsilon > 0$, $\lim_{n\to\infty} Pr(|X_n - c| > \varepsilon | E_n) = 0$, meaning that $X_n | E_n \xrightarrow{p} c$. $\square$

Proof of Theorem 1. We use $\mathcal{L}(\cdot)$ and $\mathcal{L}(\cdot|\cdot)$ to denote marginal and conditional distributions of random variables. By construction of $\Pi_n \sim ESC_{[n]}(\boldsymbol{\mu})$, we have

$$\mathcal{L}(K_n) = \mathcal{L}(Y_n|E_n) \;\text{ and }\; \mathcal{L}(S_j) = \mathcal{L}(X_j|E_n) \qquad n \geq 1; j = 1, \dots, K_n \tag{1.3}$$

where $X_1, X_2, \dots \stackrel{iid}{\sim} \boldsymbol{\mu}$, $Y_n = \max\{k : \sum_{j=1}^{k} X_j \leq n\}$ and

$$E_n = \left\{ \omega \in \Omega : \text{ for some } k \geq 1 \text{ it holds } \sum_{j=1}^{k} X_j = n \right\}. \tag{1.4}$$

Theorem 1.1 implies $\liminf_{n\to\infty} Pr(E_n) > 0$. Also, the strong law of large numbers for renewal processes (see e.g. Barbu & Limnios 2009, Thm.2.3) implies that $n^{-1}Y_n$ converges almost surely to $(\sum_{s=1}^{\infty} s\mu_s)^{-1}$, and thus, also in probability. Since $n^{-1}Y_n \xrightarrow{p} (\sum_{s=1}^{\infty} s\mu_s)^{-1}$ and $\liminf_{n\to\infty} Pr(E_n) > 0$, it follows by Lemma 1.1 and Equation (1.3) that $n^{-1}K_n \xrightarrow{p} (\sum_{s=1}^{\infty} s\mu_s)^{-1}$, as desired. $\square$

Proof of Theorem 2. By construction of $\Pi_n \sim ESC_{[n]}(\boldsymbol{\mu})$ we have

$$\mathcal{L}(M_{s,n}) = \mathcal{L}(L_{s,n}|E_n) \qquad n \geq 1\,, \tag{1.5}$$

where $L_{s,n} = \sum_{j=1}^{Y_n} \mathbb{1}(X_j = s)$, and X_j, Y_n and E_n are defined as in the proof of Theorem 1. Since $\mathbb{1}(X_j = s)$ are independent and identically distributed Bernoulli random variables with mean μ_s and $\lim_{n\to\infty} Y_n = \infty$ almost surely, the strong law of large numbers imply that

$$\lim_{n\to\infty} \frac{L_{s,n}}{Y_n} = \lim_{n\to\infty} \frac{\sum_{j=1}^{Y_n} \mathbb{1}(X_j = s)}{Y_n} = \lim_{n\to\infty} \frac{\sum_{j=1}^{n} \mathbb{1}(X_j = s)}{n} = \mu_s \qquad \text{almost surely}\,. \tag{1.6}$$

Thus,

$$\lim_{n\to\infty} \frac{L_{s,n}}{n} = \lim_{n\to\infty} \frac{L_{s,n}}{Y_n}\frac{Y_n}{n} = \mu_s \left(\sum_{\ell=1}^{\infty} \ell\mu_\ell\right)^{-1} \qquad \text{almost surely}\,,$$

where we used the fact that $\lim_{n\to\infty} n^{-1}Y_n = (\sum_{\ell=1}^{\infty} \ell\mu_\ell)^{-1}$ almost surely by the strong law of large numbers for renewal processes (see e.g. Barbu & Limnios 2009, Thm.2.3Since almost sure convergence implies convergence in probability, we have $n^{-1}L_{s,n} \xrightarrow{p} \mu_s(\sum_{\ell=1}^{\infty} \ell\mu_\ell)^{-1}$, which implies $n^{-1}M_{s,n} \xrightarrow{p} \mu_s(\sum_{\ell=1}^{\infty} \ell\mu_\ell)^{-1}$ by Equation (1.5) and Lemma 1.1, as desired.

Consider now part (b). The size of cluster chosen uniformly at random from the clusters of Π_n is a random variable S_{U_n}, where $S_1, \ldots, S_{K_n}$ are the sizes of the clusters of Π_n and U_n is a random variable satisfying $U_n|\Pi_n \sim \text{Uniform}\{1, \ldots, K_n\}$. For any positive integer s, by the definition of U_n, we have $Pr(S_{U_n} = s|\Pi_n) = K_n^{-1}M_{s,n}$ and thus

$$Pr(S_{U_n} = s) = \mathbb{E}[Pr(S_{U_n} = s|\Pi_n)] = \mathbb{E}\left[\frac{M_{s,n}}{K_n}\right]. \tag{1.7}$$

By construction of $\Pi_n \sim ESC_{[n]}(\boldsymbol{\mu})$, we have

$$\mathcal{L}\left(\frac{M_{s,n}}{K_n}\right) = \mathcal{L}\left(\frac{L_{s,n}}{Y_n}\,\middle|\, E_n\right) \qquad n \geq 1\,,$$

and by Equation (1.6) we have $Y_n^{-1} L_{s,n} \xrightarrow{p} \mu_s$. Thus Lemma 1.1 implies $K_n^{-1} M_{s,n} \xrightarrow{p} \mu_s$. Since $K_n^{-1} M_{s,n} \in [0,1]$ it follows that $\mathbb{E}[K_n^{-1} M_{s,n}] \to \mu_s$ and thus, by Equation (1.7), $Pr(S_{U_n} = s) \to \mu_s$ as desired. $\square$

Proof of Theorem 3. Let X_j, Y_n and E_n be defined as in the proof of Theorem 1. By construction of $\Pi_n \sim ESC_{[n]}(\boldsymbol{\mu})$, we have

$$\mathcal{L}(M_n) = \mathcal{L}(L_n | E_n) \qquad n \geq 1 \tag{1.8}$$

where $L_n = \max\{X_1, \ldots, X_{Y_n}\}$. For any $\varepsilon > 0$ consider

$$\begin{aligned}\Pr\big(n^{-1} L_n > \varepsilon\big) &= \Pr\big(n^{-1} \max\{X_1, \ldots, X_{Y_n}\} > \varepsilon\big) \leq \Pr\big(n^{-1} \max\{X_1, \ldots, X_n\} > \varepsilon\big) \\ &= 1 - \Pr\big(\cap_{j=1}^{n} \{X_j \leq n\varepsilon\}\big) = 1 - \left(\sum_{j=1}^{\lceil \varepsilon n \rceil} \mu_j\right)^n ,\end{aligned}$$

where the inequality in the first row of the display follows from $Y_n \geq n$. Since $1 - x^n \leq n(1-x)$ for all $x \in [0,1]$ and $n \geq 1$, we have

$$\begin{aligned}1 - \left(\sum_{j=1}^{\lceil \varepsilon n \rceil} \mu_j\right)^n \;\leq\; n\left(1 - \sum_{j=1}^{\lceil \varepsilon n \rceil} \mu_j\right) \;=\; n \sum_{j=\lfloor \varepsilon n \rfloor + 1}^{\infty} \mu_j \\ = \varepsilon^{-1} \sum_{j=\lfloor \varepsilon n \rfloor + 1}^{\infty} \varepsilon n \mu_j \;\leq\; \varepsilon^{-1} \sum_{j=\lfloor \varepsilon n \rfloor + 1}^{\infty} j \mu_j \to 0 \quad \text{as } n \to \infty ,\end{aligned}$$

where the convergence $\lim_{n\to\infty} \sum_{j=\lfloor \varepsilon n \rfloor + 1}^{\infty} j\mu_j = 0$ follows from $\sum_{j=1}^{\infty} j\mu_j < \infty$ and $\lim_{n\to\infty} \lfloor \varepsilon n \rfloor + 1 = \infty$. Combining the last inequalities we obtain $\lim_{n\to\infty} \Pr(n^{-1} L_n > \varepsilon) \to 0$ or, in other words, $n^{-1} L_n \xrightarrow{p} 0$ as $n \to \infty$. Thus, by Equation (1.8), Lemma 1.1, and $\liminf_{n\to\infty} Pr(E_n) > 0$ (which follows from Theorem 1.1), we obtain $n^{-1} M_n \xrightarrow{p} 0$ as $n \to \infty$, as desired. $\square$

2 Samplers for Posterior Inference

In this section, we provide additional details regarding the samplers used for posterior inference. In the following derivations, we use the fact that, under the $ESC_{[n]}(P_{\boldsymbol{\mu}})$ model, the joint distribution of $\boldsymbol{\mu}$ and Π_n is

$$\Pr(d\boldsymbol{\mu}, \Pi_n) = \frac{P_{\boldsymbol{\mu}}(d\boldsymbol{\mu})}{\Pr(E_n)} \frac{K!}{n!} \prod_{j=1}^{K} S_j! \mu_{S_j} \, , \tag{2.1}$$

which can be easily derived using Equation (9). It follows that the conditional distribution of $\boldsymbol{\mu}$ given Π_n satisfies

$$\Pr(d\boldsymbol{\mu}|\Pi_n) \propto P_{\boldsymbol{\mu}}(d\boldsymbol{\mu}) \prod_{j=1}^{K} S_j! \mu_{S_j} \, . \tag{2.2}$$

The precise mathematical interpretation of Equation (2.2) is that the Radon–Nikodym derivative between the distribution of $\boldsymbol{\mu}$ conditional on Π_n and the distribution $P_{\boldsymbol{\mu}}$ is proportional to $\prod_{j=1}^{K} S_j! \mu_{S_j}$. The key aspect of Equation (2.2) is that the conditional distribution of $\boldsymbol{\mu}$ does not depend on the intractable term $\Pr(E_n|\boldsymbol{\mu})$, which makes the updates of $\boldsymbol{\mu}|\Pi_n$ in the MCMC algorithms for posterior sampling straightforward.

2.1 ESC-NB model

Recall that, for the ESC-NB model, $\boldsymbol{\mu} = \boldsymbol{\mu}(r, p)$ is a deterministic function of r and p specified by Equation (13).

Derivation of Equation (14). Since r, p are conditionally independent of $\boldsymbol{x}$ given Π_n we have $\Pr(r, p|\Pi_n, \boldsymbol{x}) = \Pr(r, p|\Pi_n)$. Then, combining Equation (2.2) with Equation (13) and the prior specification $r \sim Gamma(\eta_r, s_r)$, $p \sim Beta(u_p, v_p)$, we obtain

$$\begin{aligned}
\Pr(r, p|\Pi_n, \boldsymbol{x}) = \Pr(r, p|\Pi_n) &\propto \left(\frac{r^{\eta_r - 1} e^{-\frac{r}{s_r}}}{\Gamma(\eta_r) s_r^{\eta_r}}\right)\left(\frac{p^{u_p - 1}(1-p)^{v_p - 1}}{B(u_p, v_p)}\right)\prod_{j=1}^{K} S_j! \mu_{S_j} \\
&\propto r^{\eta_r - 1} e^{-\frac{r}{s_r}} p^{u_p - 1}(1-p)^{v_p - 1}\prod_{j=1}^{K} S_j! \gamma \frac{\Gamma(S_j + r) p^{S_j}}{\Gamma(r) S_j!} \\
&\propto r^{\eta_r - 1} e^{-\frac{r}{s_r}} p^{n + u_p - 1}(1-p)^{v_p - 1}\gamma^K \prod_{j=1}^{K} \frac{\Gamma(S_j + r)}{\Gamma(r)}\,,
\end{aligned}$$

which proves Equation (14). □

Derivation of Equation (15). Given the dependence structure of (r, p), Π_n, and $\boldsymbol{x}$, we have $\Pr(\Pi_n|r, p, \boldsymbol{x}) \propto \Pr(\Pi_n|r, p)\Pr(\boldsymbol{x}|\Pi_n)$ and thus

$$\Pr(z_i = j|\mathbf{z}_{-i}, \boldsymbol{x}, r, p) \propto \Pr(\boldsymbol{x}|\mathbf{z}_{-i}, z_i = j) \times \Pr(z_i = j|\mathbf{z}_{-i}, r, p)\,. \tag{2.3}$$

Corollary 1 implies

$$\Pr(z_i = j|\mathbf{z}_{-i}, r, p) \propto \begin{cases} (S_j + 1)\frac{\mu_{(S_j+1)}}{\mu_{S_j}} & \text{if } j = 1, \ldots, K_{-i}, \\ (K_{-i} + 1)\mu_1 & \text{if } j = K_{-i} + 1\,, \end{cases}$$

where, by Equation (13), we have $\mu_1 = \gamma\, r\, p$ and

$$\frac{\mu_{(S_j+1)}}{\mu_{S_j}} = \frac{\gamma \frac{\Gamma(S_j+1+r) p^{S_j+1}}{\Gamma(r)(S_j+1)!}}{\gamma \frac{\Gamma(S_j+r) p^{S_j}}{\Gamma(r) S_j!}} = p\frac{\Gamma(S_j + 1 + r)}{\Gamma(S_j + r)}\frac{S_j!}{(S_j+1)!} = p\frac{S_j + r}{S_j + 1}.$$

Therefore

$$\begin{aligned}
\Pr(z_i = j|\mathbf{z}_{-i}, r, p) &\propto \begin{cases} (S_j + 1) p \frac{S_j + r}{S_j + 1} & \text{if } j = 1, \ldots, k_{-i}, \\ (K_{-i} + 1)\gamma\, r\, p & \text{if } j = k_{-i} + 1\,, \end{cases} \\
&\propto \begin{cases} S_j + r & \text{if } j = 1, \ldots, K_{-i}, \\ (K_{-i} + 1)\gamma r & \text{if } j = K_{-i} + 1\,. \end{cases}
\end{aligned}$$

□

2.2 ESC-D model

Derivation of Equation (17). While in the ESC-NB model $\boldsymbol{\mu}$ is a deterministic function of r and p, for the ESC-D model we have $\boldsymbol{\mu}|r,p \sim Dir(\alpha, \boldsymbol{\mu}^{(0)})$, where $\boldsymbol{\mu}^{(0)} = \boldsymbol{\mu}^{(0)}(r,p)$ is defined in Equation (16). Thus, integrating out $\boldsymbol{\mu}$ in Equation (2.1) and using $r \sim Gamma(\eta_r, s_r)$ and $p \sim Beta(u_p, v_p)$, we obtain

$$P(r,p,\Pi_n) = \frac{1}{P(E_n)} \left(\frac{r^{\eta_r - 1} e^{-\frac{r}{s_r}}}{\Gamma(\eta_r) s_r^{\eta_r}} \right) \left(\frac{p^{u_p-1}(1-p)^{v_p-1}}{B(u_p,v_p)} \right) \mathbb{E}_{\boldsymbol{\mu} \sim Dir(\alpha, \boldsymbol{\mu}^{(0)})} \left[\frac{K!}{n!} \prod_{j=1}^{K} S_j! \mu_{S_j} \right]. \tag{2.4}$$

Using $M_{s,n} = \sum_{j=1}^{K} \mathbb{1}(S_j = s)$ and standard expressions for the moments of the Dirichlet distribution we obtain

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\mu} \sim Dir(\alpha, \boldsymbol{\mu}^{(0)})} \left[\frac{K!}{n!} \prod_{j=1}^{K} S_j! \mu_{S_j} \right] &= \frac{K!}{n!} \left(\prod_{s=1}^{M_n} s!^{M_{s,n}} \right) \mathbb{E}_{\boldsymbol{\mu} \sim Dir(\alpha, \boldsymbol{\mu}^{(0)})} \left[\prod_{s=1}^{M_n} \mu_s^{M_{s,n}} \right] \\
&= \frac{K!}{n!} \left(\prod_{s=1}^{M_n} s!^{M_{s,n}} \right) \frac{\Gamma(\alpha)}{\Gamma(K+\alpha)} \prod_{s=1}^{M_n} \frac{\Gamma(M_{s,n} + \alpha\, \mu_s^{(0)})}{\Gamma(\alpha\, \mu_s^{(0)})} \\
&= \frac{K!}{n!} \frac{\Gamma(\alpha)}{\Gamma(K+\alpha)} \prod_{s=1}^{M_n} \frac{s!^{M_{s,n}} \Gamma(M_{s,n} + \alpha\, \mu_s^{(0)})}{\Gamma(\alpha\, \mu_s^{(0)})}.
\end{aligned} \tag{2.5}$$

Combining Equations (2.4) and (2.5) we obtain that the joint distribution of r, p and Π_n under the ESC-D model satisfies

$$P(r,p,\Pi_n) \propto \frac{r^{\eta_r-1} e^{-\frac{r}{s_r}} p^{u_p-1} (1-p)^{v_p-1} K!}{\Gamma(K+\alpha)} \prod_{s=1}^{M_n} \frac{s!^{M_{s,n}} \Gamma(M_{s,n} + \alpha\, \mu_s^{(0)})}{\Gamma(\alpha\, \mu_s^{(0)})}. \tag{2.6}$$

The expression in Equation (17) follows from Equation (2.6) and the fact that $\Pr(r,p|\Pi_n, \boldsymbol{x}) = \Pr(r,p|\Pi_n)$ because r and p are conditionally independent of $\boldsymbol{x}$ given Π_n. □

3 Importance Sampler for ESC models

In this section we describe an importance sampler that can be used to generate weighted samples from random partitions $\Pi_n \sim ESC_{[n]}(P_{\boldsymbol{\mu}})$. The propose algorithm is not a fully standard importance sampler and thus we prove its validity in Theorem 3.1. In the context of Bayesian inferences, this algorithm can be used to generate samples from a $ESC_{[n]}(P_{\boldsymbol{\mu}})$ prior distribution for random partition. Unlike the rejection sampler described in the main document, we expect the importance sampler described here to be efficient even when $\mathbb{E}_{\boldsymbol{\mu}\sim P_{\boldsymbol{\mu}}}[(\sum_{s=1} s\mu_s)^{-1}]$ becomes small.

Algorithm 1. (Importance Sampler for ESC models)

1. *Sample $\boldsymbol{\mu} \sim P_{\boldsymbol{\mu}}$ and $S_1, \dots, S_R | \boldsymbol{\mu} \overset{iid}{\sim} \boldsymbol{\mu}$ until the first value R such that $\sum_{j=1}^R S_j \geq n$.*
2. *For $k = 1, \dots, R$ define $D_k = n - \sum_{j=1}^{k-1} S_j$ and $W = \sum_{k=1}^R \mu_{D_k}$.*
3. *Sample K from $\{1, \dots, R\}$ with probability* $\Pr(K = k) = \mu_{D_k}/W$*, and define the cluster allocation variables $(z_1, \dots, z_n)$ as a uniformly at random permutation of the vector*

$$(\underbrace{1, \dots, 1}_{S_1 \ times}, \underbrace{2, \dots, 2}_{S_2 \ times}, \dots\dots, \underbrace{K-1, \dots, K-1}_{S_{K-1} \ times}, \underbrace{K, \dots, K}_{D_K \ times}). \tag{3.1}$$

4. *Output the resulting partition Π_n as a weighted sample from the model $ESC_{[n]}(P_{\boldsymbol{\mu}})$ with importance weight* $\Pr(E_n)^{-1}W$.

Intuitively, given each vector of cluster sizes $(S_1, \dots, S_{k-1})$, Algorithm 1 considers the probability μ_{D_k} of sampling $S_k = D_k$ and weights the resulting vector of cluster sizes $(S_1, \dots, S_{k-1}, D_k)$ accordingly. The following theorem shows that the algorithm is valid, in the sense that it returns weighted samples from the distribution $ESC_{[n]}(P_{\boldsymbol{\mu}})$ that produce unbiased and consistent Monte Carlo estimators like standard Importance Sampling does.

Theorem 3.1. *For every real-valued function h defined over the space of partitions of $[n]$ we have*

$$\mathbb{E}_{(\Pi_n,W)\sim Alg1}[\Pr(E_n)^{-1}W\,h(\Pi_n)] = \mathbb{E}_{\Pi_n\sim ESC_{[n]}(P_{\boldsymbol{\mu}})}[h(\Pi_n)]\,, \tag{3.2}$$

where the notation $(\Pi_n, W) \sim Alg1$ means that Π_n is a partition produced by Algorithm 1 with associated importance weight $\Pr(E_n)^{-1}W$. Also, given a sequence $(\Pi_n^{(t)}, W^{(t)})_{t=1}^{\infty} \overset{iid}{\sim} Alg1$, we have

$$\frac{\sum_{t=1}^{T} W^{(t)}\,h(\Pi_n^{(t)})}{\sum_{t=1}^{T} W^{(t)}} \overset{a.s.}{\to} \mathbb{E}_{\Pi_n\sim ESC_{[n]}(P_{\boldsymbol{\mu}})}[h(\Pi_n)] \qquad \textit{as } T\to\infty\,. \tag{3.3}$$

Proof. By Proposition 1, or equivalently by (1.2), we have

$$\mathbb{E}_{\Pi_n\sim ESC_{[n]}(P_{\boldsymbol{\mu}})}[h(\Pi_n)] = \frac{1}{n!\Pr(E_n)}\sum_{\Pi_n} h(\Pi_n)\int K!\prod_{j=1}^{K}|S_j|!\mu_{S_j}P_{\boldsymbol{\mu}}(d\boldsymbol{\mu})\,, \tag{3.4}$$

where the sum over Π_n runs over all partitions of $[n]$. We now consider the expectation $\mathbb{E}_{(\Pi_n,W)\sim Alg1}[\Pr(E_n)^{-1}W\,h(\Pi_n)]$ and show that it is equal to the same expression. To simplify the proof we consider an equivalent formulation of Algorithm 1, where we simulate $\boldsymbol{\mu}\sim P_{\boldsymbol{\mu}}$ and $S_1,\dots,S_n|\boldsymbol{\mu} \overset{iid}{\sim} \boldsymbol{\mu}$ in Step 1; we set $W=\sum_{k=1}^{n}\mu_{D_k}$ with $\mu_{D_k}=0$ when $D_k\leq 0$ in Step 2; we sample K from $\{1,\dots,n\}$ with probability $\Pr(K=k)=\mu_{D_k}/W$ in Step 3 and leave the rest of the algorithm unchanged. The latter is an equivalent formulation of Algorithm 1 that is computationally less efficient because it generates additional variables $S_{R+1},\dots,S_n$ that are not necessary in practice, but is slightly simpler to analyse because it avoids the use of the auxiliary variable R. In order to keep the notation light, we denote $\mathbf{S}=(S_1,\dots,S_n)$ and $\mathbf{z}=(z_1,\dots,z_n)$ and we denote random variables (e.g. $\mathbf{S}$, K and $\mathbf{z}$)

and their possible realizations with the same symbols. We have

$$
\begin{aligned}
&\mathbb{E}_{(\Pi_n,W)\sim Alg1}[W\,h(\Pi_n)] \\
&= \int \sum_{\mathbf{S}\in\{1,2,\dots\}^n} \Pr(\mathbf{S}|\boldsymbol{\mu}) \sum_{K=1}^{n} \Pr(K|\mathbf{S},\boldsymbol{\mu}) \sum_{\mathbf{z}} \Pr(z|K,\mathbf{S}) W h(\Pi_n(\mathbf{z})) P_{\boldsymbol{\mu}}(d\boldsymbol{\mu}) \\
&= \int \sum_{\mathbf{S}\in\{1,2,\dots\}^n} \left(\prod_{j=1}^{n} \mu_{S_j}\right) \sum_{K=1}^{n} \frac{\mu_{D_K}}{\sum_{k=1}^{n}\mu_{D_k}} \sum_{\mathbf{z}} \frac{\left(\prod_{j=1}^{K-1} S_j!\right) D_K!}{n!} \left(\sum_{k=1}^{n}\mu_{D_k}\right) h(\Pi_n(\mathbf{z})) P_{\boldsymbol{\mu}}(d\boldsymbol{\mu}) \\
&= \frac{1}{n!} \int \sum_{\mathbf{S}\in\{1,2,\dots\}^n} \sum_{K=1}^{n} \left(\prod_{j=1}^{n} \mu_{S_j}\right) \mu_{D_K} \left(\prod_{j=1}^{K-1} S_j!\right) D_K! \sum_{\mathbf{z}} h(\Pi_n(\mathbf{z})) P_{\boldsymbol{\mu}}(d\boldsymbol{\mu})\,, \qquad (3.5)
\end{aligned}
$$

where the sum over $\mathbf{z}$ runs over all the vectors that can be obtained as a permutation of the vector in (3.1). Reorganizing the sum and exploiting the fact that $\mathbf{z}$ and Π_n depend only on $(S_1,\dots,S_{K-1})$ and K, we can integrate out $(S_K,\dots,S_n)$ and write (3.5) as

$$
\frac{1}{n!} \sum_{K=1}^{n} \sum_{(S_1,\dots,S_{K-1})\in\{1,2,\dots\}^{K-1}} \sum_{\mathbf{z}} h(\Pi_n(\mathbf{z})) \int \left(\prod_{j=1}^{K-1} \mu_{S_j} S_j!\right) \mu_{D_K} D_K! P_{\boldsymbol{\mu}}(d\boldsymbol{\mu})\,.
$$

Re-writing the sums above in terms of the resulting partition Π_n, and exploiting the fact that each partition Π_n can be obtained through $K!$ different cluster assignments $\mathbf{z}$, we have

$$
\mathbb{E}_{(\Pi_n,W)\sim Alg1}[W\,h(\Pi_n)] = \frac{1}{n!} \sum_{\Pi_n} h(\Pi_n) K! \left(\prod_{j=1}^{K-1} |S_j|!\mu_{S_j}\right) \mu_{D_K} D_K!\,, \qquad (3.6)
$$

where the sum over Π_n runs over all partitions of $[n]$ and the cluster sizes of Π_n are denoted as $(S_1,\dots,S_{K-1},D_K)$ for coherence with the notation of Algorithm 1. Comparing (3.4) and (3.6) we obtain (3.2).

The almost sure convergence in (3.3) follows by applying the strong law of large numbers to both numerator and denominator in the fraction on the left-hand side, and then noting that by (3.2) we have

$$
\frac{\mathbb{E}_{(\Pi_n,W)\sim Alg1}[W\,h(\Pi_n)]}{\mathbb{E}_{(\Pi_n,W)\sim Alg1}[W]} = \frac{\Pr(E_n)\mathbb{E}_{\Pi_n\sim ESC_{[n]}(P_{\boldsymbol{\mu}})}[h(\Pi_n)]}{\Pr(E_n)} = \mathbb{E}_{\Pi_n\sim ESC_{[n]}(P_{\boldsymbol{\mu}})}[h(\Pi_n)]\,.
$$

$\square$

Note that the normalized importance weight $\Pr(E_n)^{-1}W$ involves the constant $\Pr(E_n)$ that is typically not available in closed form. However, this is not a problem because the self-normalized importance sampling estimator defined in (3.3) is not sensitive to multiplicative constants in the importance weights. Thus, one can directly use W as an importance weight, ignoring the unknown constant $\Pr(E_n)$.

4 Likelihood Derivation for Entity Resolution

In this section, we provide the derivation of the likelihood that is used in our ER task. Recall that the observed data $\boldsymbol{x}$ consist of n records $(x_i)_{i=1}^n$ and each record x_i contains L fields $(x_{i\ell})_{\ell=1}^L$. Each field ℓ is associated to two hyperparameters: a distortion probability $\beta_\ell \in (0,1)$ and a density vector $\boldsymbol{\theta}_\ell = (\theta_{\ell d})_{d=1}^{D_\ell} \in [0,1]^{D_\ell}$, where D_ℓ denotes the number of categories for field ℓ and $\sum_{d=1}^{D_\ell} \theta_{\ell d} = 1$. As mentioned in Section 4, we assume that clusters are conditionally independent given the partition Π_n and the hyperparameters $\boldsymbol{\beta} = (\beta_\ell)_{\ell=1}^L$ and $\boldsymbol{\theta} = (\boldsymbol{\theta}_\ell)_{\ell=1}^L$, resulting in

$$P(\boldsymbol{x}|\Pi_n, \boldsymbol{\beta}, \boldsymbol{\theta}) = \prod_{j=1}^{K} \prod_{\ell=1}^{L} P(\boldsymbol{x}_{j\ell}|\beta_\ell, \boldsymbol{\theta}_\ell)\,, \tag{4.1}$$

where $\boldsymbol{x}_{j\ell} = \{x_{i\ell} : i \in C_j\}$. For each $C_j \in \Pi_n$, the distribution of $\boldsymbol{x}_{j\ell}|\beta_\ell, \boldsymbol{\theta}_\ell$ is given by

$$y_{j\ell} \sim \boldsymbol{\theta}_\ell \tag{4.2}$$

$$x_{i\ell}|y_{j\ell} \overset{iid}{\sim} \beta_\ell \boldsymbol{\theta}_\ell + (1-\beta_\ell)\delta_{y_{j\ell}} \qquad i \in C_j\,, \tag{4.3}$$

where $y_{j\ell}$ represent the correct l-th feature of the entity associated to cluster C_j, and β_ℓ is the probability of distortion in feature ℓ. Integrating out $y_{j\ell}$ from Equations (4.2) and

(4.3) it follows

$$
\begin{aligned}
P(\boldsymbol{x}_{j\ell}|\beta_\ell, \boldsymbol{\theta}_\ell) &= \sum_{d=1}^{D_\ell} P(y_{j\ell} = d|\boldsymbol{\theta}_\ell) \prod_{i\in C_j} P(x_{i\ell}|\beta_\ell, y_{j\ell} = d) \\
&= \sum_{d=1}^{D_\ell} \theta_{\ell d} \prod_{i\in C_j} (\beta_\ell \theta_{\ell x_{n\ell}} + (1-\beta_\ell)\mathbb{1}(x_{i\ell} = d)) \\
&= \left(\prod_{i\in C_j} \beta_\ell \theta_{\ell x_{i\ell}}\right) \sum_{d=1}^{D_\ell} \theta_{\ell d} \prod_{i\in C_j} \frac{(\beta_\ell \theta_{\ell x_{i\ell}} + (1-\beta_\ell)\mathbb{1}(x_{i\ell} = d))}{\beta_\ell \theta_{\ell x_{i\ell}}} \qquad (4.4)
\end{aligned}
$$

To proceed we denote by $x_{1\ell}^{(j)}, \ldots, x_{m^{(j)}\ell}^{(j)}$ the collection of unique values in $\boldsymbol{x}_{j\ell}$ and by $q_{1\ell}^{(j)}, \ldots, q_{m^{(j)}\ell}^{(j)}$ the corresponding frequencies, meaning that for each $i \in \{1, \ldots, m^{(j)}\}$ the value $x_{i\ell}^{(j)}$ appears exactly $q_{i\ell}^{(j)}$ times in $\boldsymbol{x}_{j\ell}$. Then from Equation (4.4) we have

$$
P(\boldsymbol{x}_{j\ell}|\beta_\ell, \boldsymbol{\theta}_\ell) = \left(\prod_{i\in C_j} \beta_\ell \theta_{\ell x_{i\ell}}\right) f(\boldsymbol{x}_{j\ell}, \beta_\ell, \boldsymbol{\theta}_\ell)\,, \qquad (4.5)
$$

where

$$
f(\boldsymbol{x}_{j\ell}, \beta_\ell, \boldsymbol{\theta}_\ell) = 1 - \sum_{i=1}^{m^{(j)}} \theta_{\ell x_{i\ell}^{(j)}} + \sum_{i=1}^{m^{(j)}} \theta_{\ell x_{i\ell}^{(j)}} \left(\frac{\beta_\ell \theta_{\ell x_{i\ell}^{(j)}} + (1-\beta_\ell)}{\beta_\ell \theta_{\ell x_{i\ell}^{(j)}}}\right)^{q_{i\ell}^{(j)}}. \qquad (4.6)
$$

Combining Equations (4.5) and (4.1), we obtain the desired likelihood function

$$
P(\boldsymbol{x}|\Pi_n, \boldsymbol{\beta}, \boldsymbol{\theta}) = \left(\prod_{\ell=1}^{L} \prod_{i=1}^{n} \beta_\ell \theta_{\ell x_{i\ell}}\right) \prod_{j=1}^{K} \prod_{\ell=1}^{L} f(\boldsymbol{x}_{j\ell}, \beta_\ell, \boldsymbol{\theta}_\ell)\,. \qquad (4.7)
$$

5 Implementation details of MCMC algorithms

In this section, we provide more details on the MCMC algorithms used to approximate posterior quantities of interest in Section 5 of the paper. Posterior computation is performed using the samplers described in Sections 3.3.1-3.3.2 of the paper. The results are based on MCMC runs of 2×10^7 iterations, thinning every $1,000$ iterations[1] and then discarding the first 5000 out of 20000 resulting samples as burn-in. In all cases standard convergence diagnostics and plotting of traceplots did not highlight significant mixing issues. In the real data experiments of Section 5.3, four MCMC runs for each dataset were performed to reduce Monte Carlo error, see more details below. MCMC runtimes were roughly 1 hour per run for Section 5.1, 20 hours per run for Section 5.3.1 and 50 hours per run for Section 5.3.1. The algorithms were implemented in R and a desktop computer with 32GB of RAM and an i9 Intel processor was used to perform the simulations.

When implementing the *chaperones algorithm* of Miller et al. (2015, Appendix B), we used a non-uniform probability of selecting chaperones $i, j \in \{1, \ldots, n\}$, assigning higher probability to pairs of records whose values agree on a large number of randomly selected fields [2]. This approach greatly improves convergence of the algorithm and respects the assumptions that the probability of selecting any pair of records is strictly greater than zero and is independent of the current partition, which are necessary to ensure the validity

[1] More precisely, we perform 2×10^4 MCMC iterations, and within each iteration perform one update of the global parameters and 1000 updates of the partition given the global parameters using the chaperones algorithm.

[2] The latter is done by first sampling a random number N_f of fields between 0 and L, then picking N_f fields uniformly at random and then pick the chaperones $i, j \in \{1, \ldots, n\}$ uniformly at random among those that agree on those N_f fields. Other strategies could be used to favor pairs of chaperones that agree on various fields and we claim no optimality of this specific implementation.

of the chaperones algorithm (see Miller et al. 2015, Appendix B). We expect the use of the chaperones algorithm with non-uniform proposals to be particularly beneficial in contexts with very small clusters, while for cases of larger clusters we expect the latter algorithm to behave similarly to standard split and merge schemes (Jain & Neal 2004).

Figures 1 and 2 show the traceplots for K, FNR and FDR for the four chains used for the SDS and SIPP data sets, respectively. No issues of convergence are observed in either case. However, the mixing of the chains for the SDS is slower compared to the SIPP data. Table 1 displays the estimated MCMC standard errors for the estimation of the average posterior FNR and FDR using the four chains and discarding the first 5,000 iterations of each run as a burn-in. The MCMC standard errors were computed using the function *summary.mcmc* from the R package **CODA** (Plummer et al. 2006). The estimated standard errors are all between 0.01% and 0.04%, indicating that the FNR and FDR estimates presented in Section 5.3 of the main document are reliable up to one decimal place (in percentage), which is the level of precision reported in Tables 2 and 3 of the main document.

	SDS		SIPP	
Model	FNR SE	FDR SE	FNR SE	FDR SE
DP	0.03	0.04	0.02	0.01
PY	0.02	0.04	0.02	0.01
ESCNB	0.02	0.04	0.01	0.02
ESCD	0.02	0.02	0.01	0.01

Table 1: Time-series MCMC error (in percentages) for the posterior expected values of FNR and FDR for SDS and SIPP data sets.

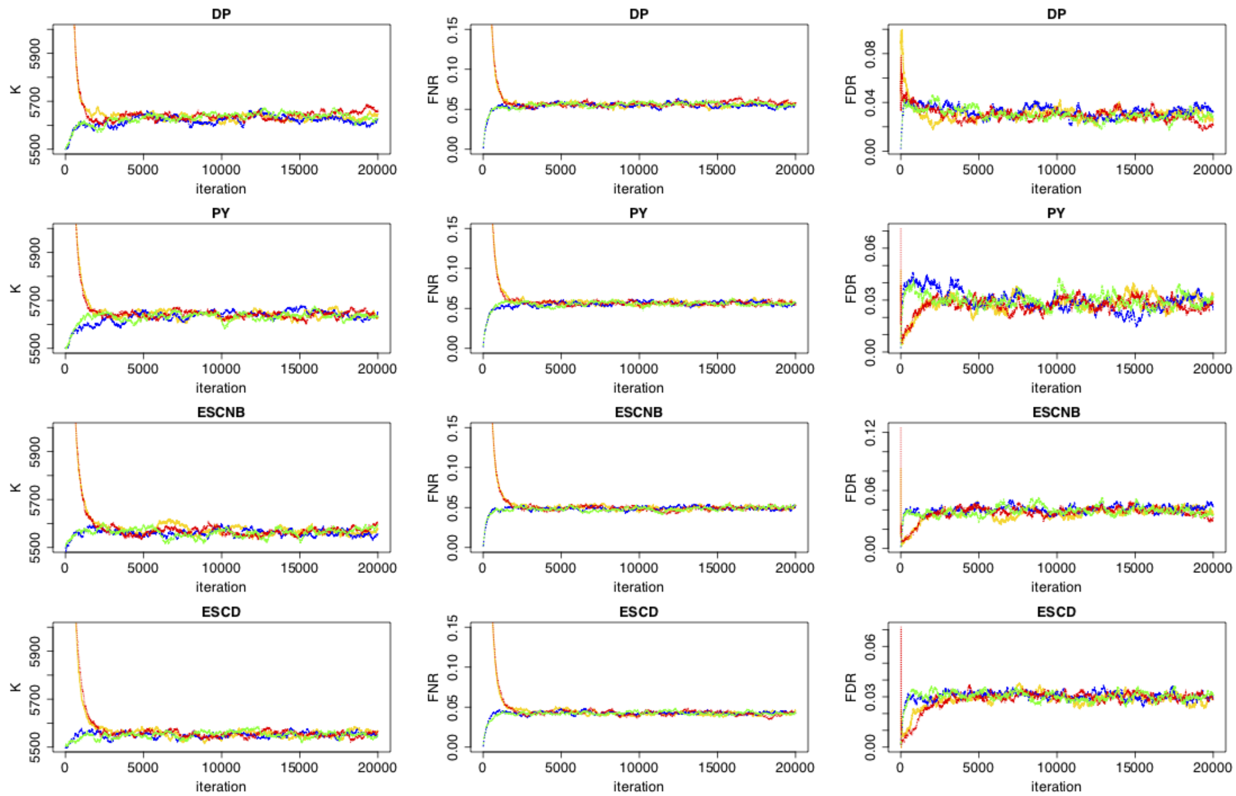

Figure 1: ***SDS dataset.*** *Trace plots of number of clusters (K), false negative rate (FNR) and false discovery rate (FDR) for four chains of 20,000 iterations of DP, PY, ESC-NB and ESC-D models for SDS data set of* $K = 5,500$*.*

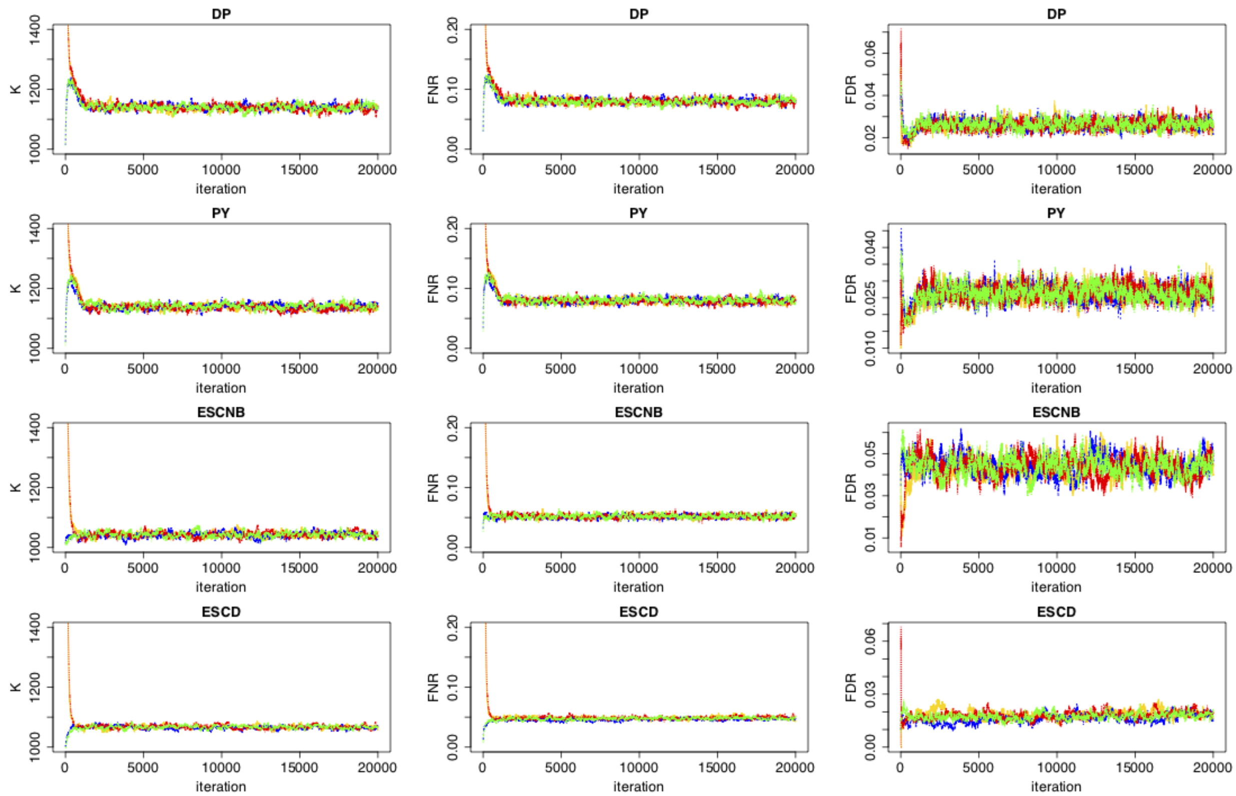

Figure 2: ***SIPP dataset.*** *Trace plots of number of clusters (K), false negative rate (FNR) and false discovery rate (FDR) for four chains of 20,000 iterations of DP, PY, ESC-NB and ESC-D models for SIPP data set of* $K = 1,000$.

6 Additional results for the simulation study

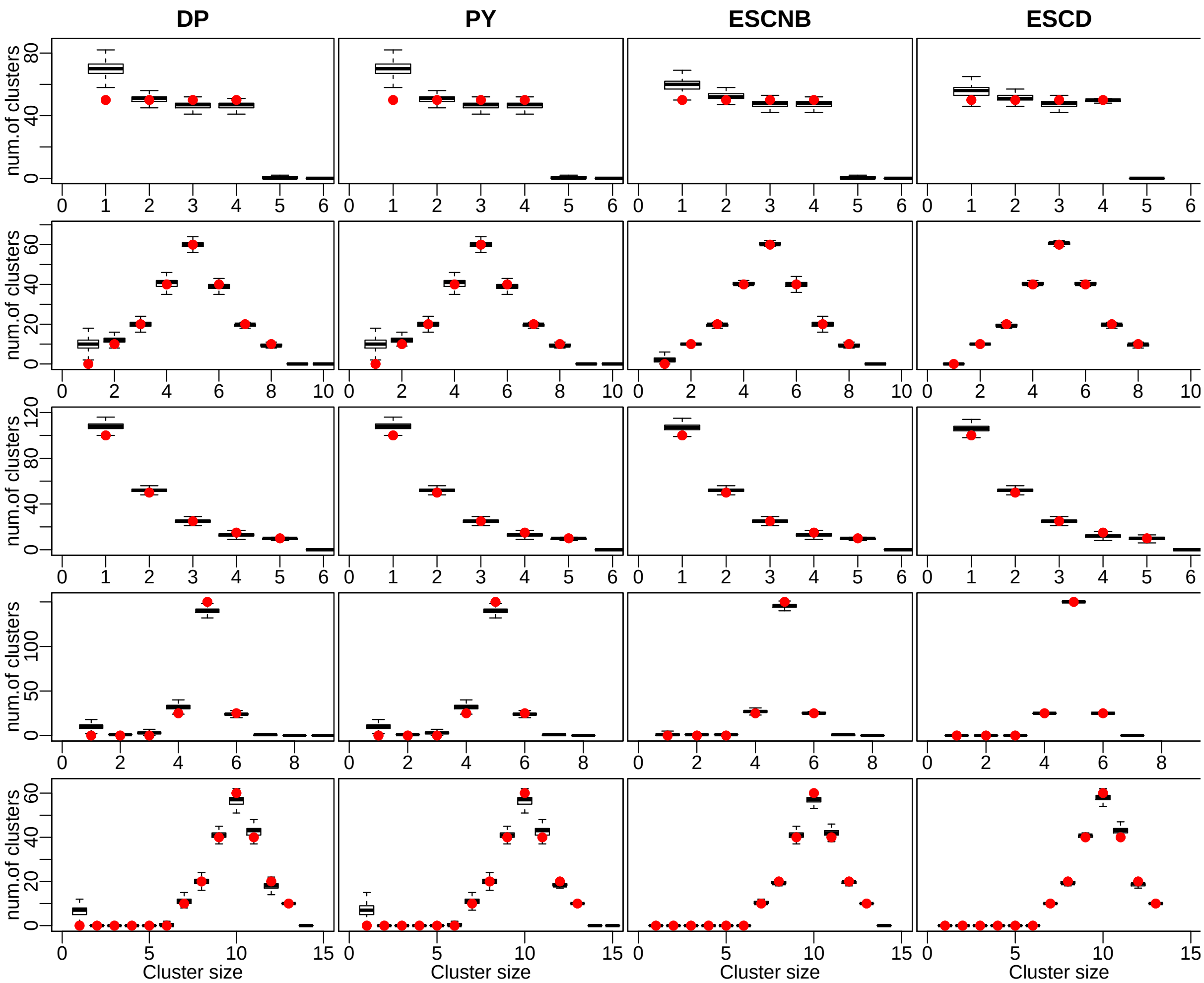

Figure 3: Posterior distribution of the number of clusters of each size (black boxplots based on 20k MCMC samples from the posterior after thinning) versus number of clusters of each size in the true data-generating partition (red dots) for $\beta = 0.01$. Each column corresponds to a different prior for the partition, and each row to a different data generating partition.

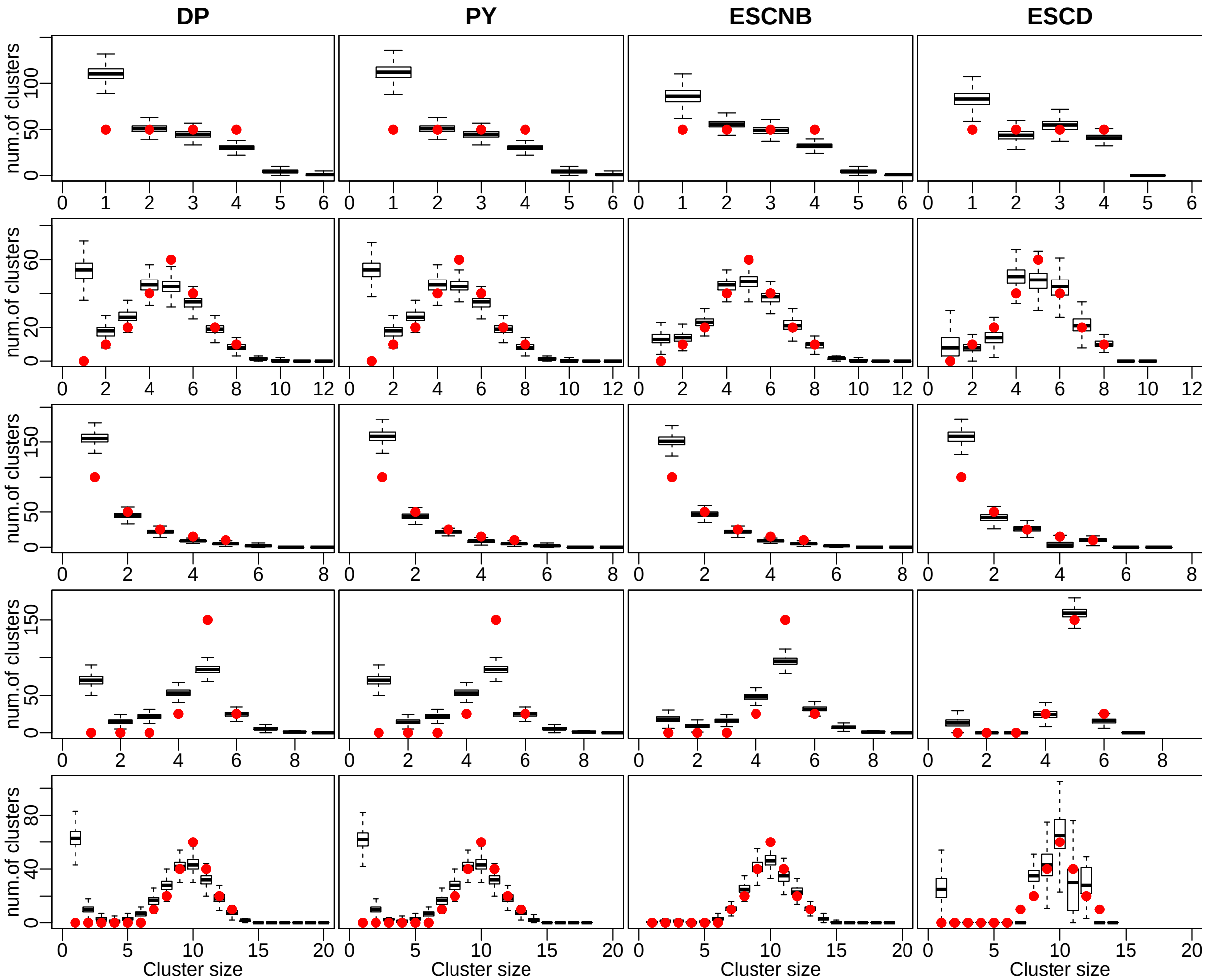

Figure 4: Posterior distribution of the number of clusters of each size (black boxplots based on 20k MCMC samples from the posterior after thinning) versus number of clusters of each size in the true data-generating partition (red dots) for $\beta = 0.10$. *Each column corresponds to a different prior for the partition, and each row to a different data generating partition.*

References

Barbu, V. S. & Limnios, N. (2009), *Semi-Markov chains and hidden semi-Markov models toward applications: their use in reliability and DNA analysis*, Vol. 191, Springer Science & Business Media.

Jain, S. & Neal, R. M. (2004), 'A split-merge markov chain monte carlo procedure for the dirichlet process mixture model', *Journal of computational and Graphical Statistics* **13**(1), 158–182.

Miller, J., Betancourt, B., Zaidi, A., Wallach, H. & Steorts, R. (2015), 'The Microclustering Problem: When the Cluster Sizes Don't Grow with the Number of Data Points', *NIPS Bayesian Nonparametrics: The Next Generation Workshop Series* .

Plummer, M., Best, N., Cowles, K. & Vines, K. (2006), 'Coda: Convergence diagnosis and output analysis for mcmc', *R News* **6**(1), 7–11.