JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2020, VOL. 00, NO. 0, 1–12
https://doi.org/10.1080/10618600.2020.1807995

Taylor & Francis
Taylor & Francis Group

Fast Search and Estimation of Bayesian Nonparametric Mixture Models Using a Classification Annealing EM Algorithm

George Karabatsos

Department of Educational Psychology, and Department of Mathematics, Statistics, and Computer Science, University of Illinois-Chicago, Chicago, IL

ABSTRACT

Bayesian nonparametric (BNP) infinite-mixture models provide flexible and accurate density estimation, cluster analysis, and regression. However, for the posterior inference of such a model, MCMC algorithms are complex, often need to be tailor-made for different BNP priors, and are intractable for large datasets. We introduce a BNP classification annealing EM algorithm which employs importance sampling estimation. This new fast-search algorithm, for virtually any given BNP mixture model, can quickly and accurately calculate the posterior predictive density estimate (by posterior averaging) and the maximum a-posteriori clustering estimate (by simulated annealing), even for datasets containing millions of observations. The algorithm can handle a wide range of BNP priors because it primarily relies on the ability to generate prior samples. The algorithm can be fast because in each iteration, it performs a sampling step for the (missing) clustering of the data points, instead of a costly E-step; and then performs direct posterior calculations in the M-step, given the sampled (imputed) clustering. The new algorithm is illustrated and evaluated through BNP Gaussian mixture model analyses of benchmark simulated data and real datasets. MATLAB code for the new algorithm is provided in the supplementary materials. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received June 2019
Revised May 2020

KEYWORDS
Bayesian nonparametrics; Clustering; Completely random measures; Density estimation; Normalized random measures

1. Introduction

Bayesian nonparametric (BNP) infinite-mixture models provide flexible and accurate methods of density estimation, cluster analysis, and regression, for many scientific fields (e.g., Daumé III 2007; Hjort et al. 2010; Mitra and Müller 2015, and references therein). A typical BNP mixture model has a mixing distribution defined by a completely random measure (CRM) (introduced by Kingman 1967), an infinite mixture of point mass distributions assigned a BNP (CRM) prior distribution (Lijoi and Prünster 2010). For example, the popular Dirichlet process mixture (DPM) model (Lo 1984) has a mixing distribution defined by a Dirichlet process (DP) (Ferguson 1973), a random probability measure (BNP prior) which can be obtained by normalizing the increments of a gamma CRM. Indeed, other more general and flexible classes of BNP priors are available.

The increasing availability of big data through cheap computing power has motivated developments of various deterministic fast-search algorithms for estimating BNP mixture models (e.g., Daumé III 2007; Raykov, Boukouvalas, and Little 2016; Fuentes-García, Meña, and Walker 2019; Zuanetti et al. 2019, and references therein). Such an algorithm provides faster alternative to MCMC, sequential Monte Carlo (SMC), and related algorithms which can compute or converge slowly for such data. Certain fast-search algorithms, including variational Bayes, predictive recursion, and sequential methods, also aim to improve computational speed, but do so by relying on factorization or prediction rule assumptions which depart from the underlying probabilistic properties of the general BNP mixture model (Raykov, Boukouvalas, and Little 2016; Fortini and Petrone 2020; Zuanetti et al. 2019).

A typical fast-search algorithm can rapidly produce the approximate maximum-a-posteriori (MAP) estimate of the clustering of the data points, and the posterior predictive density estimate for the given BNP mixture model, within seconds or minutes, even from a large dataset, while making one or few passes over all the data points. For the model, in research practice, these are the estimates of main interest from the posterior distribution (e.g., Daumé III 2007; Raykov, Boukouvalas, and Little 2016), while the MAP estimator of the clusters is coherent (Fuentes-García, Meña, and Walker 2019).

The current fast-search algorithms do not easily apply to the entire class of BNP priors, but instead are limited to BNP priors which admit tractable representations, such as the class of Gibbs type priors (DeBlasi et al. 2015), and most stick-breaking priors (Ishwaran and James 2001), including the normalized generalized gamma process (see Lijoi, Meña, and Prünster 2007), the Poisson–Dirichlet process, and submodels such as the DP. However, other BNP priors can provide better fit and more realistic clustering of data (Lijoi and Prünster 2010). But the current fast-search algorithms are not easily applicable to less-tractable BNP (CRM) priors, such as the generalized Dirichlet process (Lijoi, Meña, and Prünster 2005a), the stable 3-parameter beta

CONTACT George Karabatsos gkarabatsos1@gmail.com University of Illinois-Chicago, Chicago, IL 60607.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

(Teh and Gorür 2009) process, other priors with explicit series (e.g., inverse Lévy, species sampling) or superposition representations (Campbell et al. 2019), or any future novel BNP priors.

Also, the current fast-search algorithms are deterministic. Thus, for research practice, the literature has suggested restarting such an algorithm for several random starting parameter values or permutations of the data ordering, to produce estimates that are not trapped in a suboptimal local posterior mode or overly influenced by poor starting values. In contrast, MCMC and SMC algorithms have been developed for many BNP priors. However, such algorithms typically need to be tailor-made to the specific BNP prior considered, and appear complex and very different across the BNP priors. Further, MAP clustering estimation from MCMC (or SMC) posterior samples is nontrivial (Rastelli and Friel 2018).

This article introduces the BNP-CAEM algorithm, a novel fast-search algorithm for clustering and posterior predictive density estimation, which employs importance sampling. This extends the classification annealing EM (CAEM) algorithm, a K-means type algorithm for maximum likelihood estimation (MLE) of the Gaussian mixture model (Celeux and Govaert 1992, sec. 4.2) without importance sampling. The BNP-CAEM algorithm is based on the complete likelihood function for the mixture model, which, for any set of "missing" cluster assignments of the data points, is defined by the component density likelihood of the data points (resp.) times a multinomial kernel density for the cluster groups frequencies. This algorithm easily applies to any BNP mixture model defined by any chosen BNP prior, with simple changes to the algorithm; relies on any suitable, fixed finite-dimensional truncation approximation of the prior (see Arbel and Prünster 2017); all while merely relying on the ability to sample from the chosen BNP prior distribution.

The *default* BNP-CAEM algorithm applies to any usual BNP prior with conjugate baseline measure for the mixture component parameters. In each algorithm iteration, the C-step generates a sample from the current posterior predictive distribution of the clustering of the data points; and the M-step directly calculates posterior updates of the component predictive densities and the mixture weight parameters, given the sampled (imputed) clustering. The C-step replaces the computationally costly E-step of the original EM algorithm (Dempster, Laird, and Rubin 1977) with a less-costly sampling step. The M-step updates the mixture weights using an importance sampling estimator, based on a large number of BNP prior (proposal) samples of the mixture weights, which are (resp.) assigned multinomial (kernel) importance weights given the updated cluster group frequencies. The same large sample can be reused in the M-step over BNP-CAEM iterations, a computational savings feature of the general importance sampling method (Beckman and McKay 1987). The BNP-CAEM algorithm can be extended to handle non-conjugate priors for the component parameters, with some extra computational cost, by adding an importance sampling estimator for the component predictive densities based on a prior (proposal) sample of these parameters.

Over initial iterations of the general BNP-CAEM algorithm, the temperature is at the maximum value of 1. During this time, the algorithm acts as a stochastic EM algorithm (Celeux and Govaert 1992) which eventually produces a sequence of ergodic time-homogeneous Markov chain of posterior predictive density estimates. This sequence converges to a random density estimate variable, which has the stationary distribution of the Markov chain as the number of iterations grows, and has an asymptotic normal distribution centered on the true density when the data sample size is large (from Nielsen 2000). Hence, the density estimates produced over the initial converged BNP-CAEM iterations can be averaged to provide the final density estimate, a multiple (missing clustering) imputation estimate. In later BNP-CAEM algorithm iterations, using annealing, the temperature is gradually decreased toward 0, so that the amount of randomness in the simulations decreases with the iterations, ending up with an approximate MAP clustering estimate of the data points. Yet, because this algorithm is stochastic, it can produce clustering and density estimates that can randomly escape suboptimal local posterior modes. Thus, it is not necessary to restart this algorithm for several random starting values, unlike the other fast search algorithms.

The BNP-CAEM algorithm is described next. Details are given by Section 2.4, after Section 2.1 concisely reviews CRMs and BNP priors, Section 2.2 describes the ϵ-approximation truncation methods used for BNP (CRM) priors, and Section 2.3 represents the posterior distribution for general BNP mixture models. Appendices A1–A6 in the supplementary materials give more technical details. In Section 3, the BNP-CAEM algorithm is illustrated through the BNP Gaussian mixture model analysis of simulated and real benchmark datasets, and evaluated and compared with standard MCMC, VB, and EM MLE algorithms, in terms of density and clustering estimation accuracy, and computation time. Finally, Section 4 discusses how the BNP-CAEM algorithm can be used to accelerate MCMC convergence, used in a distributed parallel computing scheme for massive data analysis, and used for real-time streaming data analysis.

2. Methodology

2.1. Review of CRMs

Let $\mathbb{Y}$ be a complete and separable metric space, and $M_{\mathbb{Y}}$ be the space of boundedly finite measures on $\mathbb{Y}$, with Borel σ-algebras $\mathcal{Y}$ and $\mathcal{M}_{\mathbb{Y}}$, respectively. (Then $\mu \in M_{\mathbb{Y}}$ implies $\mu(A) < \infty$ for any bounded set A.) A basic object in BNP modeling is the CRM,

$$\widetilde{\mu}(\cdot) = \sum_{j=1}^{\infty} \mathcal{J}_j \delta_{\boldsymbol{\theta}_j}(\cdot), \tag{2.1}$$

an almost-surely discrete random measure that is defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and takes on values in $(M_{\mathbb{Y}}, \mathcal{M}_{\mathbb{Y}})$; with $\widetilde{\mu}(A_1), \ldots, \widetilde{\mu}(A_K)$ mutually independent for any $K > 1$ pairwise-disjoint sets $A_1, \ldots, A_K$ in $\mathcal{Y}$; and defined by random jumps (masses) $(\mathcal{J}_j)_{j\geq 1}$ at random locations $(\boldsymbol{\theta}_j)_{j\geq 1}$, where $\delta_{\boldsymbol{\theta}}$ is a unit point mass measure at $\boldsymbol{\theta}$ (Kingman 1967).

The distribution of the CRM (2.1) has expectation ($\mathbb{E}$) determined by its Laplace functional transform, with Lévy–Khintchine representation:

$$\begin{aligned} &\mathbb{E}[\exp\{-\textstyle\int_{\mathbb{Y}}\varphi(y)\widetilde{\mu}(\mathrm{d}y)\}] \qquad (2.2)\\ &\quad = \exp\left\{-\int_{\mathbb{R}_+\times\mathbb{Y}}[1-\exp\{-\upsilon\varphi(y)\}]\nu(\mathrm{d}\upsilon,\mathrm{d}y)\right\}, \end{aligned}$$

for any measurable function $\varphi : \mathbb{Y} \to \mathbb{R}$, with $\int_{\mathbb{Y}} |\varphi| \mathrm{d}\widetilde{\mu} < \infty$ and $\int_{\mathbb{R}_+ \times \mathbb{Y}} \min\{\upsilon, 1\} \nu(\mathrm{d}\upsilon, \mathrm{d}y) < \infty$. Above, ν is the Lévy intensity measure on $\mathbb{R}_+ \times \mathbb{Y}$ that describes the distribution of the jump sizes $\mathcal{J}_j$'s and locations $\boldsymbol{\theta}_j$'s of the CRM, and can be conveniently rewritten as $\nu(\mathrm{d}\upsilon, \mathrm{d}y) = \rho(\mathrm{d}\upsilon \mid y) \alpha(\mathrm{d}y)$, where ρ is a transition kernel on $\mathbb{R}_+ \times \mathbb{Y}$ controlling the jump intensity, and α is a measure on $\mathbb{Y}$ determining the locations of the jumps. The Lévy intensity measure ν is *homogeneous* if it has the form $\nu(\mathrm{d}\upsilon, \mathrm{d}y) = \rho(\mathrm{d}\upsilon) \alpha(\mathrm{d}y)$, which implies that the jumps and locations are independent.

Any BNP prior is uniquely defined by its Lévy intensity ν for the corresponding CRM, and thus can be denoted by $\widetilde{\mu} \sim \mathrm{CRM}(\nu)$ (e.g., Lijoi and Prünster 2010). Virtually any standard BNP (CRM) prior (almost-surely) supports positive and finite CRMs, such that $0 < \widetilde{\mu}(\mathbb{Y}) < \infty$ with probability 1, based on the conditions that $\rho(\mathbb{R}_+) = \infty$ and $\alpha(\mathbb{Y}) \in (0, \infty)$. The latter condition allows α to be treated as a probability measure, and is usually rewritten as $\alpha(\mathrm{d}y) = aG_0(\mathrm{d}y)$, with positive parameter $a > 0$, and with G_0 a probability measure on $\mathbb{Y}$ (Regazzini, Lijoi, and Prünster 2003).

A normalized random measure with independent increments (NRMI) refers to a CRM $\widetilde{\mu}$ (with BNP prior) that is transformed to the random probability measure (r.p.m.), $G(\cdot) = \widetilde{\mu}(\cdot)/\widetilde{\mu}(\mathbb{Y})$ (Regazzini, Lijoi, and Prünster 2003). NRMIs define a large class of BNP priors. Table 1 presents important examples of CRM (BNP), among others. Inhomogeneous BNP (CRM) priors (not shown) include neutral to the right and extended gamma processes (see Lijoi and Prünster 2010).

Any BNP (CRM) prior can admit another more tractable representation that implies $\widetilde{\mu} \sim \mathrm{CRM}(\nu)$ based on the prior-defining intensity measure ν (e.g., Campbell et al. 2019). For example, an inverse-Lévy (series) representation of the CRM $\widetilde{\mu}$ (BNP prior) is given by (2.1), with decreasing jumps $\mathcal{J}_1 \geq \mathcal{J}_2 \geq \mathcal{J}_3 \geq \cdots$ obtained by $\mathcal{J}_j = \nu^{\leftarrow}(\xi_j) = \inf\{\upsilon : \nu([\upsilon, \infty), \mathbb{Y}) \leq \xi_j\}$ and $\boldsymbol{\theta}_j \overset{\text{ind}}{\sim} G(\mathbf{t} \mid \mathcal{J}_j)$, where $\xi_j = \sum_{\ell=1}^{j} E_\ell$, $E_\ell \overset{\text{iid}}{\sim} \mathrm{Exp}(1)$, are the ordered jumps times of a standard Poisson process of unit rate on $\mathbb{R}_+$ (Ferguson and Klass 1972). Here, $\nu([\upsilon, \infty), \mathbb{Y})$ is the Lévy jump intensity of $\mathrm{CRM}(\nu)$, a decreasing function of υ. And $\boldsymbol{\psi} = (\mathcal{J}_1, \mathcal{J}_2, \ldots) = (\mathcal{J}_j)_{j=1}^{\infty}$.

Some normalized BNP priors admit a series representation in terms of a species sampling model (Pitman 1996), which is defined by the r.p.m.:

$$G(\cdot) = \sum_{j=1}^{\infty} \omega_j(\boldsymbol{\psi}) \delta_{\boldsymbol{\theta}_j}(\cdot); \quad \omega_j(\boldsymbol{\psi}) \geq 0,$$

$$\textstyle\sum_{j=1}^{\infty} \omega_j(\boldsymbol{\psi}) = 1; \quad \boldsymbol{\psi} \sim \Pi, \quad \boldsymbol{\theta}_j \overset{\text{iid}}{\sim} G_0, \tag{2.3a}$$

with mixture weights $\omega_j(\boldsymbol{\psi})$ and parameters $\boldsymbol{\psi}$ and $\boldsymbol{\theta}_j$ that may depend on covariates.

Species sampling models comprise the largest class of normalized homogeneous CRMs (BNP priors). Such a model is

Table 1. Definitions of various CRMs (BNP priors).

CRM	Lévy intensity measure, $\nu(\mathrm{d}\upsilon, \mathrm{d}y) =$	Special cases:
Normalized generalized Gamma process hNRMI, $\mathcal{NGG}(\tau, \gamma, a, G_0)$	$\dfrac{\exp(-\tau\upsilon)}{\Gamma(1-\gamma)\upsilon^{1+\gamma}} \mathrm{d}\upsilon \, aG_0(\mathrm{d}y)$, for generalized gamma CRM $\widetilde{\mu}$, with parameters: $\tau \geq 0, \gamma \in [0, 1)$ (at least one positive), normalized to r.p.m. $G(\cdot) = \dfrac{\widetilde{\mu}(\cdot)}{\widetilde{\mu}(\mathbb{Y})}$ (Lijoi, Meña, and Prünster 2007)	$\tau = 1, \gamma \to 0$: Dirichlet process $\mathcal{DP}(a, G_0)$ (Ferguson 1973). $\gamma = \frac{1}{2}$: inverse-Gaussian hNRMI (Lijoi, Meña, and Prünster 2005b) $\tau = 0$: σ-stable hNRMI (Kingman 1975)
Generalized Dirichlet process hNRMI, $\mathcal{GD}(\gamma, a, G_0)$	$\dfrac{1-\exp(-\gamma\upsilon)}{1-\exp(-\upsilon)} \dfrac{\exp(-\upsilon)}{\upsilon} \mathrm{d}\upsilon \, aG_0(\mathrm{d}y)$, of a superposed gamma CRM $\widetilde{\mu}$ with parameter $\gamma > 0$, normalized to r.p.m. $G(\cdot) = \dfrac{\widetilde{\mu}(\cdot)}{\widetilde{\mu}(\mathbb{Y})}$ (e.g., Lijoi, Meña, and Prünster 2005a).	$\gamma = 1$, CRM unnormalized: Gamma process. $\gamma = 1$, CRM normalized: Dirichlet process $\mathcal{DP}(a, G_0)$
Stable (3-parameter) Beta process, $\mathcal{SB}(\varsigma, c, a, G_0)$	$\dfrac{\Gamma(c+1)\upsilon^{-\varsigma-1}(1-\upsilon)^{c+\varsigma-1}}{\Gamma(1-\varsigma)\Gamma(c+\varsigma)} \mathrm{d}\upsilon \, aG_0(\mathrm{d}y)$, discount parameter $\varsigma \in [0, 1)$, concentration parameter $c > -\varsigma$ (Teh and Gorür 2009).	$\varsigma = 0$: beta CRM (Hjort 1990) $c = 1 - \varsigma$: stable CRM only with jumps ≤ 1.
Poisson –Dirichlet (stick-breaking) process, $\mathcal{PD}(\ddot{a}, b, G_0)$	Poisson–Kingman $\mathrm{CRM}(\nu)$ (Pitman 2003) represented by the stick-breaking r.p.m. $G(\cdot) = \sum_{j=1}^{\infty} \omega_j(\boldsymbol{\psi}) \delta_{\boldsymbol{\theta}_j}(\cdot)$, $\omega_j(\boldsymbol{\psi}) = \phi_j \prod_{\ell=1}^{j=1} (1 - \phi_\ell)$, $\phi_j \overset{\text{ind}}{\sim} \mathrm{Beta}(1 - \ddot{a}, b + j\ddot{a})$, (parameters $0 \leq \ddot{a} < 1, b > -\ddot{a}$) $\{\boldsymbol{\theta}_j\}_{j=1}^{\infty} \overset{\text{iid}}{\sim} G_0$ (e.g., Ishwaran and James 2001).	$\ddot{a} = 0$: Dirichlet process $\mathcal{DP}(b, G_0)$ (Ferguson 1973) with weights (as $K \to \infty$) $(\omega_1, \ldots, \omega_K)$ $\sim \mathrm{Dirichlet}_K(b/K, \ldots, b/K)$ (Neal 2000). $\phi_j = \phi$, and $\ddot{a} = 0$: a geometric weights process (Meña 2013).

called a homogeneous NRMI (hNRMI; Regazzini, Lijoi, and Prünster 2003), a special type of Poisson–Kingman model (Pitman 2003), if it is defined by mixture weights of the form $\omega_j(\boldsymbol{\psi}) = \widetilde{\mu}(\cdot)/\widetilde{\mu}(\mathbb{Y}) = \mathcal{J}_j / \sum_{\ell=1}^{\infty}\mathcal{J}_\ell$, for $j = 1, 2, \ldots$, where $\boldsymbol{\psi} = (\mathcal{J}_j)_{j=1}^{\infty}$; and is called a stick-breaking prior (Ishwaran and James 2001) if it has mixture weights of the form $\omega_j(\boldsymbol{\psi}) = \phi_j \prod_{\ell=1}^{j-1}(1-\phi_\ell)$, with $\phi_j \overset{\text{ind}}{\sim} \text{Beta}(a_j, b_j)$, for $j = 1, 2, \ldots$, where $\boldsymbol{\psi} = (\phi_j)_{j=1}^{\infty}$. An example of a species sampling BNP prior which models covariate ($\mathbf{x}$) dependence is defined by probit regression mixture weights $\omega_j(\boldsymbol{\psi}) = \Phi(\{j - \mathbf{x}^{\intercal}\boldsymbol{\beta}\}/\sigma(\mathbf{x})) - \Phi(\{j - \mathbf{x}^{\intercal}\boldsymbol{\beta} - 1\}/\sigma(\mathbf{x}))$, $j = 0, \pm 1, \pm 2, \ldots$, with Normal(0,1) cdf $\Phi(\cdot)$ (Karabatsos and Walker 2012).

2.2. Truncating and Sampling CRMs (BNP Priors)

Any intractable infinite-dimensional CRM $\widetilde{\mu}(\cdot) = \sum_{j=1}^{\infty}\mathcal{J}_j\delta_{\boldsymbol{\theta}_j}(\cdot)$ (or its r.p.m., $G(\cdot) = \sum_{j=1}^{\infty}\omega_j(\boldsymbol{\psi})\delta_{\boldsymbol{\theta}_j}(\cdot)$) can be ϵ-approximated by a tractable finite-dimensional CRM $\widetilde{\mu}_K = \sum_{j=1}^{K}\mathcal{J}_j\delta_{\boldsymbol{\theta}_j}(\cdot)$ (or its r.p.m. $G_K(\cdot) = \sum_{j=1}^{K}\omega_j(\boldsymbol{\psi}_K)\delta_{\boldsymbol{\theta}_j}(\cdot)$), for a suitable fixed truncation level $K < \infty$. We use either of two truncation methods, using $\epsilon = 0.001$, which are further described in Appendices A1 and A2 in the supplementary materials.

The first truncation method uses the compound Poisson process approximation of the given CRM $\widetilde{\mu}$ (BNP prior), which excludes small jump sizes less than ϵ (e.g., Argiento, Bianchini, and Guglielmi 2016). The truncation level K is set as the $1-\epsilon$ Poisson quantile. Then the Ferguson and Klass (1972) algorithm can be used to generate samples of (decreasing) truncated mixture weights $\omega_j(\boldsymbol{\psi}_K) = \mathcal{J}_j/\sum_{\ell=1}^{K}\mathcal{J}_\ell$ from the BNP prior (where $\boldsymbol{\psi}_K = (\mathcal{J}_j)_{j=1}^{K}$).

The second truncation method, which relies on the availability of a stick-breaking representation of the given BNP (CRM) prior, selects the truncation level as the smallest positive integer $K < \infty$ which produces a prior expected tail mass for the r.p.m. mixture weights that is less than ϵ (Ishwaran and James 2001, p. 165). Then, K renormalized mixture weights can be obtained by

$$\omega_j(\boldsymbol{\psi}_K) = \left\{\phi_j\prod_{\ell=1}^{j-1}(1-\phi_\ell)\right\} \Bigg/ \left\{\sum_{m=1}^{K}\phi_m\prod_{\ell=1}^{m-1}(1-\phi_\ell)\right\}, \quad \text{for } j = 1, \ldots, K, \tag{2.4}$$

with $\phi_j \overset{\text{ind}}{\sim} \text{Beta}(1-\ddot{a}, b+j\ddot{a})$ (where $\boldsymbol{\psi}_K = (\phi_j)_{j=1}^{K}$). These weights are stochastically decreasing with j, and can be sampled using ordinary methods.

2.3. General BNP Mixture Model and Posterior Distribution

For a given dataset $\mathbf{Y}_n = \{\mathbf{y}_i\}_{i=1}^{n}$, with data points $\mathbf{y}_i = (y_{i1}, \ldots, y_{ip})$ for $i = 1, \ldots, n$, and a fixed dimension $p \geq 1$, a standard BNP mixture model admits the general form:

$$f_G(\mathbf{y}) = \int f(\mathbf{y} \mid \boldsymbol{\theta})\mathrm{d}G(\boldsymbol{\theta}) = \sum_{j=1}^{\infty} f(\mathbf{y} \mid \boldsymbol{\theta}_j)\omega_j(\boldsymbol{\psi}), \tag{2.5a}$$

$$\boldsymbol{\vartheta} = \{\boldsymbol{\theta}_j\}_{j=1}^{\infty} \overset{\text{iid}}{\sim} G_0(\cdot \mid \boldsymbol{\psi}), \quad \boldsymbol{\psi} \sim \Pi(\boldsymbol{\psi} \mid \boldsymbol{\lambda}), \quad \boldsymbol{\lambda} \sim \Pi(\boldsymbol{\lambda}), \tag{2.5b}$$

where the $f(\mathbf{y} \mid \boldsymbol{\theta}_j)$ are chosen component density functions, $G(\cdot)$ is a r.p.m. with parameters $(\boldsymbol{\vartheta}, \boldsymbol{\psi})$ assigned a BNP (CRM) prior, and the mixture weights $\omega_j(\boldsymbol{\psi})$ sum to 1.

Often in practice, the component probability density functions $f(\mathbf{y} \mid \boldsymbol{\theta}_j)$ are chosen from the exponential family, and are assigned a conjugate independent prior $G_0(\cdot)$ (i.e., $G_0(\cdot \mid \boldsymbol{\psi}) = G_0(\cdot)$, with $\Pi(\boldsymbol{\psi} \mid \boldsymbol{\lambda}) = \Pi(\boldsymbol{\psi})$). A prominent example is the BNP multivariate Gaussian mixture model, with components defined by multivariate (p-variate) normal pdfs $f(\mathbf{y} \mid \boldsymbol{\theta}_j) = \mathrm{n}_p(\mathbf{y} \mid \boldsymbol{\mu}_j, \Sigma_j)$, for $j = 1, 2, \ldots$, assigned a conjugate, normal Wishart (NW) prior distribution:

$$\begin{aligned} G_0(\boldsymbol{\mu}, \Sigma) &= \mathrm{NW}_p(\boldsymbol{\mu}, \Sigma^{-1} \mid \boldsymbol{\mu}_{\boldsymbol{\mu}}, n_0, a_\Sigma, \boldsymbol{\beta}_\Sigma) \\ &= \mathrm{N}_p(\boldsymbol{\mu} \mid \boldsymbol{\mu}_{\boldsymbol{\mu}}, (1/n_0)\Sigma)\mathrm{Wi}_p(\Sigma^{-1} \mid a_\Sigma, \boldsymbol{\beta}_\Sigma), \end{aligned} \tag{2.6}$$

with mean $\boldsymbol{\mu}_{\boldsymbol{\mu}}$, prior sample size n_0, degrees of freedom a_Σ, and scale matrix $\boldsymbol{\beta}_\Sigma$.

The BNP multivariate Gaussian mixture model not only provides density and clustering estimation. It is also useful for nonparametric regression or functional data analysis, via posterior inferences of the conditional density $f_G(\mathbf{y} \mid \mathbf{x}) = f_G(\mathbf{x}, \mathbf{y})/f_G(\mathbf{x})$, where $f_G(\mathbf{x}, \mathbf{y})$ is modeled by a DP multivariate Gaussian mixture (e.g., Müller, Erkanli, and West 1996; Rodríguez, Dunson, and Gelfand 2009), for example.

Often in practice, the general BNP mixture model (2.5) is estimated from data, based on an observed likelihood function with fixed truncation level $K < \infty$, given by

$$f_G(\mathbf{Y}_n \mid \boldsymbol{\vartheta}_K, \boldsymbol{\psi}_K) = \prod_{i=1}^{n}\sum_{j=1}^{K} f(\mathbf{y}_i \mid \boldsymbol{\theta}_j)\omega_j(\boldsymbol{\psi}_K). \tag{2.7}$$

This corresponds to the complete likelihood function (Symons 1981):

$$\begin{aligned} f_G(\mathbf{Y}_n, \mathbf{d}_n \mid \boldsymbol{\vartheta}_K, \boldsymbol{\psi}_K) &= \prod_{i=1}^{n} f(\mathbf{y}_i \mid \boldsymbol{\theta}_{d_i})\omega_{d_i}(\boldsymbol{\psi}_K) \\ &= \prod_{j=1}^{K}\left\{\prod_{i:d_i=j} f(\mathbf{y}_i \mid \boldsymbol{\theta}_j)\right\}\prod_{j=1}^{K}\omega_j^{n_j}(\boldsymbol{\psi}_K), \end{aligned} \tag{2.8}$$

where $d_i = j$ if observation $\mathbf{y}_i$ arises from cluster (component) j, with conditional probability:

$$\Pr(d_i = j \mid \boldsymbol{\vartheta}_K, \boldsymbol{\psi}_K) = \frac{f(\mathbf{y}_i \mid \boldsymbol{\theta}_j)\omega_j(\boldsymbol{\psi}_K)}{\sum_{\ell=1}^{K} f(\mathbf{y}_i \mid \boldsymbol{\theta}_\ell)\omega_\ell(\boldsymbol{\psi}_K)}, \quad \text{for } i = 1, \ldots, n; \tag{2.9}$$

and cluster group frequencies are given by $n_j = \#(d_i = j) = \sum_{i=1}^{n}\mathbf{1}(d_i = j)$, for $j = 1, \ldots, K$. Summing over the independent $(d_i)_{i=1}^{n}$ in (2.8) returns the original likelihood (2.7).

A complete dataset $(\mathbf{Y}_n, \mathbf{d}_n)$ updates the BNP prior distribution to a posterior distribution:

$$\Pi(\boldsymbol{\vartheta}_K, \boldsymbol{\psi}_K \mid \mathbf{Y}_n, \mathbf{d}_n) \propto f_G(\mathbf{Y}_n, \mathbf{d}_n \mid \boldsymbol{\vartheta}_K, \boldsymbol{\psi}_K)\prod_{j=1}^{K} G_0(\boldsymbol{\theta}_j)\Pi(\boldsymbol{\psi}_K)$$
$$= \prod_{j=1}^{K}\Pi(\boldsymbol{\theta}_j \mid \{\mathbf{y}_i : d_i = j\})\Pi(\boldsymbol{\psi}_K \mid \mathbf{d}_n). \tag{2.10a}$$

The corresponding, posterior predictive distribution is given by

$$f_n(\mathbf{y}_{n+1}) = \int\cdots\int \sum\nolimits_{j=1}^{K} f(\mathbf{y}_{n+1} \mid \boldsymbol{\theta}_j)\omega_j(\boldsymbol{\psi}_K)\, \mathrm{d}\Pi(\boldsymbol{\vartheta}_K, \boldsymbol{\psi}_K \mid \mathbf{Y}_n, \mathbf{d}_n), \tag{2.11a}$$
$$= \sum_{j=1}^{K}\int f(\mathbf{y}_{n+1} \mid \boldsymbol{\theta}_j)\mathrm{d}\Pi(\boldsymbol{\theta}_j \mid \{\mathbf{y}_i : d_i = j\}) \tag{2.11b}$$
$$\int \omega_j(\boldsymbol{\psi}_K)\prod_{j=1}^{K}\omega_j^{n_j}(\boldsymbol{\psi}_K)\mathrm{d}\Pi(\boldsymbol{\psi}_K \mid \mathbf{d}_n),$$
$$= \sum_{j=1}^{K} f_{n_j}(\mathbf{y}_{n+1})\omega_{n_j}, \tag{2.11c}$$

based on posterior predictive component densities $f_{n_j}(\mathbf{y}_{n+1})$ and mixture weights $\omega_{n_j} = \Pr(d_{n+1} = j \mid \mathbf{d}_n)$. The posterior probability function of the cluster assignments given by

$$\Pr(d_i = j \mid \mathbf{y}_i) = \frac{f_{n_j}(\mathbf{y}_i)\omega_{n_j}}{\sum_{\ell=1}^{K} f_{n_\ell}(\mathbf{y}_i)\omega_{n_\ell}}, \text{ for } j = 1, \ldots, K. \tag{2.12}$$

2.4. The BNP-CAEM Algorithm

The BNP-CAEM algorithm, for posterior predictive density and MAP clustering estimation of the general BNP mixture model, is applicable to virtually any chosen BNP prior. This is because the algorithm mainly relies on the ability to generate proposal samples of the (truncated) mixture weights from the chosen prior, which are used repeatedly over iterations to provide corresponding importance sampling estimates of the mixture weights $\{\omega_{n_j}\}_{j=1}^{K}$.

The default BNP-CAEM algorithm, shown as Algorithm 1, is applicable to any general BNP mixture model which makes the usual assumption that G_0 is a conjugate prior for the component parameters $\boldsymbol{\vartheta}_K$, with prior independence between the component and mixture weight parameters.

Algorithm Step 0, the initializing step, generates a large number R of proposal samples of the mixture weight parameters. This step can also be done based on a prior $\Pi(\boldsymbol{\psi}_K \mid \boldsymbol{\lambda})$ assigned a hyper-prior distribution $\boldsymbol{\lambda} \sim \Pi(\boldsymbol{\lambda})$, by drawing proposal samples $\boldsymbol{\lambda}_r \overset{\text{iid}}{\sim} \Pi(\boldsymbol{\lambda})$ and $\boldsymbol{\psi}_{K,r} \mid \boldsymbol{\lambda}_r \overset{\text{iid}}{\sim} \Pi(\boldsymbol{\psi}_K \mid \boldsymbol{\lambda}_r)$ for $r = 1, \ldots, R$. Also, Step 0 implements a Bayesian K-component extension of a quasi-clustering technique (Woodward et al. 1984, p. 592) to set the starting values of the clustering and associated statistics. See Appendix A3 in the supplementary materials for more details.

Algorithm 1. BNP classification annealing EM algorithm.

```
Step 0  Draw a large number $R$ of prior samples $\{\boldsymbol{\psi}_{K,r}\}_{r=1}^{R} \overset{\text{iid}}{\sim} \Pi(\boldsymbol{\psi})$.
        Set the starting values $s = 0$, $\mathbf{d}_n^{(0)} = (d_i^{(0)})_{i=1}^{n}$,
        $n_j^{(0)} = \sum_{i=1}^{n}\mathbf{1}(d_i^{(0)} = j)$, $f_{n_j}^{(0)}$, and $\omega_{n_j}^{(0)}$, for $j = 1, \ldots, K$.
        Set temperature schedule $T_s = \{\max(0.97^{s-I}, 0.01)\}^{\mathbf{1}(s>I)}$,
        $s = 1, \ldots, S$.
Step 1  Classification step (C-step). Set $s = s + 1$,
        draw $d_i^{(s)} \sim \Pr(d_i = j \mid \mathbf{y}_i) \propto \{f_{n_j}^{(s-1)}(\mathbf{y}_i)\omega_{n_j}^{(s-1)}\}^{1/T_s}$, $i = 1, \ldots, n$,
        and update $n_j^{(s)} = \sum_{i=1}^{n}\mathbf{1}(d_i^{(s)} = j)$ for $j = 1, \ldots, K$.
Step 2  Maximization step (M-step). For $j = 1, \ldots, K$, calculate:
        $f_{n_j}^{(s)}$, based on the $n_j^{(s)}$ data points from $\{\mathbf{y}_i\}_{i=1}^{n}$ for which $d_i^{(s)} = j$,
        $\omega_{n_j}^{(s)} = \sum_{r=1}^{R}\omega_j(\boldsymbol{\psi}_{K,r})\frac{\prod_{j=1}^{K}\omega_j^{n_j^{(s)}}(\boldsymbol{\psi}_{K,r})}{\sum_{q=1}^{R}\prod_{j=1}^{K}\omega_j^{n_j^{(s)}}(\boldsymbol{\psi}_{K,q})}$
        $= \sum_{r=1}^{R}\omega_j(\boldsymbol{\psi}_{K,r})\overline{w}^{(s)}(\boldsymbol{\psi}_{K,r})$,
        and if $s \leq I$, calculate $f_n^{(s)}$ over a fixed grid of $\mathbf{y}_{n+1}$ values:
        $f_n^{(s)}(\mathbf{y}_{n+1}) = \frac{1}{s}\left[\sum_{j=1}^{K} f_{n_j}^{(s)}(\mathbf{y}_{n+1})\omega_{n_j}^{(s)} + (s-1)f_n^{(s-1)}(\mathbf{y}_{n+1})\right]$.
Step 3  Repeat Steps 1 and 2 for iterations $s = 1, 2, \ldots, S$,
        until two successive iterations yield no change in the clustering $\mathbf{d}_n$,
        or until the last iteration ($s = S$) is reached.
        Then $\widehat{f}_n = f_n^{(I)}$ is the posterior predictive density estimate,
        and $\mathbf{d}_n^{(s)} = \widehat{\mathbf{d}}_n$ is the MAP estimate of the clustering.
```

Further, Step 0 sets the decreasing temperature schedule $T_1 = \cdots = T_I = 1 > T_{I+1} > \cdots > T_S$, where the temperature is 1 for I initial iterations, followed by decreasing temperatures in later iterations. Since good performance of simulated annealing requires a slow convergence rate of the sequence T_s to 0 (Van Laarhoven and Aarts 1987), a temperature schedule can be chosen as $T_s = \{\max(h^{s-I}, 0.01)\}^{\mathbf{1}(s>I)}$ for some $0.9 \leq h < 1$ (the $\max(\cdot, 0.01)$ function ensures numerical stability of the algorithm). According to numerical experiments in Section 3, reasonable results were obtained by using $h = 0.97$, $R = 20{,}000$ proposal samples, $I = 500$ initial SEM iterations, and $S = 700$ total iterations, after considering several trial values of h within $0.9 \leq h < 1$. The choice $h = 0.97$ also provided reasonable results for the CAEM MLE algorithm (Celeux and Govaert 1992, sec. 5). Because BNP-CAEM is a fast search-algorithm, the user can quickly rerun the algorithm over different values $0.9 \leq h < 1$, and then select the h that attains the highest complete likelihood (2.8).

Step 1 (C-step) of the BNP-CAEM algorithm draws a new sample of the clustering, conditional on the previous posterior predictive component density and mixture weights, and then updates the cluster group frequencies. Step 2 (M-step) of the algorithm performs closed-form updates of the component posterior predictive densities $\{f_{n_j}\}_{j=1}^{K}$, and directly updates the importance sampling estimates of the mixture weights $\{\omega_{n_j}\}_{j=1}^{K}$. The closed-form estimation is possible by virtue of the conjugate NW prior G_0 for the component parameters of the BNP Gaussian mixture model (Section 2.3), which implies that each f_{n_j} is a student density. See Appendix A4 in the supplementary materials for more details on computations, including how they can accommodate weights for the n data points $\mathbf{Y}_n = \{\mathbf{y}_i\}_{i=1}^{n}$ (resp.). For the mixture weight estimation, each importance weight is proportional to multinomial density functions (later normalized), according to the last term in the complete model likelihood (2.8).

Steps 1 and 2 of Algorithm 1 are repeated for S iterations (see Step 3). Step 2 (M-step) directly updates the estimate of the posterior predictive density f_n (posterior mean of f) only while the temperature is $T = 1$ during the I initial iterations. During these initial iterations, the BNP-CAEM algorithm acts as a stochastic EM algorithm and eventually produces a sequence of ergodic time-homogeneous Markov chain of posterior predictive density estimates, which converges to a random density estimate variable which has the stationary distribution of the Markov chain as the number of iterations grows, and has an asymptotic normal distribution centered on the true density of the data when the data sample size is large (as derived from Nielsen 2000). Thus, the posterior predictive density estimates produced over these I initial BNP-CAEM iterations can be averaged to produce a final density estimate in Step 3.

After iteration I, the temperature is gradually decreased toward 0, so that the amount of randomness in the simulations decreases with the iterations, ending up with an approximate MAP estimate ($\widehat{\mathbf{d}}_n$) of the clustering of the n data points $\mathbf{Y}_n = \{\mathbf{y}_i\}_{i=1}^n$.

Algorithm 1 can be extended to handle a non-conjugate prior G_0, with possible dependence between the component and mixture weight parameters, albeit with some added computational cost. This extension involves generating proposal samples of the mixture component parameters in Step 0, and then adding a corresponding importance sampling estimator for the component posterior predictive densities in Step 2. See Appendix A5 in the supplementary materials for more details.

3. Illustrations

Section 3.1 illustrates the BNP-CAEM algorithm for various BNP Gaussian mixture models, through density estimation and clustering analysis of various benchmark simulated datasets and real datasets, univariate and multivariate. Each mixture model is defined by one of six types of BNP priors for the mixture distribution, namely, the $\mathcal{DP}(1, G_0)$, $\mathcal{PD}(1/4, 1, G_0)$, $\mathcal{NGG}(1, 1/4, 1, G_0)$, $\mathcal{NGG}(1, 1/2, 1, G_0)$, $\mathcal{GD}(1/2, 1, G_0)$ processes, and a normalized stable-beta $\mathcal{NSB}(1/2, 1, 1, G_0)$ process; with G_0 the NW prior distribution (2.6) with parameters $\boldsymbol{\mu}_{\boldsymbol{\mu}} = \frac{1}{n}\sum_{i=1}^n \mathbf{y}_i$, $n_0 = 1$, $a_\Sigma = \dim(\mathbf{y}) + 1$, and $\boldsymbol{\beta}_\Sigma = \text{diag}([b_0 \cdot (\text{range}\{y_{i,1}\}_{i=1}^n, \ldots, \text{range}\{y_{i,p}\}_{i=1}^n)]^{-1})$, with $b_0 = 3$ for univariate data, and $b_0 = 50$ for multivariate data (see Richardson and Green 1997; Ishwaran and James 2001; Lijoi, Meña, and Prünster 2005a; Arbel and Prünster 2017). Section 3.2 illustrates the $\mathcal{GD}(1/2, 1, G_0)$ BNP bivariate Gaussian mixture models for the nonparametric regression analysis of simulated and real datasets.

Sections 3.1 and 3.2 mainly serve to illustrate the BNP-CAEM algorithm and its easy applicability to a wide range of BNP priors, by simple changes to algorithm Step 0 (prior $\Pi(\boldsymbol{\psi})$ selection and sampling), and possible simple changes to algorithm Step 2 (f_{n_j} calculation depends on whether G_0 is conjugate). Given the model-based nature of the above BNP mixture models, the density and clustering estimates will depend on the choice of BNP prior (see Lijoi, Meña, and Prünster 2007). However, users can rerun the fast-search BNP-CAEM algorithm on the same dataset for different choices of prior, to perform sensitivity analyses or model averaging.

Section 3.1 also compares BNP-CAEM against three other mixture model estimation algorithms using available licensed MATLAB code (Chen 2016; Eisenstein 2012; Chen 2019), namely:
(1) the collapsed Gibbs sampling algorithm (Neal 2000, Algorithm 3) for the $\mathcal{DP}(1, G_0)$ process Gaussian mixture model, with Rao-Blackwellized density estimator (Gelfand and Mukhopadhyay 1995) and least-squares clustering estimator (Dahl 2006). (Algorithm starting values are given by one iteration of an online collapsed algorithm run on a random permutation of the data points);
(2) the VB algorithm for clustering and density estimation with the $\mathcal{DP}(1, G_0)$ process Gaussian mixture model (Blei and Jordan 2006, and using eq. (23)). (Estimates are from the VB algorithm run with the highest observed likelihood, over 10 runs using random starting clustering probabilities);
(3) the EM algorithm (EM-MLE FMM) for maximum likelihood density and clustering estimation of the K-component finite-mixture model, with K found by BIC optimization (e.g., Stahl and Sallis 2012). (Algorithm starting values are given by a K-group hierarchical clustering if $n < 10^4$, or by K quantile groups of univariate data or of Mahalanobis depths of multivariate data).

All four algorithms are compared in terms of density estimation accuracy, clustering accuracy, and computation time. For each algorithm, density estimation accuracy is measured by the Kullback–Leibler divergence, $\text{KL}(f, \widehat{f}_n) = \mathbb{E}_f[\log\{f(\boldsymbol{Y})/\widehat{f}_n(\boldsymbol{Y})\}] = \int_{\mathbb{R}^p} \log\{f(\mathbf{y})/\widehat{f}_n(\mathbf{y})\} f(\mathbf{y}) d\mathbf{y}$, where f is the true data-simulating density and $\widehat{f}_n$ is the algorithm's density estimate (see Appendix A6 in the supplementary materials for computational details). Clustering accuracy is measured by the Rand index (and adjusted Rand index) of proportional agreement between the algorithm's clustering estimate and the true data-simulated (or criterion) clustering (Gates and Ahn 2017). The Rand index has range $[0, 1]$. The adjusted Rand index takes values in $(-\infty, 1]$, and equals 0 if there is perfect agreement with random chance, according to a uniform distribution over all clusterings of n elements. All results reported in Sections 3.1 and 3.2 were obtained from a modest laptop computer with Intel Core i7-8565U 2GHz processor and 16GB RAM.

3.1. Univariate and Multivariate Examples

Thirteen benchmark datasets, including ten simulated and three real datasets, were analyzed by the BNP-CAEM, Collapsed Gibbs, VB, and EM-MLE FMM algorithms for Gaussian mixture model estimation.

Six univariate datasets were each simulated as n iid samples from the Gaussian mixture:

$$f(y) = 0.3 \times \text{n}(y \mid -2, 0.4) + .5 \times \text{n}(y \mid 0, 0.3) + 0.2 \times \text{n}(y \mid 2.5, 0.3), \quad (3.1)$$

(Wang and Dunson 2011, p. 205), using $n = 80$, 200, 2000, 20,000, 200,000, or 2,000,000. Similarly, four multivariate datasets were each simulated as n iid samples from the mixture:

$$f(\mathbf{y}) = 0.3 \times \mathrm{n}_p(\mathbf{y} \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + .5 \times \mathrm{n}_p(\mathbf{y} \mid \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2) + .2 \times \mathrm{n}_p(\mathbf{y} \mid \boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3), \tag{3.2}$$

with $n = 140$ or 200,000 and dimension $p = 2$ or 10. For $p = 2$, the mean vectors are $\boldsymbol{\mu}_1 = (3^{3/2}/2, 0)^\intercal$, $\boldsymbol{\mu}_2 = (-3^{3/2}/2, 3)^\intercal$, and $\boldsymbol{\mu}_3 = (-3^{3/2}/2, -3)^\intercal$. For $p = 10$, $\boldsymbol{\mu}_1 = (3^{3/2}/2, 0, 1, -1, \ldots, 1, -1)^\intercal$, $\boldsymbol{\mu}_2 = (-3^{3/2}/2, 3, -1, 1, \ldots, -1, 1)^\intercal$, and $\boldsymbol{\mu}_3 = (-3^{3/2}/2, -3, -1, 1, \ldots, -1, 1)^\intercal$. The covariance matrices are $\boldsymbol{\Sigma}_1 = (0.4^{|k-m|})_{p\times p}$, $\boldsymbol{\Sigma}_2 = (0.3^{|k-m|})$, and $\boldsymbol{\Sigma}_3 = \mathbf{I}_p$, for $k, m = 1, \ldots, p$. In general, each data point $\mathbf{y}_i$ was simulated by drawing $\mathbf{y}_i \sim \mathrm{n}(\mathbf{y} \mid \boldsymbol{\mu}_{d_i}, \boldsymbol{\Sigma}_{d_i})$, after drawing a cluster membership $d_i \sim \text{Discrete}(0.3, 0.5, 0.2)$.

The three real benchmark datasets include the univariate `Galaxy` data on the velocities of $n = 82$ galaxies in km/sec (Roeder 1990), and `Enzyme` data (Richardson and Green 1997) on the metabolic activity of carcinogenic substances for $n = 245$ persons (shown in Figure 1). The multivariate `Diabetes` dataset (Banfield and Raftery 1993) contains measurements of $n = 145$ subjects on glucose area, insulin area, and steady-state plasma glucose response (sspg). Each subject was classified as either chemical diabetes (36 cases; with medians (476.5, 251.5, 223) for glucose, insulin, sspg), overt diabetes (33 cases; medians (972, 83, 320)), or normal (76 cases; medians (353, 157, 105)). Mixture model analysis was performed on the z-scores of each variable (having mean 0 and variance 1), using the subject classifications as the true clustering.

The results in Tables 2–4 and Figure 1 show that for the BNP-CAEM algorithm and the six BNP mixture models, the Kullback–Leibler divergence approached zero as n increased, the density estimates fit the real univariate data well, and that BNP-CAEM was competitive with other algorithms in terms of density estimation accuracy (Kullback–Leibler divergence) and clustering accuracy (Rand and adjusted Rand indices). BNP-CAEM was always faster than Collapsed Gibbs, often faster than EM-MLE for $n \geq 2000$, and always faster than EM-MLE for $n \geq 200{,}000$. The Collapsed Gibbs algorithm was infeasible $n \geq 20{,}000$. But such scenarios are exactly those which motivate the development of fast-search algorithms, such as BNP-CAEM and VB.

BNP-CAEM, over all BNP mixture models, simulated datasets, and the real `Diabetes` dataset, attained Rand indices with median 0.74 and range [0.62, 1], and adjusted Rand indices with median 0.32 and range [0, 1]. BNP-CAEM attained perfect (or near perfect) Rand indices for the ($n = 140, p = 2$) simulated multivariate dataset, with the $\mathcal{PD}(1/4, 1, G_0)$, $\mathcal{GD}(1/2, 1, G_0)$, and $\mathcal{NSB}(1/2, 1, 1, G_0)$ Gaussian mixture models. For the real `Diabetes` data analysis, BNP-CAEM, with the $\mathcal{DP}(1, G_0)$, $\mathcal{PD}(1/4, 1, G_0)$, $\mathcal{GD}(1/2, 1, G_0)$, and $\mathcal{NSB}(1/2, 1, 1, G_0)$ Gaussian mixture models, had better clustering accuracy than EM-MLE. With the $\mathcal{DP}(1, G_0)$ normal mixture model, BNP-CAEM always had better clustering accuracy than the VB and Collapsed Gibbs algorithms.

In conclusion, among the BNP-CAEM and VB fast-search algorithms which aim to provide faster and approximate alternatives to MCMC and EM algorithms, BNP-CAEM was rather competitive in density estimation accuracy and better in clustering accuracy, and is more easily applicable to different BNP

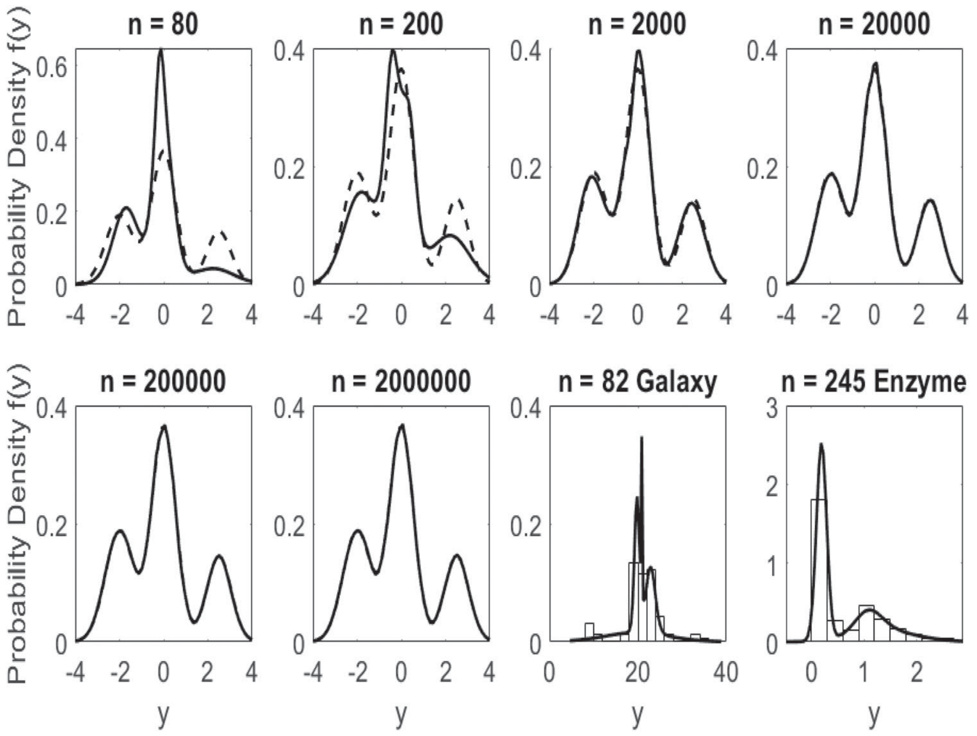

Figure 1. First six panels (from left to right): Density estimate (solid line) of the $\mathcal{GD}(1/2, G_0)$ mixture model, and true density (dashed line), for the 6 simulated datasets with sample size n, respectively. Two lower right panels: Density estimate (solid line) of the $\mathcal{PD}(1/4, 1, G_0)$ normal mixture model obtained from the `Galaxy` data (histogram), and density estimate of the $\mathcal{DP}(1, G_0)$ normal mixture model obtained from the `Enzyme` data (histogram).

Table 2. Results of the univariate simulation study, for algorithms and BNP priors.

	BNP-CAEM $\mathcal{DP}(1, G_0)$						BNP-CAEM $\mathcal{PD}(1/4, 1, G_0)$					
Size, $n =$	80	200	2K	20K	200K	2M	80	200	2K	20K	200K	2M
KL	0.09	0.03	0.02	0.02	0.02	0.02	0.10	0.03	0.01	0.01	0.008	0.0004
Rand index	0.76	0.83	0.94	0.92	0.92	0.94	0.89	0.78	0.78	0.76	0.76	0.78
Adj. Rand	0.44	0.56	0.84	0.80	0.78	0.84	0.73	0.44	0.42	0.38	0.37	0.41
# clusters	4	4	3	5	7	5	4	5	7	9	17	22
K (trunc.)	11	11	11	11	11	11	55	55	55	55	55	55
Time (sec)	6	5	8	25	218	2085	20	21	34	115	963	9673
Iterations	535	552	616	700	700	700	529	552	628	700	700	700

	BNP-CAEM $\mathcal{NGG}(1, 1/4, 1, G_0)$						BNP-CAEM $\mathcal{NGG}(1, 1/2, 1, G_0)$					
Data $n =$	80	200	2K	20K	200K	2M	80	200	2K	20K	200K	2M
KL	0.26	0.07	0.008	0.0006	0.0001	0.00007	0.46	0.17	0.02	0.0009	0.0001	0.00004
Rand index	0.68	0.75	0.71	0.66	0.65	0.65	0.68	0.67	0.72	0.68	0.64	0.64
Adj. Rand	0.24	0.35	0.24	0.10	0.09	0.09	0.24	0.16	0.25	0.17	0.05	0.04
# clusters	6	8	22	27	27	27	6	7	22	45	53	53
K (trunc.)	27	27	27	27	27	27	53	53	53	53	53	53
Time (sec)	15	17	22	73	557	5376	27	32	45	146	1088	10424
Iterations	576	579	669	700	700	700	582	579	681	700	700	700

	BNP-CAEM $\mathcal{GD}(1/2, 1, G_0)$						BNP-CAEM $\mathcal{NSB}(1/2, 1, 1, G_0)$					
Data $n =$	80	200	2K	20K	200K	2M	80	200	2K	20K	200K	2M
KL	0.13	0.05	0.004	0.0003	0.0001	0.0001	0.17	0.04	0.02	0.003	0.0001	0.0002
Rand index	0.74	0.78	0.74	0.74	0.74	0.74	0.78	0.83	0.80	0.74	0.72	0.71
Adj. Rand	0.39	0.43	0.32	0.32	0.31	0.31	0.48	0.55	0.47	0.32	0.25	0.24
# clusters	4	6	10	10	10	10	4	4	5	11	38	58
K (trunc.)	10	10	10	10	10	10	59	59	59	59	59	59
Time (sec)	6	6	9	27	214	2076	22	25	44	147	1154	11530
Iterations	543	561	633	700	700	700	538	560	632	700	700	700

	Variational Bayes $\mathcal{DP}(1, G_0)$						Collapsed Gibbs $\mathcal{DP}(1, G_0)$					
Data $n =$	80	200	2K	20K	200K	2M	80	200	2K	20K	200K	2M
KL	0.12	0.03	0.003	0.15	0.15	0.15	0.09	0.04	0.002	na	na	na
Rand index	0.78	0.81	0.79	0.63	0.67	0.80	0.72	0.73	0.87	na	na	na
Adj. Rand	0.48	0.51	0.46	0.04	0.14	0.46	0.33	0.32	0.66	na	na	na
# clusters	4	4	7	2	3	5	8	8	6	na	na	na
K (trunc.)	50	50	50	50	50	50	na	na	na	na	na	na
Time (sec)	2	4	2	3	42	451	172	464	25124	days	days	days
Iterations	707	1220	270	90	90	90	10^4	10^4	10^4	10^4	10^4	10^4

	EM-MLE finite mixture model					
Data $n =$	80	200	2K	20K	200K	2M
KL	0.17	0.01	0.0009	0.0001	0.00004	0.00001
Rand index	0.95	0.94	0.94	0.94	0.94	0.94
Adj. Rand	0.88	0.84	0.84	0.84	0.84	0.84
# clusters	4	4	4	4	4	4
K (trunc.)	4	4	4	4	4	4
Time (sec)	3	7	39	91	1304	13,885
Iterations	4405	8788	15,657	4931	5075	5016

NOTE: VB and EM-MLE iterations are counted over multiple starting values and trial K values, resp. For data sample size n, 2K = 2000, 20K = 20,000, 200K = 200,000, 2M = 2,000,000).

priors, without relying on factorization assumptions which depart from the underlying probabilistic mixture model to gain computational speed.

3.2. *Regression Examples*

We now consider nonparametric regression, via marginal posterior inferences of the conditional (regression) density function $f(y \mid x) = f_G(x, y)/f(x)$, based on a BNP bivariate Gaussian mixture model for the joint density $f_G(x, y)$.

First, for illustrative purposes, we applied a $\mathcal{NGG}(1, 1/4, 1, G_0)$ bivariate Gaussian mixture model to analyze a dataset $\{(x_i, y_i)\}_{i=1}^{n=61}$, simulated by $y_i = 0.2x_i^3 + \varepsilon_i$, for $i = 1, \ldots, 61$, where $x_1 = -3, x_2 = -2.9, \ldots, x_{60} = 2.9, x_{61} = 3$, and $\varepsilon_i \overset{\text{iid}}{\sim} \text{n}(0, 0.25)$. For this model and dataset, the BNP-CAEM algorithm completed in 11 sec, with 519 BNP-CAEM iterations, truncation level $K = 27$, and 17 clusters. Figure 2 shows that the conditional posterior predictive density estimates, $\widehat{f}_n(y \mid x) \approx \widehat{f}_n(x, y)/\widehat{f}_n(x)$, tracked the simulated y_i observations well.

Next, we analyze a large bivariate dataset of 851,450 class sizes and math scores of 170,290 Grade 4 students from 37 countries and 726 schools, obtained in 2011 (*https://timssandpirls.bc.edu/timsspirls2011/international-database.html*). Each student received 5 plausible math scores. Before data analysis, all math scores were rescaled to have mean 0 and variance 1, and class sizes were converted to classSize/10. Figure 3 (left

Table 3. `Galaxy` ($n = 82$) and `Enzyme` ($n = 245$) analysis results by algorithm.

BNP-CAEM	$\mathcal{DP}(1, G_0)$		$\mathcal{PD}(1/4, 1, G_0)$		$\mathcal{NGG}(1, 1/4, 1, G_0)$	
Dataset	`Galaxy`	`Enzyme`	`Galaxy`	`Enzyme`	`Galaxy`	`Enzyme`
# clusters	4	3	4	3	5	6
K (truncation)	11	11	55	55	27	27
Time (sec)	5	5	23	20	12	17
Iterations	528	532	700	556	580	581
BNP-CAEM	$\mathcal{NGG}(1, 1/2, 1, G_0)$		$\mathcal{GD}(1/2, 1, G_0)$		$\mathcal{NSB}(1/2, 1, 1, G_0)$	
Dataset	`Galaxy`	`Enzyme`	`Galaxy`	`Enzyme`	`Galaxy`	`Enzyme`
# clusters	8	7	5	5	4	4
K (truncation)	53	53	10	10	59	59
Time (sec)	23	34	5	6	21	27
Iterations	589	589	539	554	552	556
	VB $\mathcal{DP}(1, G_0)$		CG $\mathcal{DP}(1, G_0)$		EM-MLE FMM	
Dataset	`Galaxy`	`Enzyme`	`Galaxy`	`Enzyme`	`Galaxy`	`Enzyme`
# clusters	4	3	4	4	5	3
K (truncation)	50	50	na	na	5	3
Time (sec)	1	1	155	493	2	18
Iterations	441	351	10^4	10^4	2506	17542

NOTE: VB, EM-MLE iterations are over multiple starting and trial K values, resp.

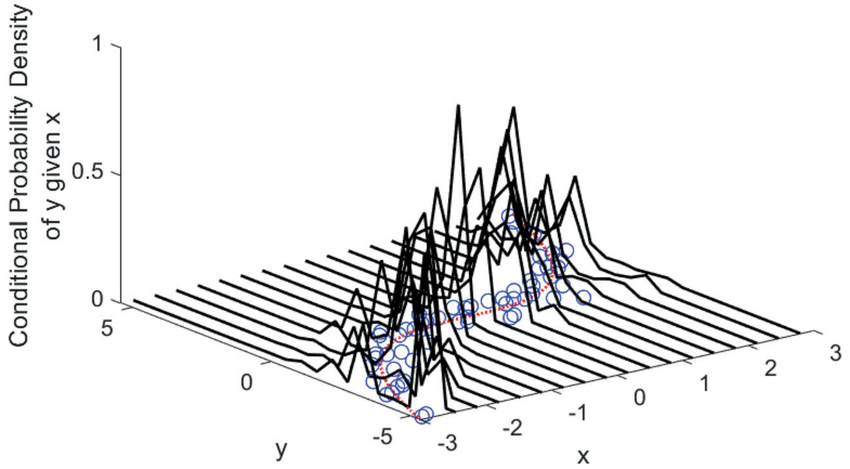

Figure 2. For the simulated data (circular markers) analyzed by the $\mathcal{NGG}(1/4, 1, G_0)$ bivariate normal mixture model, the conditional density estimate of Y given $X = x$ (solid line), and the true mean regression function (dashed line).

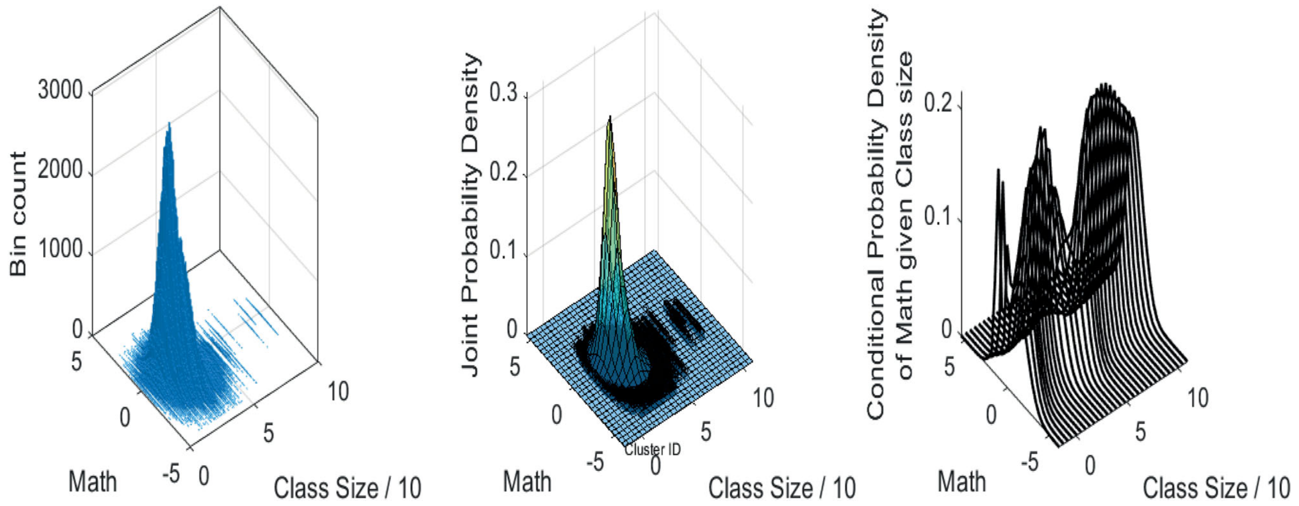

Figure 3. Left panel: Bivariate histogram of the class size and math score data ($n = 851{,}450$). Middle and right panels: Bivariate density estimate, and conditional density estimates of math score given class size, based on fitting the generalized Dirichlet process normal mixture model to the squashed bivariate data.

Table 4. Results of multivariate simulation study, and `Diabetes` analysis ($n = 145, p = 3$), for algorithms and BNP priors.

	BNP-CAEM $\mathcal{DP}(1, G_0)$					BNP-CAEM $\mathcal{PD}(1/4, 1, G_0)$				
$(n, p) =$	140,2	200K,2	140,10	200K,10	145,3	140,2	200K,2	140,10	200K,10	145,3
KL	0.39	0.02	2.73	0.01	na	0.12	0.25	3.52	0.04	na
Rand index	0.85	0.88	0.65	0.94	0.83	1	0.83	0.63	0.85	0.81
Adj. Rand	0.58	0.68	0.08	0.83	0.57	1	0.55	0.01	0.60	0.51
# Clusters	2	10	8	11	4	3	29	17	48	3
K (trunc.)	11	11	11	11	11	55	55	55	55	55
Time (sec)	4	132	6	245	4	17	552	22	941	16
Iterations	501	700	501	700	528	501	700	501	700	517

	BNP-CAEM $\mathcal{NGG}(1, 1/4, 1, G_0)$					BNP-CAEM $\mathcal{NGG}(1, 1/2, 1, G_0)$				
$(n, p) =$	140,2	200K,2	140,10	200K,10	145,3	140,2	200K,2	140,10	200K,10	145,3
KL	0.93	0.004	22.34	0.03	na	1.63	0.01	22.42	0.03	na
Rand index	0.78	0.67	0.63	0.74	0.63	0.68	0.64	0.63	0.64	0.63
Adj. Rand	0.38	0.14	0.01	0.30	0.06	0.12	0.06	0.01	0.05	0.05
# Clusters	18	27	27	27	26	30	53	53	53	40
K (trunc.)	27	27	27	27	27	53	53	53	53	53
Time (sec)	13	286	16	530	15	26	546	32	971	27
Iterations	520	700	501	700	514	513	700	501	700	512

	BNP-CAEM $\mathcal{GD}(1/2, 1, G_0)$					BNP-CAEM $\mathcal{NSB}(1/2, 1, 1, G_0)$				
$(n, p) =$	140,2	200K,2	140,10	200K,10	145,3	140,2	200K,2	140,10	200K,10	145,3
KL	0.30	0.004	5.40	0.02	na	0.37	0.006	5.82	0.07	na
Rand index	0.98	0.73	0.62	0.82	0.73	1	0.69	0.63	0.68	0.81
Adj. Rand	0.94	0.30	0.00	0.53	0.29	1	0.19	0.03	0.15	0.52
# Clusters	4	10	10	10	8	3	59	48	59	5
K (trunc.)	10	10	10	10	10	59	59	59	59	59
Time (sec)	5	121	6	226	6	19	597	32	1080	18
Iterations	511	700	501	700	531	501	700	501	700	522

	Variational Bayes $\mathcal{DP}(1, G_0)$					Collapsed Gibbs $\mathcal{DP}(1, G_0)$				
$(n, p) =$	140,2	200K,2	140,10	200K,10	145,3	140,2	200K,2	140,10	200K,10	145,3
KL	0.20	0.004	6.59	0.003	na	0.57	na	13.70	na	na
Rand index	0.92	0.84	0.62	0.80	0.70	0.71	na	0.63	na	0.83
Adj. Rand	0.79	0.58	0.00	0.46	0.24	0.18	na	0.03	na	0.56
# clusters	5	38	11	36	7	2	na	79	na	4
K (trunc.)	50	50	50	50	50	na	na	na	na	na
Time (sec)	1	215	1	486	1	16	days	2550	days	241
Iterations	165	311	110	223	205	10^4	10^4	10^4	10^4	10^4

	EM-MLE finite mixture model				
$(n, p) =$	140,2	200K,2	140,10	200K,10	145,3
KL	0.05	0.00002	1.40	0.0005	na
Rand index	1	1	0.82	1	0.72
Adj. Rand	1	1	0.53	1	0.27
# clusters	3	3	2	3	4
K (trunc.)	3	3	2	3	4
Time (sec)	1	623	0.24	7952	1
Iterations	590	2104	100	6356	685

NOTE: VB, EM iterations counted over multiple starting values, trial K values, respectively. For data sample size n, 2K = 2000, 20K = 20,000, 200K = 200,000, 2M = 2,000,000.

panel) presents the data in a bivariate histogram of class size (X) and math score (Y), using the Freedman–Diaconis rule for outlier-robust bin sizes for the two dimensions (given by $\mathrm{IQR}(\{x_i\}_{i=1}^{n})2n^{-1/4}$ and $\mathrm{IQR}(\{y_i\}_{i=1}^{n})2n^{-1/4}$).

Given the very large sample size, we instead analyzed a smaller, squashed version of the dataset. The squashed dataset has $n = 3622$ pseudo data points, such that for each of the 3622 non-empty bins of the bivariate histogram (Figure 3), the pseudo data point is represented by the average class sizes and math scores within the bin, and assigned an observation weight equal to the bin frequency. Then the BNP-CAEM algorithm was used to fit the $\mathcal{GD}(1/2, 1, G_0)$ bivariate Gaussian mixture model to the 3622 pseudo-data points and their respective observation (frequency) weights. This histogram-based data squashing is done in the same spirit as a previous method (Pennell and Dunson 2006), which instead squashed data by using a K-means type algorithm, and then analyzed the squashed data using a $\mathcal{DP}$ mixture model.

The BNP-CAEM, $\mathcal{GD}(1/2, 1, G_0)$ bivariate Gaussian mixture analysis of the squashed data completed in 9.5 sec, yielding 638 BNP-CAEM iterations, truncation level $K = 10$, and 8 clusters. Figure 3 shows the posterior predictive bivariate density estimate $\widehat{f}_n(x, y)$, and conditional density estimates, $\widehat{f}_n(y \mid x) \approx \widehat{f}_n(x, y)/\widehat{f}_n(x)$, of math score ($y$) and class sizes ($x$).

4. Conclusions

We introduced, described, and illustrated the BNP-CAEM algorithm, a new fast search algorithm for performing the common inferential tasks of posterior predictive density estimation, and approximate MAP clustering estimation, under general BNP mixture models. The new algorithm is a Bayesian version of the original CAEM algorithm (Celeux and Govaert 1992), combined with importance sampling methods. The BNP-CAEM algorithm is easy to construct and apply for any BNP mixture model, defined by any given BNP prior. The applicability of this new algorithm mainly depends on the ability to generate samples from the prior, and is not confined to tractable BNP priors. This new algorithm can easily deal with different BNP priors with varying complexity, with small changes to the algorithm.

The BNP-CAEM algorithm, because of its speed and applicability, can be used to speed up other posterior estimation algorithms. For example, when a sample from the posterior distribution is desired for the given BNP mixture model, the BNP-CAEM output of the MAP clustering and associated statistics can be used to provide starting values, or to define a proposal distribution, for a MCMC Metropolis–Hastings posterior sampling algorithm (see also Daumé III 2007; Fuentes-García, Meña, and Walker 2019). Such a strategy spares the MCMC sampling algorithm from having to spend many initial iterations to find a high posterior probability region, while accelerating MCMC convergence. Also, the BNP-CAEM algorithm can be used in any distributed parallel computing scheme that perform multi-stage hierarchical clustering of massive data (Ni et al. 2019; Zuanetti et al. 2019), in place of the MCMC algorithm.

Finally, the BNP-CAEM algorithm can be applied to the analysis of streaming data. Such data can arrive at high-speed, are large or potentially infinite in size, and face severe storage limitations. This means that each new data point in the stream can be examined at most once, and all old data points need to be discarded (Nguyen, Woon, and Ng 2015). Yet, in many real applications, people require real-time responses from continuously updated results from streaming data (e.g., density estimates and/or clustering estimates), while only making a single pass through the data stream, to address the storage and time constraints (Nguyen, Woon, and Ng 2015). The BNP-CAEM algorithm can provide an accurate analysis of streaming data, in a single pass through the data stream, through the analysis of a reservoir sample obtained from the data stream at a given time point of interest (Guha and Mishra 2016). Indeed, reservoir sampling (Vitter 1985) can be used to represent the data stream continuously over time.

These other applications of the BNP-CAEM algorithm deserves future research.

Supplementary Materials

Information which supplements the main article is available in the following online files.

Appendices.pdf It contains all appendices to the main document. (Portable document format).

Code.zip It includes the MATLAB code to perform all the described BNP-CAEM, VB, MCMC, and EM-MLE algorithms and methods, and includes all the datasets analyzed in this article. (Compressed folder).

Funding

The author's research was supported by NSF grant SES-1156372. He appreciated helpful conversations with Fabrizio Leisen and Antonio Lijoi about certain properties of BNP priors.

References

Arbel, J., and Prünster, I. (2017), "A Moment-Matching Ferguson & Klass Algorithm," *Statistics and Computing*, 27, 3–17. [2,6]

Argiento, R., Bianchini, I., and Guglielmi, A. (2016), "Posterior Sampling From ε-Approximation of Normalized Completely Random Measure Mixtures," *Electronic Journal of Statistics*, 10, 3516–3547. [4]

Banfield, J., and Raftery, A. (1993), "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, 49, 803–821. [7]

Beckman, R., and McKay, M. (1987), "Monte Carlo Estimation Under Different Distributions Using the Same Simulation," *Technometrics*, 29, 153–160. [2]

Blei, D., and Jordan, M. (2006), "Variational Inference for Dirichlet Process Mixtures," *Bayesian Analysis*, 1, 121–144. [6]

Campbell, T., Huggins, J., How, J., and Broderick, T. (2019), "Truncated Random Measures," *Bernoulli*, 25, 1256–1288. [2,3]

Celeux, G., and Govaert, G. (1992), "A Classification EM Algorithm for Clustering and Two Stochastic Versions," *Computational Statistics and Data Analysis*, 14, 315–332. [2,5,11]

Chen, M. (2016), "Dirichlet Process Gaussian Mixture Model," MATLAB Central File Exchange, MATLAB Code, available at *https://www.mathworks.com/matlabcentral/fileexchange/55865-dirichlet-process-gaussian-mixture-model*. [6]

——— (2019), "EM Algorithm for Gaussian Mixture Model (EM GMM)", MATLAB Central File Exchange, MATLAB Code, available at *https://www.mathworks.com/matlabcentral/fileexchange/26184-em-algorithm-for-gaussian-mixture-model-em-gmm*. [6]

Dahl, D. (2006), "Model-Based Clustering for Expression Data via a Dirichlet Process Mixture Model," in *Bayesian Inference for Gene Expression and Proteomics*, eds. K.-A. Do, P. Müller, and M. Vannucci, Cambridge, UK: Cambridge University Press, pp. 201–218. [6]

Daumé III, H. (2007), "Fast Search for Dirichlet Process Mixture Models," in *Proceedings of Machine Learning Research*, eds. M. Meila and X. Shen, San Juan, Puerto Rico: PMLR, pp. 83–90. [1,11]

DeBlasi, P., Favaro, S., Lijoi, A., Meña, R., Prünster, I., and Ruggiero, M. (2015), "Are Gibbs-Type Priors the Most Natural Generalization of the Dirichlet Process?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 212–229. [1]

Dempster, A., Laird, N., and Rubin, D. (1977), "Maximum Likelihood From Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, Series B, 39, 1–38. [2]

Eisenstein, J. (2012), "Dirichlet Process Mixture Model Code in MATLAB. Sampling and Variational," GitHub, available at *https://github.com/jacobeisenstein/DPMM*. [6]

Ferguson, T. (1973), "A Bayesian Analysis of Some Nonparametric Problems," *The Annals of Statistics*, 1, 209–230. [1,3]

Ferguson, T., and Klass, M. (1972), "A Representation of Independent Increment Processes Without Gaussian Components," *The Annals of Mathematical Statistics*, 43, 1634–1643. [3,4]

Fortini, S., and Petrone, S. (2020), "Quasi-Bayes Properties of a Recursive Procedure for Mixtures," *Journal of the Royal Statistical Society, Series B*, 82, 1087–1114 [1]

Fuentes-García, R., Meña, R., and Walker, S. (2019), "Modal Posterior Clustering Motivated by Hopfield's Network," *Computational Statistics and Data Analysis*, 137, 92–100. [1,11]

Gates, A., and Ahn, Y.-Y. (2017), "The Impact of Random Models on Clustering Similarity," *Journal of Machine Learning Research*, 18, 1–28. [6]

Gelfand, A., and Mukhopadhyay, S. (1995), "On Nonparametric Bayesian Inference for the Distribution of a Random Sample," *Canadian Journal of Statistics*, 23, 411–420. [6]

Guha, S., and Mishra, N. (2016), "Clustering Data Streams," in *Data Stream Management: Processing High-Speed Data Streams*, eds. M. Garofalakis, J. Gehrke, and R. Rastogi, Berlin, Heidelberg: Springer, pp. 169–187. [11]

Hjort, N. (1990), "Nonparametric Bayes Estimators Based on Beta Processes in Models for Life History Data," *The Annals of Statistics*, 18, 1259–1294. [3]

Hjort, N., Holmes, C., Müller, P., and Walker, S. (2010), *Bayesian Nonparametrics*, New York: Cambridge University Press. [1]

Ishwaran, H., and James, L. (2001), "Gibbs Sampling Methods for Stick-Breaking Priors," *Journal of the American Statistical Association*, 96, 161–173. [1,3,4,6]

Karabatsos, G., and Walker, S. (2012), "Adaptive-Modal Bayesian Nonparametric Regression," *Electronic Journal of Statistics*, 6, 2038–2068. [4]

Kingman, J. (1967), "Completely Random Measures," *Pacific Journal of Mathematics*, 21, 59–78. [1,2]

——— (1975), "Random Discrete Distributions," *Journal of the Royal Statistical Society*, Series B, 37, 1–22. [3]

Lijoi, A., Meña, R., and Prünster, I. (2005a), "Bayesian Nonparametric Analysis for a Generalized Dirichlet Process Prior," *Statistical Inference for Stochastic Processes*, 8, 283–309. [1,3,6]

——— (2005b), "Hierarchical Mixture Modeling With Normalized Inverse-Gaussian Priors," *Journal of the American Statistical Association*, 100, 1278–1291. [3]

——— (2007), "Controlling the Reinforcement in Bayesian Nonparametric Mixture Models," *Journal of the Royal Statistical Society*, Series B, 69, 715–740. [1,3,6]

Lijoi, A., and Prünster, I. (2010), "Models Beyond the Dirichlet Process," in *Bayesian Nonparametrics*, eds. N. Hjort, C. Holmes, P. Müller, and S. Walker, Cambridge: Cambridge University Press, pp. 80–136. [1,3]

Lo, A. (1984), "On a Class of Bayesian Nonparametric Estimates," *The Annals of Statistics*, 12, 351–357. [1]

Meña, R. (2013), "Geometric Weight Priors and Their Applications in Bayesian Nonparametrics," in *Bayesian Theory and Applications*, eds. P. Damien, P. Dellaportas, N. Polson, and D. Stephens, Oxford: Oxford University Press, pp. 271–296. [3]

Mitra, R., and Müller, P. (2015), *Nonparametric Bayesian Inference in Biostatistics*, Basel: Springer. [1]

Müller, P., Erkanli, A., and West, M. (1996), "Bayesian Curve Fitting Using Multivariate Normal Mixtures," *Biometrika*, 83, 67–79. [4]

Neal, R. (2000), "Markov Chain Sampling Methods for Dirichlet Process Mixture Models," *Journal of Computational and Graphical Statistics*, 9, 249–265. [3,6]

Nguyen, H.-L., Woon, Y.-K., and Ng, W.-K. (2015), "A Survey on Data Stream Clustering and Classification," *Knowledge and Information Systems*, 45, 535–569. [11]

Ni, Y., Müller, P., Diesendruck, M., Williamson, S., Zhu, Y., and Ji, Y. (2019), "Scalable Bayesian Nonparametric Clustering and Classification," *Journal of Computational and Graphical Statistics*, 29, 53–65. [11]

Nielsen, S. (2000), "The Stochastic EM Algorithm: Estimation and Asymptotic Results," *Bernoulli*, 6, 457–489. [2,6]

Pennell, M., and Dunson, D. (2006), "Bayesian Semiparametric Dynamic Frailty Models for Multiple Event Time Data," *Biometrics*, 62, 1044–1052. [10]

Pitman, J. (1996), "Some Developments of the Blackwell-MacQueen Urn Scheme," in *Statistics, Probability and Game Theory. Papers in Honor of David Blackwell*, eds. T. Ferguson, L. Shapeley, and J. MacQueen, Hayward, CA: Institute of Mathematical Sciences, pp. 245–268. [3]

——— (2003), "Poisson–Kingman partitions," in *Science and Statistics: A Festschrift for Terry Speed, Institute of Mathematical Statistics, Hayward, Lecture Notes, Monograph Series*, ed. D. Goldstein, Beachwood, OH: Institute of Mathematical Statistics, pp. 1–35. [3,4]

Rastelli, R., and Friel, N. (2018), "Optimal Bayesian Estimators for Latent Variable Cluster Models," *Statistics and Computing*, 28, 1169–1186. [2]

Raykov, Y., Boukouvalas, A., and Little, M. (2016), "Simple Approximate MAP Inference for Dirichlet Processes Mixtures," *Electronic Journal of Statistics*, 10, 3548–3578. [1]

Regazzini, E., Lijoi, A., and Prünster, I. (2003), "Distributional Results for Means of Normalized Random Measures With Independent Increments," *The Annals of Statistics*, 31, 560–585. [3,4]

Richardson, S., and Green, P. (1997), "On Bayesian Analysis of Mixtures With an Unknown Number of Components," *Journal of the Royal Statistical Society*, Series B, 59, 731–792. [6,7]

Rodríguez, A., Dunson, D., and Gelfand, A. (2009), "Bayesian Nonparametric Functional Data Analysis Through Density Estimation," *Biometrika*, 96, 149–162. [4]

Roeder, K. (1990), "Density Estimation With Confidence Sets Exemplified by Superclusters and Voids in the Galaxies," *Journal of the American Statistical Association*, 85, 617–624. [7]

Stahl, D., and Sallis, H. (2012), "Model-Based Cluster Analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, 4, 341–358. [6]

Symons, M. (1981), "Clustering Criteria and Multivariate Normal Mixtures," *Biometrics*, 37, 35–43. [4]

Teh, Y., and Gorür, D. (2009), "Indian Buffet Processes With Power-Law Behavior," in *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, eds. Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Red Hook, NY: Curran Associates, Inc., pp. 1838–1846. [2,3]

Van Laarhoven, P., and Aarts, E. (1987), *Simulated Annealing: Theory and Applications*, Dordrecht: Reidel. [5]

Vitter, J. (1985), "Random Sampling With a Reservoir," *ACM Transactions on Mathematical Software*, 11, 37–57. [11]

Wang, L., and Dunson, D. (2011), "Fast Bayesian Inference in Dirichlet Process Mixture Models," *Journal of Computational and Graphical Statistics*, 20, 196–216. [6]

Woodward, W., Parr, W., Schucany, W., and Lindsey, H. (1984), "A Comparison of Minimum Distance and Maximum Likelihood Estimation of a Mixture Proportion," *Journal of the American Statistical Association*, 79, 590–598. [5]

Zuanetti, D., Müller, P., Zhu, Y., Yang, S., and Ji, Y. (2019), "Bayesian Nonparametric Clustering for Large Data Sets," *Statistics and Computing*, 29, 203–215. [1,11]