

# BNPlib for density estimation

A nonparametric C++ library

Bruno Guindani

Elena Zazzetti



**POLITECNICO**  
MILANO 1863

November 22, 2019

<https://github.com/poliprojects/BNPlib>

# Non-Parametric statistics

- Goal: density estimation
- **Infinite-dimensional** parameters, e.g. functions

# Non-Parametric statistics (the Bayesian way)

- Goal: density estimation
- **Infinite-dimensional** parameters, e.g. functions
- Model:

$$y_i|G \stackrel{\text{iid}}{\sim} G, \quad i = 1, \dots, n$$
$$G \sim \mathcal{P}$$

# Non-Parametric statistics (the Bayesian way)

- Goal: density estimation
- **Infinite-dimensional** parameters, e.g. functions
- Model:

$$y_i | G \stackrel{\text{iid}}{\sim} G, \quad i = 1, \dots, n$$
$$G \sim \mathcal{P}$$

$$\mathcal{P} : \Omega \rightarrow M(S) \text{ fixed}$$
$$[ \omega \mapsto G(\cdot) ]$$

- Model name: **BNP model**

# Dirichlet Process prior

$$y_i | G \stackrel{\text{iid}}{\sim} G$$
$$G \sim \mathcal{P} = DP(MG_0)$$

- Parameters:  $M > 0$ ,  $G_0 \in M(S)$
- Defining property:  $\forall \{B_{1:k}\}$  partition of  $S$ ,

$$[G(B_1), \dots, G(B_k)] \sim \text{Dir}(MG_0(B_1), \dots, MG_0(B_k))$$

# Dirichlet Process prior

$$y_i | G \stackrel{\text{iid}}{\sim} G$$
$$G \sim \mathcal{P} = DP(MG_0)$$

- Parameters:  $M > 0$ ,  $G_0 \in M(S)$
- Defining property:  $\forall \{B_{1:k}\}$  partition of  $S$ ,

$$[G(B_1), \dots, G(B_k)] \sim \text{Dir}(MG_0(B_1), \dots, MG_0(B_k))$$

- **Discreteness** (stick-breaking):  $G(\cdot) = \sum_{k=1}^{+\infty} w_h \delta_{m_h}(\cdot)$
- **Conjugacy**:  $G|\mathbf{y} \sim DP(MG_0 + \sum_i \delta_{y_i}) \implies$  density estimation

# Continuous density estimation

- **Mixtures** (kernel  $f$  + mixing distribution  $G$ ):

$$y_i|G \sim f_G(y) = \int f_{\vartheta}(y) \mathrm{d}G(\vartheta)$$

# Continuous density estimation

- **Mixtures** (kernel  $f$  + mixing distribution  $G$ ):

$$y_i|G \sim f_G(y) = \int f_{\vartheta}(y) \mathrm{d}G(\vartheta)$$
$$G \sim DP(MG_0)$$

- Model name: **DPM model**



# Continuous density estimation

- **Mixtures** (kernel  $f$  + mixing distribution  $G$ ):

$$y_i|G \sim f_G(y) = \int f_{\vartheta}(y) \mathrm{d}G(\vartheta)$$
$$G \sim DP(MG_0)$$

- Model name: **DPM model**
- Equivalent to:

$$y_i|\vartheta_i \stackrel{\text{iid}}{\sim} f_{\vartheta_i}$$
$$\vartheta_i|G \stackrel{\text{iid}}{\sim} G$$
$$G \sim DP(MG_0)$$

- $\vartheta_i$  “latent variables”  $\forall i = 1, \dots, n$

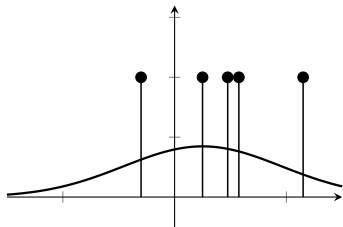
# Clustering in the DPM

- Discreteness: the  $\vartheta_i$  have one of the  $k$  **unique values**  $\phi_j$  ( $j = 1, \dots, k$ )
- $k \simeq M \log(n) \ll n$
- All  $i$  s.t.  $\vartheta_i = \phi_j$  belong to cluster  $S_j$  ( $j = 1, \dots, k$ ), and  $n_j = |S_j|$

# Clustering in the DPM

- Discreteness: the  $\vartheta_i$  have one of the  $k$  **unique values**  $\phi_j$  ( $j = 1, \dots, k$ )
- $k \simeq M \log(n) \ll n$
- All  $i$  s.t.  $\vartheta_i = \phi_j$  belong to cluster  $S_j$  ( $j = 1, \dots, k$ ), and  $n_j = |S_j|$
- Conditional prior for  $\vartheta_i$ :

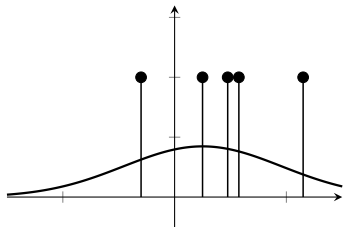
$$\mathcal{L}(\vartheta_i | \boldsymbol{\vartheta}_{-i}) \propto \sum_{j=1}^{k^-} \underset{\uparrow}{n_j^-} \delta_{\underset{\uparrow}{\phi_j^-}}(\vartheta_i) + MG_0(\vartheta_i)$$



# Clustering in the DPM

- Discreteness: the  $\vartheta_i$  have one of the  $k$  **unique values**  $\phi_j$  ( $j = 1, \dots, k$ )
- $k \simeq M \log(n) \ll n$
- All  $i$  s.t.  $\vartheta_i = \phi_j$  belong to cluster  $S_j$  ( $j = 1, \dots, k$ ), and  $n_j = |S_j|$
- Conditional prior for  $\vartheta_i$ :

$$\mathcal{L}(\vartheta_i | \boldsymbol{\vartheta}_{-i}) \propto \sum_{j=1}^{k^-} \underset{\uparrow}{n_j^-} \delta_{\phi_j^-}(\vartheta_i) + \underset{\uparrow}{M} G_0(\vartheta_i)$$



- Conditional posterior for  $\vartheta_i$ :

$$\mathcal{L}(\vartheta_i | \boldsymbol{\vartheta}_{-i}, y_i) \propto \sum_{j=1}^{k^-} f_{\vartheta}(y_i) \delta_{\phi_j^-}(\vartheta_i) + M r_i G_0(\vartheta_i | y_i)$$

# Discrete model

Equivalent models as  $K \rightarrow +\infty$ :

$(Y_i \vartheta_i) \sim F(\vartheta_i)$	$(Y_i \phi, c_i) \sim F(\phi_{c_i})$
$(\vartheta_i G) \sim G$	$(c_i \mathbf{p}) \sim \text{Discrete}(p_1, \dots, p_K)$
$G \sim DP(M, G_0)$	$\phi_c \sim G_0$
	$\mathbf{p} \sim \text{Dirichlet}(M/K, \dots, M/K)$
(hierarchical model)	(discrete model)

with  $c_i = j \iff \vartheta_i \in S_j$  allocation parameters and

$$\boldsymbol{\vartheta} \rightsquigarrow (\boldsymbol{\phi}, \mathbf{c})$$

# Neal's Algorithm 2

## Gibbs sampling algorithm:

- $(\phi, \mathbf{c})$  is the **state** of a Markov chain
- For  $i = 1, \dots, n$ : update  $c_i$ 
  - ▶ If  $c_i$  allocates  $\phi_i$  to a singleton, remove  $\phi_{c_i}$  from the state
  - ▶ Sample  $c_i$  as follows:

$$\text{If } c = c_j \text{ for some } j \neq i: \mathbb{P}(c_i = c | c_{-i}, y_i, \phi) \propto \frac{n_{-i,c}}{n - 1 - M} F(y_i, \phi_c)$$

$$\mathbb{P}(c_i \neq c_j \text{ for all } j | c_{-i}, y_i, \phi) \propto \frac{M}{n - 1 - M} \int F(y_i, \phi) dG_0(\phi)$$

# Neal's Algorithm 2

## Gibbs sampling algorithm:

- $(\phi, \mathbf{c})$  is the **state** of a Markov chain
- For  $i = 1, \dots, n$ : update  $c_i$ 
  - ▶ If  $c_i$  allocates  $\phi_i$  to a singleton, remove  $\phi_{c_i}$  from the state
  - ▶ Sample  $c_i$  as follows:

$$\text{If } c = c_j \text{ for some } j \neq i: \mathbb{P}(c_i = c | c_{-i}, y_i, \phi) \propto \frac{n_{-i,c}}{n-1-M} F(y_i, \phi_c)$$
$$\mathbb{P}(c_i \neq c_j \text{ for all } j | c_{-i}, y_i, \phi) \propto \frac{M}{n-1-M} \int F(y_i, \phi) dG_0(\phi)$$

- ▶ If the new  $c_i$  allocates  $\phi_i$  to a singleton, draw  $\phi_{c_i} \sim G_0 | y_i$  and add it to the state
- For  $c \in \{c_1, \dots, c_n\}$ : update  $\phi_c$ , given all the  $y_i$  with  $c_i = c$

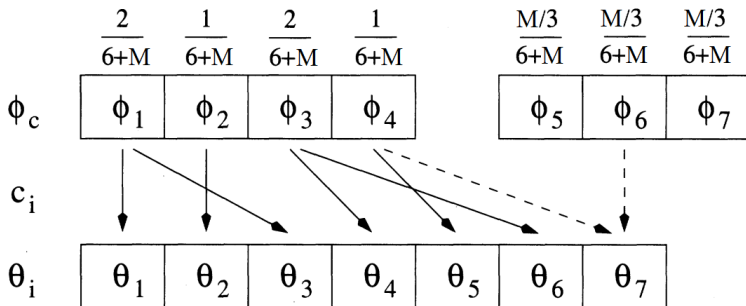
# Advantages

- Feasible if we can compute  $\int F(y_i, \phi) dG_0(\phi)$  and sample from  $H_i$  (generally conjugate case)
- Change the  $\vartheta$  for more than one observation simultaneously
-



# Neal's Algorithm 8

- **Gibbs sampling** to the state extended by the addition of  $m$  **auxiliary parameters**



- Prior for  $c_i$ :

$$\text{If } c = c_j \text{ for some } j: \mathbb{P}(c_i = c | c_{-i}) = \frac{n_{-i,c}}{n - 1 - M}$$

$$\mathbb{P}(c_i \neq c_j \text{ for all } j) = \frac{M}{n - 1 - M} \Rightarrow \text{split among the } m \text{ auxiliary parameters}$$

# Neal's Algorithm 8

Algorithm:

- For  $i = 1, \dots, n$ : update  $c_i$ 
  - ▶ Sample auxiliary parameters:
    - $c_i = c_j$  for some  $j \Rightarrow$  no connection
    - $c_i \neq c_j \Rightarrow$  association to one of  $m$

The other  $\phi$  values drawn from  $G_0$

- ▶ Draw  $c_i$  as follows:

$$P(c_i = c | c_{-i}, y_i, \phi_1, \dots, \phi_h) \propto \begin{cases} \frac{n_{-i,c}}{n-1-M} F(y_i, \phi_c), & \text{for } 1 \leq c \leq k^- \\ \frac{M/m}{n-1-M} F(y_i, \phi_c), & \text{for } k^- + 1 < c \leq h \end{cases}$$

- ▶ Discard  $\phi$  values not associated to any cluster
- For  $c \in \{c_1, \dots, c_n\}$ : update  $\phi_c$  given  $y_i$  such that  $c_i = c$

# Advantages

- Models with non-conjugate priors
- As  $m \rightarrow +\infty$  it approaches Algorithm 2 but equilibrium distribution is exact
- More efficient than similar algorithms (e.g. no-gaps)
- Hierarchical extensions

# Stick-Breaking Priors

$$\mathcal{P}(\cdot) = \sum_{k=1}^N p_k \delta_{Z_k}(\cdot)$$

$$p_k = V_1 \text{ and } p_k = (1 - V_1)(1 - V_2) \cdots (1 - V_{k-1})V_k$$

$$\mathbf{Z}_k \stackrel{\text{iid}}{\sim} H$$

$$V_k \stackrel{\text{iid}}{\sim} \text{Beta}(a_k, b_k)$$

$$\mathbf{a} = (a_1, a_2, \dots) \text{ and } \mathbf{b} = (b_1, b_2, \dots)$$

$$0 \leq p_k \leq 1 \text{ and } \sum_{k=1}^N p_k = 1$$

- $N < +\infty$ :  $\mathcal{P}_N(\mathbf{a}, \mathbf{b})$ 
  - ▶  $\mathbf{p} \sim \mathcal{GD}(\mathbf{a}, \mathbf{b})$
  - ▶ e.g. all finite dimensional Dirichlet priors
- $N = +\infty$ :  $\mathcal{P}_\infty(\mathbf{a}, \mathbf{b})$ 
  - ▶ e.g. Dirichlet process, the two parameter Poisson-Dirichlet process

# Blocked Gibbs Algorithm

- Assumption: **finite-dimensional** prior  $P \sim \mathcal{P}_N(\mathbf{a}, \mathbf{b})$
- Finite number of variables  $\Rightarrow$  *blocks of parameters*
- **Model:**

$$(Y_i | \phi, \mathbf{c}) \stackrel{\parallel}{\sim} F(\phi_{c_i}), \quad i = 1, \dots, n$$

$$(c_i | \mathbf{p}) \stackrel{\text{iid}}{\sim} \sum_{k=1}^N p_k \delta_k(\cdot)$$

$$\mathbf{p} \sim \mathcal{GD}(\mathbf{a}, \mathbf{b})$$

$$\phi_c \sim G_0$$

# Blocked Gibbs Algorithm

## Algorithm:

Repeatedly drawing values from conditional distributions of the blocked variables:

- $(\phi|\mathbf{c}, \mathbf{Y})$
- $(\mathbf{c}|\phi, \mathbf{p}, \mathbf{Y})$
- $(\mathbf{p}|\mathbf{c})$



## Direct sampling of the posterior $\mathcal{P}(\cdot|\mathbf{Y})$ :

- The Algorithm produces draws from  $(\phi, \mathbf{c}, \mathbf{p}|\mathbf{Y})$
- Each draw  $(\phi, \mathbf{c}, \mathbf{p})$  defines a measure  $P(\cdot) = \sum_{k=1}^N p_k \delta_{\phi_k}(\cdot)$
- Each  $P$  is a drawn from  $\mathcal{P}(\cdot|\mathbf{Y})$

# Advantages

- Handles the issue of conjugacy
- Good mixing
- Hierarchical extensions

# Bibliography

-  Muller, Quintana, *Bayesian Nonparametric Data Analysis*
-  Neal (2000), *Markov Chain Sampling Methods for Dirichlet Process Mixture Models*
-  Ishwaran, James (2001), *Gibbs Sampling Methods for Stick-Breaking Priors*