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
- Model:

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

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

- Model name: **BNP model**

Dirichlet Process Prior

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

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

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

- **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):

$$
\begin{aligned}
y_i | G \sim F_G(y) &= \int F(y, \vartheta)\, G(\mathrm{d}\vartheta) \\
G &\sim DP(MG_0)
\end{aligned}
$$

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

$$
\begin{aligned}
y_i | \vartheta_i &\overset{\perp\!\!\!\perp}{\sim} F(\cdot, \vartheta_i) \\
\vartheta_i | G &\overset{\text{iid}}{\sim} G \\
G &\sim DP(MG_0)
\end{aligned}
$$

- ϑ_i "latent variables" $\forall i = 1, \ldots, n$

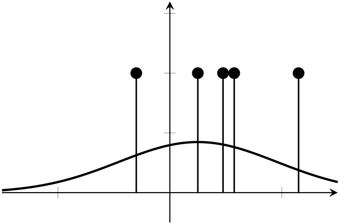
Clustering In The DPM

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

Clustering In The DPM

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

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

Discrete Model

$$\begin{aligned}(Y_i|\vartheta_i) &\sim F(\cdot, \vartheta_i)\\ (\vartheta_i|G) &\sim G\\ G &\sim DP(M, G_0)\end{aligned} \quad \overset{K\to+\infty}{\Longleftrightarrow} \quad \begin{aligned}(Y_i|\phi, c_i) &\sim F(\cdot, \phi_{c_i})\\ (c_i|\mathbf{p}) &\sim \text{Discrete}(p_1, \ldots, p_K)\\ \phi_c &\sim G_0\\ \mathbf{p} &\sim \mathrm{Dir}(M/K, \ldots, M/K)\end{aligned}$$

(hierarchical model) (K-discrete model)

with c_i allocation parameters and

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

Neal's Algorithm 2

Gibbs sampling algorithm:

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

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

$$\mathbb{P}(c_i \neq c_j \text{ for all } j | \mathbf{c}_{-i}, y_i, \boldsymbol{\phi}) \propto \frac{M}{n-1-M} \int F(y_i, \phi)\, G_0(\mathrm{d}\phi)$$

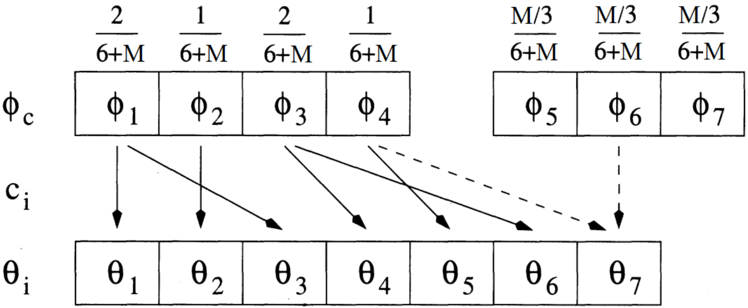
  - If the new c_i allocates ϕ_i to a singleton, draw $\phi_{c_i} \sim G_0(\cdot | y_i)$ and add it to the state
- For $c \in \{c_1, \ldots, c_n\}$: update ϕ_c, given all the y_i with $c_i = c$

Advantages

- Feasible if we can compute $\int F(y_i, \phi) G_0(\mathrm{d}\phi)$ and sample from $G_0(\cdot|y_i)$ (conjugate case)
- Change the ϑ for more than one observation simultaneously

Neal's Algorithm 8

- **Gibbs sampling** on the state, which is extended by the addition of m **auxiliary parameters**



- Prior for c_i:

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

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

Neal's Algorithm 8

Algorithm:

- For $i = 1, \ldots, n$: 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 ϕ values drawn from G_0
  - Draw c_i as follows:

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

  - Discard values in $\boldsymbol{\phi}$ not associated to any ϑ_j
- For $c \in \{c_1, .., c_n\}$: update ϕ_c given y_i such that $c_i = c$

Advantages

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

Stick-Breaking Priors

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

with:

- $Z_k \overset{\text{iid}}{\sim} H$ (allocations)
- $V_k \overset{\text{iid}}{\sim} \mathrm{Beta}(a_k, b_k)$ with $\mathbf{a} = (a_1, a_2, \ldots)$ and $\mathbf{b} = (b_1, b_2, \ldots)$
- $p_k = (1 - V_1)(1 - V_2)\cdots(1 - V_{k-1})V_k$ (weights)

  with $0 \leq p_k \leq 1, \sum_{k=1}^{N} p_k = 1$

Dimension:

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

Blocked Gibbs Algorithm

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

$$
\begin{aligned}
(Y_i | \phi, \mathbf{c}) &\overset{\perp\!\!\!\perp}{\sim} F(\cdot, \phi_{c_i}),\ \ i = 1, .., n \\
(c_i | \mathbf{p}) &\overset{\text{iid}}{\sim} \sum_{k=1}^{N} p_k \delta_k(\cdot) \\
\mathbf{p} &\sim \mathscr{G}\mathscr{D}(\mathbf{a}, \mathbf{b}) \\
\phi_c &\sim G_0
\end{aligned}
$$

Blocked Gibbs Algorithm

Algorithm:

- Repeatedly draw values from the conditional distributions of the blocked variables:

$$\begin{aligned}
\boldsymbol{\phi} &\sim \mathcal{L}(\boldsymbol{\phi}|\mathbf{c}, \mathbf{y}) \\
\mathbf{c} &\sim \mathcal{L}(\mathbf{c}|\boldsymbol{\phi}, \mathbf{p}, \mathbf{y}) \\
\mathbf{p} &\sim \mathcal{L}(\mathbf{p}|\mathbf{c})
\end{aligned}$$

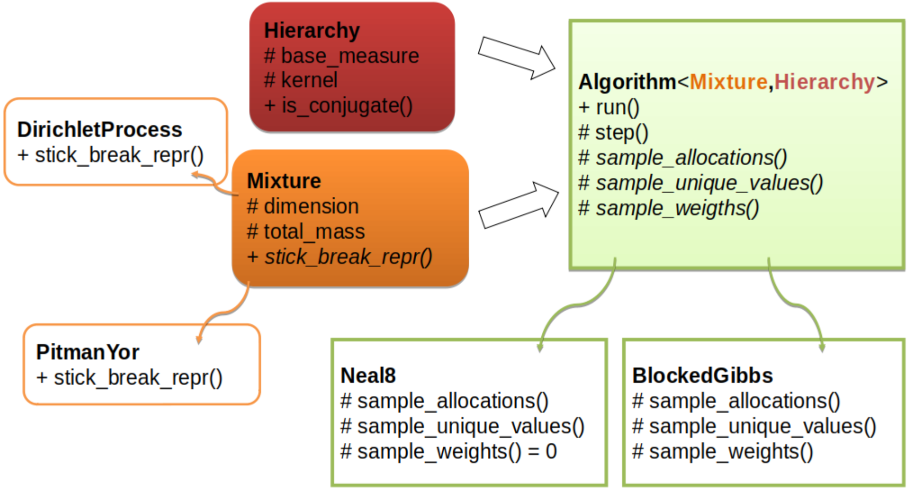
Direct sampling of the **posterior** $\mathscr{P}(\cdot|\mathbf{y})$:

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

Advantages

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

Code Structure



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*