

BNPlib for density estimation:

A nonparametric C++ library
(part 2)

Bruno Guindani
Elena Zazzetti

January 8th, 2020



POLITECNICO
MILANO 1863

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

Model

$$\begin{array}{ll} (Y_i|\vartheta_i) \sim F(\cdot, \vartheta_i) & (Y_i|\phi, c_i) \sim F(\cdot, \phi_{c_i}) \\ (\vartheta_i|G) \sim G & (c_i|\mathbf{p}) \sim \sum_{k=1}^K p_k \delta_k(\cdot) \\ G \sim DP(M, G_0) & \xLeftrightarrow{K \rightarrow \infty} \\ & \phi_c \sim G_0 \\ & \mathbf{p} \sim \text{Dir}(M/K, \dots, M/K) \\ \text{(hierarchical model)} & \text{(K-discrete model)} \end{array}$$

with $\vartheta \longleftrightarrow (\phi, \mathbf{c})$

Algorithms

- Neal2, Neal8, blocked Gibbs
- **Gibbs sampling** procedures
- General structure:
 - ▶ sample **allocations** c from some conditional distribution
 - ▶ sample **unique values** ϕ from some conditional distribution
 - ▶ (sample **weight Gibbs samplings** p of the unique values deltas)

Title

- Text

The algorithms in C++

```
299
300     void run(){
301         initialize();
302         unsigned int iter = 0;
303         while(iter < maxiter){
304             step();
305             if(iter >= burnin)
306                 save_iteration(iter);
307         }
308     }
```

- Example: Normal-NormalInvGamma hierarchy, no hyperpriors
- initialize(): random allocation
- step()
 - ▶ sample_allocations(): vector card of cardinalities of clusters
 - ▶ 4 cases handled separately: singleton vs !singleton, aux vs old
 - ▶ sample_unique_values(): vector clust_idx to record which data are in each cluster
- Actual cluster structures?

Impending extensions





- **Hyperpriors:** objects of class `Hypers` store pointers to objects of class `HypersFixed`
- **Non-conjugacy:** via Stan's HMC sampler
- **R interface:** via **protocol buffers**

Protocol Buffers

- API developed by Google
- Data is saved in XML-like structures, called **messages**, that are defined in .proto files
- Each message corresponds to a class in C++
- The protoc compiler produces the C++ files that make up the API
- RProtoBuf
- Compromise between efficiency and human-readability

A general library?

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*
-  <https://developers.google.com/protocol-buffers/docs/cpptutorial>