# **BNPlib** for density estimation:

A nonparametric C++ library

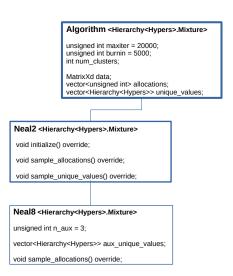
Bruno Guindani Elena Zazzetti



https://github.com/poliprojects/BNPlib

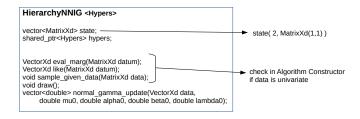
## **Algoritmhs**

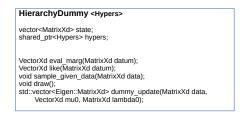
### Algorithm<Hierarchy<Hypers>, Hypers, Mixture>



## Hierarchies

## Template parameter for Algorithm: common interface needed





# Hyperparameters

Template parameter for Algorithm: common interface needed

#### HypersFixedNNIG

double mu0, lambda, alpha0, beta0;

#### HypersDummy

VectorXd mu0; MatrixXd lambda0;

### Mixtures

#### Template parameter for Algorithm: common interface needed

#### DirichletMixture

double totalmass:

double const prob\_existing\_cluster( int card, unsigned int n) double const prob\_new\_cluster( unsigned int n, unsigned int n\_unique)  $\,$ 

#### PitYorMixture

double strength; double discount;

double const prob\_existing\_cluster( int card, unsigned int n) double const prob\_new\_cluster( unsigned int n, unsigned int n\_unique)  $\,$ 

## **Factory**

To choose the Algorithm at runtime:

```
template < class AbstractProduct, typename... Args>
class Factory{
private:
    std::map<Identifier, Builder> storage;
    //[...]
public:
    static Factory& Instance();
    std::unique ptr<AbstractProduct> create object(
        const Identifier &name, Args... args) const;
    void add builder(const Identifier &name,
        const Builder &builder);
   //[...]
```

## **Protocol Buffers**

For multivariate data storage

```
message Par Col {
    repeated double elems = 1;
    }
message Param {
    repeated Par Col par cols = 1;
    }
message UniqueValues {
    repeated Param params = 1;
message IterationOutput {
    repeated int32 allocations = 1;
    repeated UniqueValues uniquevalues = 2;
}
message ChainOutput {
    repeated IterationOutput chain = 1;
}
```

### Collectors

