## **BNPlib** for density estimation:

A nonparametric C++ library

Bruno Guindani Elena Zazzetti



https://github.com/poliprojects/BNPlib

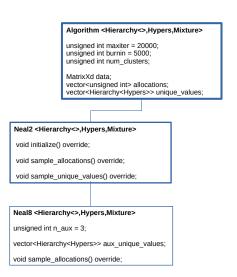
### Algorithms

```
Commmon structure:
```

```
void step(){
    sample allocations();
    sample unique values();
}
void run(BaseCollector* collector){
    print startup message();
    initialize():
    unsigned int iter = 0;
    while(iter < maxiter){
        step();
        if(iter >= burnin){
            save_state(collector, iter);
    iter++;
    }
    print_ending_message();
```

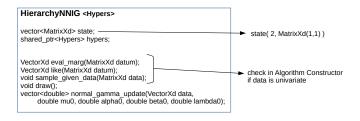
#### **Algoritmhs**

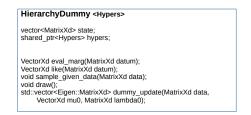
- Structure: Algorithm<Hierarchy<>, Hypers, Mixture>
- Multivariate data supported



# Hierarchies (Hierarchy)

• Template parameter for Algorithm: common interface needed





# Hyperparameters (Hypers)

- Template parameter for Hierarchy
- Used through a pointer to allow simultaneous update of several hierarchies

#### HypersFixedNNIG

double mu0, lambda, alpha0, beta0;

#### HypersDummy

VectorXd mu0; MatrixXd lambda0;

### Mixtures (Mixture)

• 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 API aka protobuf

- For multivariate data storage
- Model in output.proto:

```
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

• To save data with the protobuf structures

