

HierarchyNNIG <Hypers>

```
vector<MatrixXd> state;  
shared_ptr<Hypers> hypers;
```

state(2, MatrixXd(1,1))

```
VectorXd eval_marg(MatrixXd datum);  
VectorXd like(MatrixXd datum);  
void sample_given_data(MatrixXd data);  
void draw();  
vector<double> normal_gamma_update(VectorXd data,  
    double mu0, double alpha0, double beta0, double lambda0);
```

check in Algorithm Constructor
if data is univariate

HierarchyDummy <Hypers>

```
vector<MatrixXd> state;  
shared_ptr<Hypers> hypers;
```

```
VectorXd eval_marg(MatrixXd datum);  
VectorXd like(MatrixXd datum);  
void sample_given_data(MatrixXd data);  
void draw();  
std::vector<Eigen::MatrixXd> dummy_update(MatrixXd data,  
    VectorXd mu0, MatrixXd lambda0);
```