BNP.jl Documentation

Release 0.0.1

OFAI

December 02, 2016

Contents

The *BNP* package integrates a wide range of state-of-the-art Bayesian nonparametric models. In particular:

- Dirichlet Process Mixture Models
- Hierarchical Dirichlet Process Mixture Models
- Factor analysis Models (e.g. Variable Clustering Model)

Contents:

CHAPTER 1

Getting Started

1.1 Installation

The *BNP* package is currently not available through the Julia package system but can easily installed by running `Pkg.clone("https://github.com/trappmartin/BNP.jl")`.

1.2 Clustering data using Dirichlet Process Mixture Model

In this example we start by drawing 100 observations from two bivariate Normal distributions.

```
julia> X = cat(2, rand(2, 50), rand(2, 50) + 10)
julia> Y = cat(2, zeros(50), ones(50))
```

Now we can initialize the package and construct a Gaussian data distribution using a Normal Inverse Wishart prior.

```
julia> using BNP
julia> μ0 = vec( mean(X, 2) )
julia> κ0 = 1.0
julia> ν0 = 4.0
julia> Ψ = eye(2) * 10
julia> G0 = GaussianWishart(μ0, κ0, ν0, Ψ)
```

After constructing G0 we can easily apply a Dirichlet Process Mixture Model using collapsed Gibbs sampling.

```
julia> models = train(DPM(G0), Gibbs(), KMeansInitialisation(), X)
```

Please note that this example can also be found in the demos folder, allowing interactive exploration of the model.

CHAPTER 2

Initialization Methods

In order to initialize the Bayesian nonparametric models we provide a set of initialization approaches. Currently not every initialization approach is available for all models.

2.1 Random Initialization

The Random Initialization randomly assigns the data to a predefined number of groups.

```
julia> init = RandomInitialisation() #  Random Initialization with k = 2
julia> init = RandomInitialisation(k = 5) #  Random Initialization with k = 5
```

2.2 Incremental Initialization

The Incremental Initialization sequentially assigns the data to groups.

```
julia> init = IncrementalInitialisation() #  Incremental Initialization k = 5
```

2.3 K-Means Initialization

The K-Means Initialization assigns the data using k-Means clustering to a predefined number of groups.

```
julia> init = KMeansInitialisation() #  K-Means Initialisation with k = 2
julia> init = KMeansInitialisation(k = 5) #  K-Means Initialisation with k = 5
```

Distributions

The following distributions are currently supported. We will add additional support for the Distributions package in near future.

3.1 Common Interface

A common interface to access the sufficient statistics and the log likelihood is provided for all distributions.

```
julia> add_data!(dist, X) # add datum to dist
julia> dist2 = add_data(dist, X) # add datum to copy of dist
```

```
julia> remove_data!(dist, X) # remove datum from dist
julia> dist2 = remove_data(dist, X) # remove datum from copy of dist
```

```
julia> logpred(dist, X) # log likelihood datum under dist
```

3.2 Beta-Binomial

The Binomial distribution with Beta prior of dimensionality D can be created using:

```
julia> dist = BinomialBeta(D) # with default α = 1.0 and β = 1.0
julia> dist = BinomialBeta(D, α = 3, β = 4) # specify α and β parameter of Beta distribution
```

3.3 Dirichlet-Multinomial

The Multinomial distribution with Dirichlet prior of dimensionality D can be created using:

```
julia> dist = MultinomialDirichlet(D, 1.0) # with default α = 1.0
```

3.4 Wishart-Gaussian

The Gaussian distribution with Wishart prior of dimensionality D can be created using:

```
julia> dist = GaussianWishart(μ, κ, ν, Ψ) # with specified μ of dimensionality D, κ, ν and Ψ of dime
```