

Package ‘Bmix’

February 14, 2012

Type Package

Title Bayesian Sampling for Stick-breaking Mixtures

Version 0.2

Date 2009-10-28

Author Matt Taddy <matt.taddy@chicagobooth.edu>

Maintainer Matt Taddy <matt.taddy@chicagobooth.edu>

Description This is a bare-bones implementation of sampling algorithms for a variety of Bayesian stick-breaking (marginally DP) mixture models, including particle learning and Gibbs sampling for static DP mixtures, particle learning for dynamic BAR stick-breaking, and DP mixture regression. The software is designed to be easy to customize to suit different situations and for experimentation with stick-breaking models. Since particles are repeatedly copied, it is not an especially efficient implementation.

License GPL (>= 2)

LazyLoad yes

Depends mvtnorm

URL <http://faculty.chicagobooth.edu/matt.taddy/software>

Repository CRAN

Date/Publication 2009-10-29 06:51:50

R topics documented:

| | |
|------------------------|----------|
| Bmix-package | 2 |
| particle | 4 |
| Index | 5 |

Description

Basic implementation for a variety of stick-breaking mixture models, including particle learning and Gibbs sampling for static DP mixtures, particle learning for dynamic BAR stick-breaking, and DP mixture regression.

Usage

```
mix(Z, alpha, g0params, times=NULL, rho=NULL, cat=0,
    state=NULL, read=FALSE, print=FALSE, N=100, niter=0, kout=FALSE)
```

Arguments

| | |
|----------|--|
| Z | data.frame of observations, with the last cat columns categorical variables. |
| alpha | Stick-breaking precision parameter. |
| g0params | A vector of the base measure parameters: $[\gamma, \kappa, \nu, \gamma_\Omega, \Psi_\Omega]$, as in Taddy (2009), followed by a list of Dirichlet parameters for each categorical variable. |
| times | Discrete time period for each data row; must be an increasing list of integers starting at 0. |
| rho | Stick-breaking correlation parameter (between 0 and 1). |
| cat | Number of categorical variables. |
| state | Random number generator seed. |
| read | Read in existing particle info? |
| print | Print out particles for each new time? WARNING: This will litter your working directory with .particle* files! |
| N | Number of particles. |
| niter | Number of Gibbs sampling iterations after filtering (only for times = NULL). |
| kout | Return filtered (not smoothed) allocations? |

Details

This is a bare-bones implementation of sampling algorithms for Bayesian stick-breaking mixture models. The software is designed to be easy to customize to suit different situations and for experimentation with stick-breaking models. Since particles are repeatedly copied, it is not especially efficient.

The package implements particle learning (Carvalho et al, 2009) for both dynamic and constant stick-breaking mixture models, and collapsed Gibbs sampling for DP mixtures. Conditional sufficient statistics for each mixture component are output as ‘particle’ files.

Mixture kernels are the product of independent multinomial densities for each categorical variable, and a multivariate normal density for continuous covariates. The base measure is conditionally

conjugate normal-Wishart-Dirichlet product, with Wishart hyperprior for inverse base covariance. Beta-autoregressive stick-breaking is used to model correlated densities.

Refer to Taddy (2009) for all specification details.

See DPreg demo for regression with categorical and continuous covariates, with additional Gibbs sampling for filtered particles.

See bar1D and bar2D demos for dynamic stick-breaking mixture density estimation, Bayes factor calculations, and comparison between correlated and independent model fit.

Value

If `print=TRUE` particle representations for each time point are printed to files `.particle[i].[t].[rho].txt`, where 'i' is the particle id, 't' is the time, and 'rho' is the stick-breaking correlation parameter. These files are read into R via the `particle()` function.

Output to R is minimal. The object returned by function `mix()` is mostly just a list of input variables, except for

| | |
|----------------------|---|
| <code>logprob</code> | Filtered marginal log-likelihood |
| <code>m</code> | Filtered mean number of mixture components |
| <code>k</code> | If <code>kout</code> , this is the list of filtered allocations |

Author(s)

Matt Taddy: <matt.taddy@chicagobooth.edu>

References

An auto-regressive mixture model for dynamic spatial Poisson processes: Application to tracking the intensity of violent crime (Taddy 2009),

Particle learning for general mixtures (Carvalho, Lopes, Polson, and Taddy 2009),

A Bayesian nonparametric approach to inference for quantile regression (Taddy and Kottas 2009).

and other papers at faculty.chicagobooth.edu/matt.taddy/research.

See Also

`particle`

Examples

```
demo(package="Bmix")
```

| | |
|----------|-------------------------------|
| particle | <i>Reading particle files</i> |
|----------|-------------------------------|

Description

This interprets the `.particle[i].[t].[rho].txt` files generated by the `mix` function and outputs a `data.frame` of sufficient statistics and posterior predictive information. The demos illustrate how to use the output for posterior inference.

Usage

```
particle(i, mixobj, t, rho=0)
```

Arguments

| | |
|---------------------|---|
| <code>i</code> | Particle index. |
| <code>mixobj</code> | Object returned by the <code>mix</code> function. |
| <code>t</code> | Time index. |
| <code>rho</code> | BAR process correlation parameter. |

Value

Returns a `data.frame` with each row corresponding to a mixture component (first is the base distribution). The first column, `n`, is the number of observations allocated to each mixture component, the next set of columns are sufficient statistics, column `p` is the predictive probability for each component, and the final columns are moments (and degrees of freedom) for the student-t and multinomial posterior predictive mixture densities.

Author(s)

Matt Taddy, <matt.taddy@chicagobooth.edu>

References

An auto-regressive mixture model for dynamic spatial Poisson processes: Application to tracking the intensity of violent crime (Taddy 2009),
and other papers at faculty.chicagobooth.edu/matt.taddy/research.

See Also

`mix`

Index

Bmix (Bmix-package), [2](#)

Bmix-package, [2](#)

mix (Bmix-package), [2](#)

particle, [4](#)