Package ‘spatial.gev.bma’

February 20, 2015

Type Package

Title Hierarchical spatial generalized extreme value (GEV) modeling with Bayesian Model Averaging (BMA)

Version 1.0

Date 2014-03-11

Author Alex Lenkoski

Depends SpatialExtremes,msm,coda

Maintainer Alex Lenkoski <alex@nr.no>

Description This package fits a hierarchical spatial model for the generalized extreme value distribution with the option of model averaging over the space of covariates.

License GPL

NeedsCompilation no

Repository CRAN

Date/Publication 2014-05-21 16:58:48

R topics documented:

`spatial.gev.bma-package`

Fit a Hierarchical Spatial Generalized Extreme Value model that allows for Bayesian Model Averaging

---

Description

This package allows one to fit a hierarhical spatial model for extremes. It enables linear specifications in all three GEV parameters and incorporates Bayesian Model Averaging to account for model uncertainty. Further, it uses recent results in automatic proposal formation of Metropolis-Hastings samplers to avoid requiring the user to have to specify a large number of tuning parameters.

Details

Package:	spatial.gev.bma
Type:	Package
Version:	1.0
Date:	2014-03-11
License:	GPL

Author(s)

Alex Lenkoski wrote the software for a specific project. He's more than happy to hear from you if you'd like to see additional features added.

Maintainer: Alex Lenkoski <alex@nr.no>

References

A. Dyrrdal, A. Lenkoski, T. Thorarinsdottir, F. Stordal (2014). Bayesian hierarchical modeling of extreme hourly precipitation in Norway. arxiv:1309.6111

---

dmvnorm — *Log density of a multivariate normal distribution*

Description

This returns the important bits of the log of a multivariate normal distribution, for use in Metropolis-Hastings ratios. One of these functions you wish they would just include in base so you don't have to keep rolling your own or calling some other package.

Usage

```
dmvnorm(x, mu, Sigma)
```

Arguments

`x`	A length p vector at which you would like the density to be evaluated
`mu`	A length p vector which is the mean of the MV normal
`Sigma`	A p x p matrix for the covariance of the MV normal

Value

The log density of a multivariate normal

Author(s)

Alex Lenkoski <alex@nr.no>

---

f.double.prime — *Second derivative of the posterior distribution of a spatial GEV with respect a location random effect*

Description

This is the second derivative of the conditional posterior probability of a sptial GEV model with respect to a given random effect in the location parameter. It is used to form the proposal for a Metropolis-Hastings update of the given parameter.

Usage

```
f.double.prime(tau, tau.hat, varsigma, xi, kappa, R)
```

Arguments

`tau`	The current value of the given random effect
`tau.hat`	The conditional mean of the random effect given the other random effects and the Gaussian process parameters
`varsigma`	The conditional variance of the random effect given the Gaussian process parameters
`xi`	The current shape parameter at this location
`kappa`	The current precision parameter at this location
`R`	The residual between the observed values and the linear part of the location parameter

Value

A scalar that gives the value of the derivative. This is then used to form the relevant proposal.

Author(s)

Alex Lenkoski <alex@nr.no>

`f.prime` *First derivative of the posterior of a spatial GEV model with respect to a random effect in the location parameter.*

Description

This internal function gives the first derivative of a spatial GEV model with respect to a random effect parameter in order to made a focused update of that variate.

Usage

```
f.prime(tau, tau.hat, varsigma, xi, kappa, R)
```

Arguments

`tau`	Current value of the random effect
`tau.hat`	Conditional mean of that random effect given the Gaussian process and the other random effects
`varsigma`	Conditional variance of the random effect given the Gaussian process
`xi`	Current shape parameter for that site
`kappa`	Current precision parameter for that site
`R`	Current vector of differences between the observations and the linear part of the location parameter at the site.

Value

A scalar giving the first derivative

Author(s)

Alex Lenkoski <alex@nr.no>

---

g.double.prime — *The second derivative of a GEV distribution with respect to a random effect parameter on the precision kappa*

Description

This function returns the second derivative of a GEV distribution with respect to the random effect tau_s^kappa. It is used in forming the proposal for the Metropolis-Hastings update of this parameter.

Usage

```
g.double.prime(tau, tau.hat, varsigma, xi, kappa.hat, eps)
```

Arguments

`tau`	The current value of the random effect
`tau.hat`	The conditional fitted value of the random effect given the others according to the Gaussian Process
`varsigma`	The conditional variance of the fitted value according to the Gaussian Process
`xi`	The current value of the shape parameter
`kappa.hat`	This is the linear bit of the precision for this location
`eps`	The vector of residuals for this site, i.e. Y_ts - mu_s

Value

A scalar giving the second derivative

Author(s)

Alex Lenkoski <alex@nr.no>

g.prime — *The first derivative of the posterior density of a spatial GEV model with respect to a given random effect on the precision parameter.*

Description

This returns the first derivative of the posterior density of a spatial GEV model with respect to a random effect on the precision parameter. It is used in forming the Metropolis-Hastings update of this parameter.

Usage

```
g.prime(tau, tau.hat, varsigma, xi, kappa.hat, eps)
```

Arguments

`tau`	Current value of the random effect
`tau.hat`	The conditional mean of the random effect given the others and the current Gaussian process parameters.
`varsigma`	The conditional variance of the random effect based on the Gaussian process parameters
`xi`	The current shape parameter for this location
`kappa.hat`	The linear part of the precision parameters
`eps`	The vector of residuals based on the observations at this site and the associated location parmeter

Value

A scalar giving the first derivative, which is used to form the Metropolis-Hasting update.

Author(s)

Alex Lenkoski <alex@nr.no>

gev.crps — *Compute the Continuous Rank Probability Score (CRPS)*

Description

This takes a large sample from a predictive distribution and computes the CRPS for a number of observations. It is used in judging the fit of predictive distributions.

Usage

```
gev.crps(Y.obs, Y.samp)
```

Arguments

`Y.obs`	A vector of observed values
`Y.samp`	A large sample from the posterior predictive distribution.

Value

A scalar giving the CRPS

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.impute` *Given the output of the MCMC, return a number of samples for a new site.*

Description

This takes the object returned from spatial.gev.bma and new information for a given site and returns a large sample of values from the posterior predictive distribution for that site.

Usage

```
gev.impute(R, X.drop, S.drop, burn = NULL, n.each = NULL)
```

Arguments

`R`	This is the object returned from spatial.gev.bma
`X.drop`	This is the vector of covariates for the site that has been left out.
`S.drop`	This is the 2 dimensional vector of locations for this site
`burn`	Optional number of repetitions to drop from R as burn in. Defaults to 10 percent.
`n.each`	Optional number of samples from each repetition under consideration. Defaults to a number such that around 1 million samples are returned.

Value

Returns a large number of sample from the posterior predictive distribution.

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.init` *Initilizes a state object for a Spatial GEV distribution*

Description

This utility function just initializes the state object that is then passed around during the course of an MCMC for a hierarchical spatial GEV distribution. It starts the object in a nice spot and fills in the relevant items.

Usage

```
gev.init(Y.list, X.all, S, prior.user, full, fixed.xi)
```

Arguments

`Y.list`	The list of Y values
`X.all`	The matrix of covariates
`S`	The matrix of spatial locations
`prior.user`	The user-specified prior parameters, if there are any
`full`	A boolean indicating if BMA is to be performed
`fixed.xi`	An indication if the parameter xi is to be estimated (NULL) or set to a given value.

Value

This returns a state object that is then used for the MCMC by the main updating routines.

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.like` *The log likelihood of a GEV distribution*

Description

Returns the log likelihood of a GEV distribution accoding to the parameterization used in spatial.gev.bma

Usage

```
gev.like(Y, mu, kappa, xi)
```

Arguments

`Y`	The outcome at which the log likelihood should be evaluated
`mu`	The location parameter
`kappa`	The precision parameter
`xi`	The shape parameter

Value

A scalar giving the log-likelihood

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.logscore` *Compute the Log Score*

Description

This takes a large sample from a predictive distribution and computes the log score for a vector of observations, judging the fit of predictive distributions.

Usage

```
gev.logscore(Y.obs, Y.samp)
```

Arguments

`Y.obs`	A vector of observed values
`Y.samp`	A large sample from the posterior predictive distribution.

Value

A scalar giving the mean log score over the observations

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.process.results` *Outputs some tables from the results of Spatial GEV MCMC run*

Description

This take the output from a spatial GEV run and returns some handy tables given posterior probabilities, effective sample sizes, acceptance probabilities, etc.

Usage

```
gev.process.results(R, burn = 100)
```

Arguments

`R`	The output from the spatial.gev.bma function
`burn`	An optional burn-in amount to be discarded before forming results.

Value

Returns a number of tables.

Author(s)

Alex <alex@nr.no>

`gev.results.init` *Initialize a results object for spatial.bma.gev*

Description

This internal function initializes the object that stores the results of an MCMC run in spatial.bma.gev

Usage

```
gev.results.init(n.s, p, n.reps)
```

Arguments

`n.s`	The number of sites
`p`	The number of covariates in X
`n.reps`	The total number of repetitions

Value

A object that will store the results.

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.update` *Updates all the parameters in a spatial GEV model*

Description

This internal function updates all the components of a spatial GEV model

Usage

```
gev.update(G)
```

Arguments

`G`	The current state of a spatial GEV model

Value

This returns an updated state

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.update.hyper` *Updates the Gaussian Process hyperparameters in the Spatial GEV model*

Description

This function is used internally to update the Gaussian Process hyperparameters of a Spatial GEV model.

Usage

```
gev.update.hyper(G)
```

Arguments

`G`	The current state of a Spatial GEV model

Value

An updated state of a Spatial GEV model.

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.update.lambda` *Update the lambda parameter in a Gaussian Process*

Description

This constructs a focused proposal to make a metropolis-hasting update of the lambda parameter in a Gaussian process. It is agnostic to which model component in the spatial GEV the GP belongs to.

Usage

```
gev.update.lambda(tau, alpha, lambda, D, a, b)
```

Arguments

`tau`	A vector of the random effects
`alpha`	The global precision of the Gaussian process
`lambda`	The current value of lambda
`D`	The matrix of distances between the points associated with the vector tau
`a`	The parameter for the prior of lambda
`b`	The second parameter for the prior of lambda

Value

A scalar which is the updated value of lambda.

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.update.M` *Sample a new model from the current model for any linear regression system*

Description

This uses a conditional Bayes factor (CBF) to update a model in a linear system given a current model and other information in a spatial GEV model. Note that it is agnostic to which part of the framework (location, precision, scale) you are updating.

Usage

```
gev.update.M(Y, X, M, alpha, lambda, D, beta.0, Omega.0)
```

Arguments

`Y`	The current dependent variable, calculated relative to the linear plus random effect terms of the given component.
`X`	The matrix of covariates
`M`	The current model. A subset of (1, ..., p) where p is the number of columns in X
`alpha`	The precision term of the Gaussian process for this component of the model
`lambda`	The length term of the Gaussian process for this component of the model
`D`	The distance matrix used in the Gaussian process
`beta.0`	The prior mean on the linear regression terms
`Omega.0`	The prior covariance on the linear regression terms

Value

This returns an updated model, which is a vector that is a subset of (1, ..., p).

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.update.tau.kappa` *Update the random effects of the precision parameter in a spatial GEV model*

Description

This goes through and updates, via M-H, each component in the random effects of a spatial GEV model

Usage

```
gev.update.tau.kappa(G)
```

Arguments

`G`	The current state of a spatial GEV model

Value

A new state of the spatial GEV model where the random effects have been updated

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.update.tau.mu` *Internal function to update the random effects of the location parameter in a Spatial GEV model.*

Description

This internal function takes the state of a Spatial GEV model, and updates the random effects for the location parameter

Usage

```
gev.update.tau.mu(G)
```

Arguments

G	An object holding the current state information for a Spatial GEV model

Value

An updated state for use in the Spatial GEV model.

Author(s)

Alex Lenkoski <alex@nr.no>

`gev.update.tau.xi` *Update the random effects for the shape parameter in a spatial GEV model*

Description

This goes through the random effects in the shape parameter of a hierarchical spatial GEV model and updates them individually using focused Metropolis-Hastings proposals.

Usage

```
gev.update.tau.xi(G)
```

Arguments

G	The current state of the model

Value

A new state of the model where the random effects have been updated.

Author(s)

Alex Lenkoski <alex@nr.no>

gev.update.theta — *Update the linear parameters in a spatial GEV model*

Description

This internal function updates the linear regression terms of a given component of a spatial GEV model. Due to blocking, the method is agnostic to which component it is updating

Usage

```
gev.update.theta(Y, X, M, alpha, lambda, D, beta.0, Omega.0)
```

Arguments

`Y`	The dependent valuesn
`X`	The matrix of covariates
`M`	The current model for this component
`alpha`	The overall variance of the GP
`lambda`	The scale of the GP
`D`	The distance matrix of the GP
`beta.0`	The prior mean for the covariates
`Omega.0`	The prior precision for the covariates

Value

Returns a new set of linear regression terms

Author(s)

Alex Lenkoski <alex@nr.no>

gev.z.p — *Calculate the 1/p return level for a GEV distribution*

Description

This simple function returns the 1/p return level for a GEV distribution

Usage

```
gev.z.p(p, mu, sigma, xi)
```

Arguments

`p`	The level on a 0 to 1 scale. Note that this is associated with the 1/p return level.
`mu`	The location parameter of a GEV distribution
`sigma`	The scale parameter of a GEV distribution
`xi`	The shape parameter of a GEV distribution

Value

A scalar giving the associated return level.

Author(s)

Alex Lenkoski <alex@nr.no>

`gp.like.lambda` *The likelihood of a Gaussian process used to initialize the lambda parameter*

Description

This is a utility that is used to initialize the lambda parameter in a Gaussian process when initializing the entire spatial GEV model. Its rather single service.

Usage

```
gp.like.lambda(lambda, alpha, tau, D)
```

Arguments

`lambda`	The lambda parameter
`alpha`	The alpha parmater of a gaussian process
`tau`	The vector of random effects
`D`	The distance matrix for the entries in tau

Value

The log density of the Gaussian process.

`j.double.prime` *The second derivative of a spatial GEV with respect to a random effect in the shape parameter*

Description

This internal function helps with the MCMC update of a RE on the shape parameter

Usage

```
j.double.prime(tau, tau.hat, varsigma, kappa, xi.hat, eps)
```

Arguments

`tau`	Current value of the RE
`tau.hat`	Conditional mean given the GP
`varsigma`	Conditional variance given the GP
`kappa`	Current precision for this site
`xi.hat`	Linear part of the shape parameter
`eps`	The residuals between the observation and the location parameter at this site

Value

A scalar giving the second derivative

Author(s)

Alex Lenkoski <alex@nr.no>

`j.prime` *The first derivative of the posterior density of a spatial GEV model with respect to a random effect parameter on the shape.*

Description

This internal function calculates the first derivative of a spatial GEV model with respect to a random effect parameter on the shape and is used during the Metropolis-Hastings update of this parameter conditional on everything else.

Usage

```
j.prime(tau, tau.hat, varsigma, kappa, xi.hat, eps)
```

Arguments

`tau`	Current value of the random effect
`tau.hat`	Conditional mean of the random effect given all others and the Gaussian process parameters.
`varsigma`	Conditional variance of the random effect according to the Gaussian process
`kappa`	Current value for the GEV scale at this site.
`xi.hat`	Current linear part of the GEV shape at this site.
`eps`	Vector of residuals for observations of this site given the current location value.

Value

A scalar giving the first derivative of this density

Author(s)

Alex Lenkoski <alex@nr.no>

`l.double.prime` *The second derivative of a Gaussian process with respect to the parameter lambda.*

Description

This utility returns the value of the second derivative of the posterior of lambda in a Gaussian process. It is used to make a focused M-H proposal when updating the lambda parameter.

Usage

```
l.double.prime(tau, alpha, lambda, D, a, b)
```

Arguments

`tau`	The vector of current random effects.
`alpha`	The current state of the precision of the Gaussian process
`lambda`	The current state of lambda
`D`	The distance matrix of the relevant entries in tau
`a`	The prior parameter for lambda
`b`	The second prior parameter for lambda

Value

This returns a scalar that is then used in the focused update. Its a pretty internal function.

Author(s)

Alex Lenkoski <alex@nr.no>

l.prime — *First derivative of a GP with respect to lambda*

Description

This internal function gives the first derivative of a GP with respect to the parameter lambda

Usage

```
l.prime(tau, alpha, lambda, D, a, b)
```

Arguments

`tau`	The current vector of REs
`alpha`	The current overall variance of the process
`lambda`	The current value of lambda
`D`	The distance matrix
`a`	The first prior parameter for lambda
`b`	The second prior parameter for lambda

Value

A scalar giving the derivative

Author(s)

Alex Lenkoski <alex@nr.no>

logdet — *Returns the log determinant for a symmetric positive definite matrix.*

Description

This internal function returns the log determinant of a positive definite matrix by using the trace of a the Cholesky trick. Its way faster than what you find in base.

Usage

```
logdet(A)
```

Arguments

`A`	A symmetric positive definite matrix.

Value

A scalar giving the log determinant of A

Author(s)

Alex Lenkoski <alex@nr.no>

---

make.D — *Form the distance matrix for use in a Gaussian Process*

Description

This takes two matrices containing the (lat,lon) locations of two collections of points and returns the matrix of euclidean distances between them.

Usage

```
make.D(x.1, x.2)
```

Arguments

`x.1`	An n1 by 2 matrix of the locations of the first set of points
`x.2`	An n2 by 2 matrix of the locations of the second set of points

Value

Returns an n1 by n2 matrix with the relevant Euclidean distances

Author(s)

Alex Lenkoski <alex@nr.no>

---

norway — *Extreme Precipitation Data at 69 Sites in Norway*

Description

This dataset provides observations of extreme precipitation in Norway at 69 sites, along with covariate information.

Usage

```
data(norway)
```

Format

A list containing three objects. Y.list is itself a list, with the observations at each site. X is a matrix with the coviates and S is a matrix with their relative position `spatial.gev.bma` *Run an MCMC to fit a hierarchical spatial generalized extreme value (GEV) model with the option for Bayesian model averaging (BMA)*

Description

This is the main function of the spatial.gev.bma package. It runs an MCMC to sample the posterior distribution of a spatial GEV model that takes a number of covariates for linear interactions as well as a spatial component to model dependent overdispersion.

Usage

```
spatial.gev.bma(Y.list, X.all, S, n.reps, prior.user = NULL,
full = FALSE, fixed.xi = NULL, print.every = 0)
```

Arguments

`Y.list`	This is a list of length n, where n is the total number of sites you have data for. Each element of the list is a vector of observations at that site.
`X.all`	This is an n by p matrix where n is the total number of sites and p is the total number of variables under consideration. Please supply your own constant in the first column. Note that this model implicitly assumes that at a given site, the covariates are constant throughout the observation period. Email me if you need something more flexible.
`S`	This is an n by 2 matrix with the coordinates of each of your n sites.
`n.reps`	The total number of repititions (i.e. includes burn in) that you would like to run the MCMC.
`prior.user`	A list containing any priors that the user would like to specify.
`full`	A boolean indicating if you'd like to force all variables to be included (TRUE) or whether you want to perform BMA (FALSE)
`fixed.xi`	If set to NULL, the shape parameter is estimated like the location and scale. If a numeric value is provided, the shape parameter is fixed at this level.
`print.every`	A simple tool to tell you how many iterations you've run. If you set it to 0 then no progress reports will be given. If it is a positive number, say x, then every x iterations it will tell you the total number of iterations.

Value

`TAU`	An n by reps by 3 array of the states of the random effects at each iteration
`THETA`	An reps by p by 3 array of the states of the linear terms at each iteration
`ALPHA`	A reps by 3 matrix of states of the alpha term at each iteration
`LAMBDA`	A reps by 3 matrix of states of the lambda term at each iteration

Author(s)

Alex Lenkoski <alex@nr.no>

Examples

```
data(norway)
attach(norway)
##To replicate our results, change 2 to 2e5 below
a <- spatial.gev.bma(Y.list,X,S,2)
```

Index